INTEGRATED CIRCUITS

DATA HANDBOOK

Programmable Logic Devices (PLD)

BOOK IC13 1991

Signetics
Philips Components

PHILIPS

Programmable Logic Devices Data Handbook 1991

Signetics

Philips Components

PHILIPS

Signetics reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Signetics assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Signetics makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS

Signetics Products are not designed for use in life support appliances, devices, or systems where malfunction of a Signetics Product can reasonably be expected to result in a personal injury. Signetics customers using or selling Signetics Products for use in such applications do so at their own risk and agree to fully indemnify Signetics for any damages resulting from such improper use or sale.

Signetics registers eligible circuits under
the Semiconductor Chip Protection Act.

Preface

Programmable Logic Devices

The 1991 Philips Components–Signetics PLD Data Handbook is loaded with information on new parts. Using the fastest technologies in the most innovative architectures, today's system designer can pick from the largest selection of PLDs in the industry. Featured in 1991 are the ultra high-speed BiCMOS devices designed to be pin, function, and fusemap identical to existing industry standard parts. Some highlights of this handbook include the fastest silicon PLDs available (PHD16N8 and PHD48N22)—at 5 nanoseconds! These devices make ideal decoders to squeeze maximum performance from powerful microprocessors.

Designers using DRAM, VRAM and graphics will appreciate the speed and power of the new line of sequencers which include the PLC415, PLC42VA12, PLUS405 and PLUS105. These sequencers also make innovative bus and LAN controllers for emerging standard protocols.

At last, the logical power of dual programmable arrays comes forth in the PLUS153 and PLUS173 devices—at 10 nanosecond propagation delays.

The PLC18V8Z is the only zero power 20-pin device which can replace 16V8's!

For maximum density in a truly compact system, the Programmable Macro Logic family now boasts four members—the PML2552 and PLHS601 are added to the original PLHS501 and PLHS502. The PML2552 is the PLD industry's first dense device to implement SCAN test.

To complement the devices, SLICE design software is offered through our Sales Offices (see Section 11) and SNAP software is available for high level support. Read about them under Product Support.

Expanding customer service has been an ongoing effort. Our Applications staff continues to answer your technical questions on PLD designs and our free computer Bulletin Board, with 24-hour service, is at (800)451-6644.

New PLD users are encouraged to read the Applications Section for an overview of PLD ideas. More seasoned PLD users are encouraged to go through the PLD applications and PML applications at the end of the handbook to gain understanding and ideas for new designs.

Product Status

Programmable Logic Devices

DEFINITIONS		
Data Sheet Identification	**Product Status**	**Definition**
Objective Specification	Formative or In Design	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
Preliminary Specification	Preproduction Product	This data sheet contains preliminary data and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	Full Production	This data sheet contains Final Specifications. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Contents

Programmable Logic Devices

Contents

Contents

Contents

Section 1
General Information

INDEX

Alphanumeric Index

Programmable Logic Devices

Selection Guide

Programmable Logic Devices

SIGNETICS PART NUMBER	ARCHITECTURE (INPUTS × TERMS × OUTPUTS)	PACKAGE	TOTAL INPUTS (# Dedicated)	LOGIC TERMS	INTERNAL STATE REGISTERS (# Dedicated)	OUTPUTS C, I/O, R, R I/O	t_{PD} (Max)	f_{MAX}	I_{CC} (Max)
PAL®-TYPE DEVICES									
PLQ16L8-5	16×64×8	20-Pin	16 (10)	64	0	2 C, 6 I/O	5ns		180mA
PLQ16R4-5	16×64×8	20-Pin	16 (8)	64	4 (0)	4 I/O, 4 R	5ns	118MHz	180mA
PLQ16R6-5	16×64×8	20-Pin	16 (8)	64	6 (0)	2 I/O, 6 R	5ns	118MHz	180mA
PLQ16R8-5	16×64×8	20-Pin	16 (8)	64	8 (0)	8 R		118MHz	180mA
PLUS16L8-7	16×64×8	20-Pin	16 (10)	64	0	2 C, 6 I/O	7.5ns		180mA
PLUS16R4-7	16×64×8	20-Pin	16 (8)	64	4 (0)	4 I/O, 4 R	7.5ns	74MHz	180mA
PLUS16R6-7	16×64×8	20-Pin	16 (8)	64	6 (0)	2 I/O, 6 R	7.5ns	74MHz	180mA
PLUS16R8-7	16×64×8	20-Pin	16 (8)	64	8 (0)	8 R		74MHz	180mA
PLUS16L8D	16×64×8	20-Pin	16 (10)	64	0	2 C, 6 I/O	10ns		180mA
PLUS16R4D	16×64×8	20-Pin	16 (8)	64	4 (0)	4 I/O, 4 R	10ns	60MHz	180mA
PLUS16R6D	16×64×8	20-Pin	16 (8)	64	6 (0)	2 I/O, 6R	10ns	60MHz	180mA
PLUS16R8D	16×64×8	20-Pin	16 (8)	64	8 (0)	8 R		60MHz	180mA
PLQ20L8-5	20×64×8	24-Pin	20 (14)	64	0	2 C, 6 I/O	5ns		210mA
PLQ20R4-5	20×64×8	24-Pin	20 (12)	64	4 (0)	4 I/O, 4 R	5ns	118MHz	210mA
PLQ20R6-5	20×64×8	24-Pin	20 (12)	64	6 (0)	2 I/O, 6 R	5ns	118MHz	210mA
PLQ20R8-5	20×64×8	24-Pin	20 (12)	64	8 (0)	8 R		118MHz	210mA
PLUS20L8-7	20×64×8	24-Pin	20 (14)	64	0	2 C, 6 I/O	7.5ns		210mA
PLUS20R4-7	20×64×8	24-Pin	20 (12)	64	4 (0)	4 I/O, 4 R	7.5ns	74MHz	210mA
PLUS20R6-7	20×64×8	24-Pin	20 (12)	64	6 (0)	2 I/O, 6 R	7.5ns	74MHz	210mA
PLUS20R8-7	20×64×8	24-Pin	20 (12)	64	8 (0)	8 R		74MHz	210mA
PLUS20L8D	20×64×8	24-Pin	20 (14)	64	0	2 C, 6 I/O	10ns		210mA
PLUS20R4D	20×64×8	24-Pin	20 (12)	64	4 (0)	4 I/O, 4R	10ns	60MHz	210mA
PLUS20R6D	20×64×8	24-Pin	20 (12)	64	6 (0)	2 I/O, 6 R	10ns	60MHz	210mA
PLUS20R8D	20×64×8	24-Pin	20 (12)	64	8 (0)	8 R		60MHz	210mA
PLHS16L8A	16×64×8	20-Pin	16 (10)	64	0	2 C, 6 I/O	20ns		155mA
PLHS16L8B	16×64×8	20-Pin	16 (10)	64	0	2 C, 6 I/O	15ns		155mA
PLHS18P8A	18×72×8	20-Pin	18 (10)	72	0	8 I/O	20ns		155mA
PLHS18P8B	18×72×8	20-Pin	18 (10)	72	0	8 I/O	15ns		155mA
PHD16N8-5	16×16×8	20-Pin	16 (10)	16	0	2 C, 6 I/O	5ns		180mA
PHD48N22-7	48×73×22	68-Pin	48 (36)	73	0	10 C, 12 I/O	7.5ns		420mA
PLC18V8Z/18V8ZI	18×74×8	20-Pin	18 (8)	74	8 (0)	8 varied	35, 40ns	21MHz	100µA, 1mA/MHz
PLQ22V10-7	22×132×10	24-Pin	22 (12)	132	10 (0)	10 R, 1 I/O	7.5ns	87MHz	
10H20EV8/10020EV8**	20×90×8	24-Pin	20 (12)	90	0	8 varied	4.5ns	222MHz	230mA
PLA									
PLS100/101	16×48×8	28-Pin	16 (16)	48	0	8 C	50ns		170mA
PLS153	18×42×10	20-Pin	18 (8)	42	0	10 I/O	40ns		155mA
PLS153A	18×42×10	20-Pin	18 (8)	42	0	10 I/O	30ns		155mA
PLUS153B	18×42×10	20-Pin	18 (8)	42	0	10 I/O	15ns		200mA
PLUS153D	18×42×10	20-Pin	18 (8)	42	0	10 I/O	12ns		200mA
PLUS153-10**	18×42×10	20-Pin	18 (8)	42	0	10 I/O	10ns		200mA
PLS173	22×42×10	24-Pin	22 (12)	42	0	10 I/O	30ns		170mA
PLUS173B	22×42×10	24-Pin	22 (12)	42	0	10 I/O	15ns		200mA
PLUS173D	22×42×10	24-Pin	22 (12)	42	0	10 I/O	12ns		200mA
PLUS173-10**	22×42×10	24-Pin	22 (12)	42	0	10 I/O	10ns		210mA

Selection Guide

SIGNETICS PART NUMBER	ARCHITECTURE (INPUTS × TERMS × OUTPUTS)	PACKAGE	TOTAL INPUTS (# Dedicated)	LOGIC TERMS	INTERNAL STATE REGISTERS (# Dedicated)	OUTPUTS C, I/O, R, R I/O	t_{PD} (Max)	f_{MAX}	I_{CC} (Max)
PLS									
PLS105	16 × 48 × 8	28-Pin	22 (16)	48	6 (6)	8 R		14MHz	180mA
PLS105A	16 × 48 × 8	28-Pin	22 (16)	48	6 (6)	8 R		20MHz	180mA
PLUS105–45**	16 × 48 × 8	28-Pin	22 (16)	48	6 (6)	8 R		45MHz	200mA
PLUS105–55**	16 × 48 × 8	28-Pin	22 (16)	48	6 (6)	8 R		55MHz	200mA
PLUS405–37	16 × 64 × 8	28-Pin	24 (16)	64	8 (8)	8 R		37MHz	225mA
PLUS405–45	16 × 64 × 8	28-Pin	24 (16)	64	8 (8)	8 R		45MHz	225mA
PLUS405–55**	16 × 64 × 8	28-Pin	24 (16)	64	8 (8)	8 R		55MHz	225mA
PLS155	16 × 45 × 12	20-Pin	16 (4)	45	4 (0)	8 I/O, 4 R I/O	50ns	14MHz	190mA
PLS157	16 × 45 × 12	20-Pin	16 (4)	45	6 (0)	6 I/O, 6 R I/O	50ns	14MHz	190mA
PLS159A	16 × 45 × 12	20-Pin	16 (4)	45	8 (0)	4 I/O, 8 R I/O	35ns	18MHz	190mA
PLS167	14 × 48 × 6	24-Pin	22 (14)	48	8 (6)	6 R		14MHz	180mA
PLS167A	14 × 48 × 6	24-Pin	22 (14)	48	8 (6)	6 R		20MHz	180mA
PLS168	12 × 48 × 6	24-Pin	22 (12)	48	10 (6)	8 R		14MHz	180mA
PLS168A	12 × 48 × 6	24-Pin	22 (12)	48	10 (6)	8 R		20MHz	180mA
PLS179	20 × 45 × 12	24-Pin	20 (8)	45	8 (0)	4 I/O, 8 R I/O	35ns	18MHz	210mA
PLC42VA12	42 × 105 × 12	24-Pin	42 (10)	105	10 (0)	10 C or R I/O, 2 I/O	35ns	25MHz	90mA*
PLC415–16	17 × 68 × 8	28-Pin	25 (17)	68	8 (8)	8 R			100µA/ 80mA
PML™									
PLHS501	104 × 116 × 24	52-Pin	24	116	0	16 C, 8 I/O	22ns		295mA
PLHS502	128 × 144 × 24	68-Pin	24	144	16 (16)	16 C or R, 8 I/O or R I/O	20ns	50MHz	370mA
PLHS601	68 × 134 × 24	68-Pin	28	134	0	12 C, 12 I/O	21ns		340mA
PML2552	185 × 226 × 24	68-Pin	29	226	36 (20)	24 C, 16 R, 16 I/O	40ns	50MHz	100mA

PAL-Type = Programmable Array Logic (Fixed OR Array)-Type
PHD = Programmable High-Speed Decoder
PLA = Programmable Logic Array
PLS = Programmable Logic Sequencer
PML = Programmable Macro Logic

OUTPUTS:
C = Combinatorial output
R = Registered output
I/O = Combinatorial I/O
R I/O = Registered I/O

NOTES:
$f_{MAX} = 1/(t_{IS} + t_{CKO})$ worst case
* Measured at 15MHz (TTL input level)
** Under development
PAL is a trademark of AMD/MMI.
PML is a trademark of Philips Components–Signetics.

Ordering Information

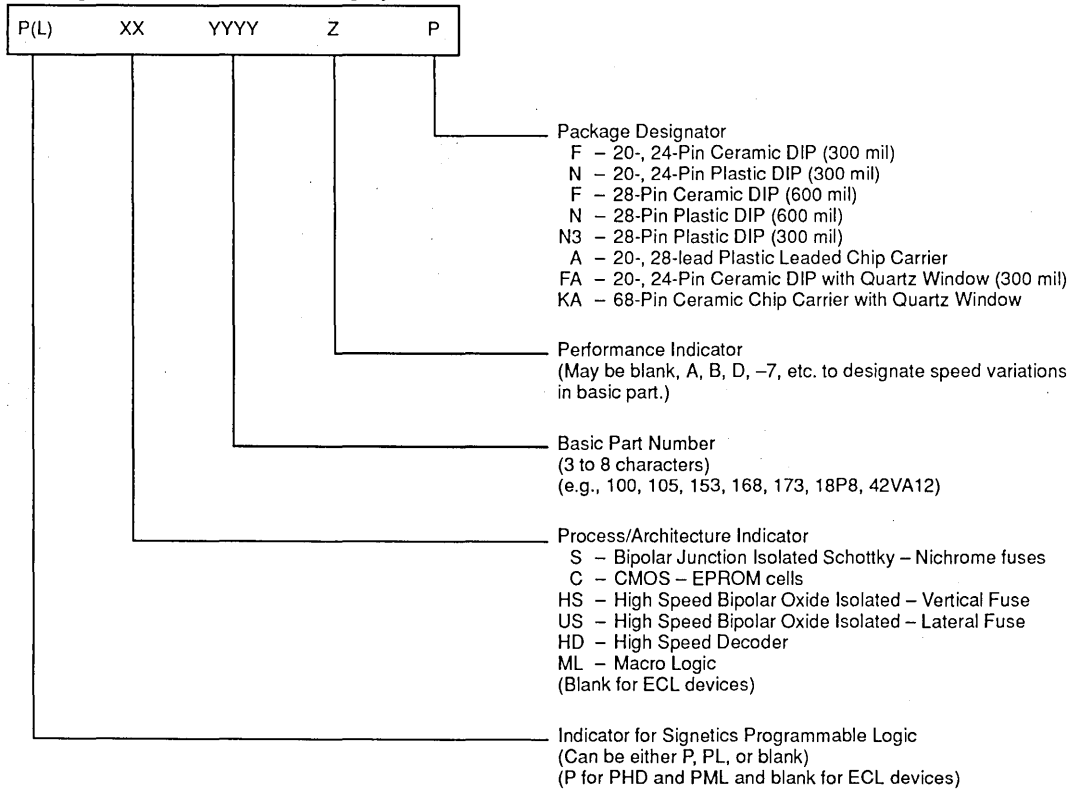
Programmable Logic Devices

Philips Components–Signetics Programmable Logic Devices may be ordered by contacting either the local Signetics sales office, Signetics representatives or authorized distributors. A complete listing is located in the back of this handbook.

Table 1 provides part number definition for Signetics PLDs. The Signetics part number system allows complete ordering information to specified in the part number. The part number and product description is located on each data sheet.

Military versions of these commercial products may be ordered. Please refer to the military products data handbook for complete ordering information.

New Signetics PLD Part Numbering System



Section 2
Introduction

INDEX

Introduction Programmable Logic

Programmable Logic Devices

WHAT IS PROGRAMMABLE LOGIC?

In 1975, Signetics Corporation developed a new product family by combining its expertise in semi-custom gate array products and fuse-link Programmable Read Only Memories (PROMs). Out of this marriage came Signetics Programmable Logic Family. The PLS100 Field-Programmable Logic Array (FPLA) was the first member of this family. The FPLA was an important industry first in two ways. First, the AND/OR/INVERT architecture allowed the custom implementations of Sum of Product logic equations. Second, the three-level fusing allows complete flexibility in the use of this device family. All logic interconnections from input to output are programmable.

Figure 1 shows the architecture of a high performance sequencer combining a PLA architecture with JK flip-flops. Table 1 shows the current spectrum of Philips Components–Signetics PLDs. Parts for every need are available in nearly every architecture and across at least three technologies. the PLUS and PLHS prefixes describe bipolar parts, the PLC prefix describes EPLD (CMOS) parts and the PLQ prefix refers to the new Signetics QUBiC BiCMOS process. Figure 2 shows a shorthand image of the PLUS153 programmable logic array (PLA), which was derived from the original PLS100.

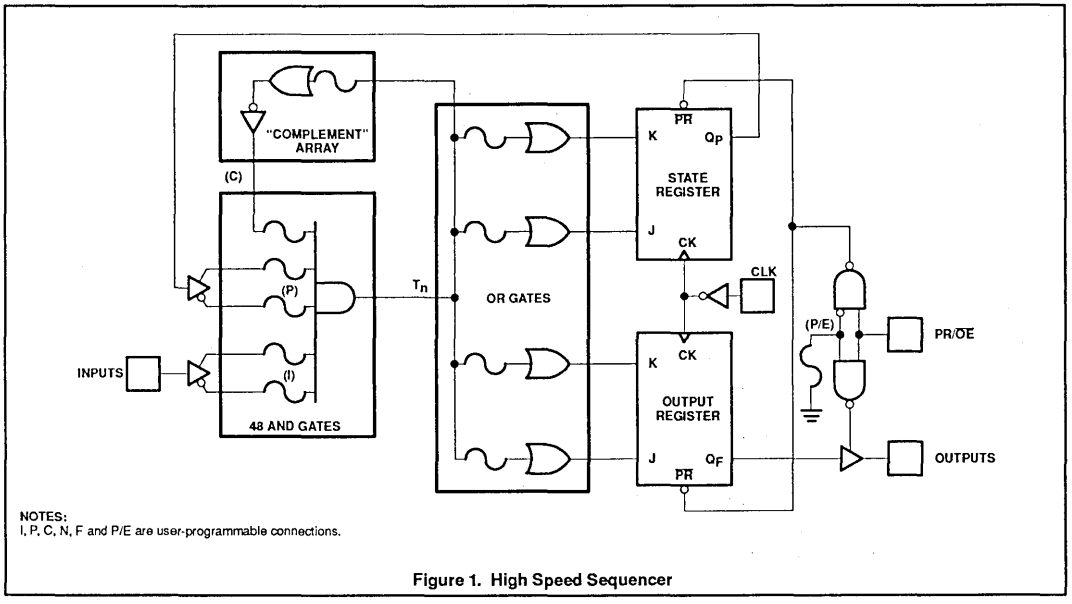


Figure 1. High Speed Sequencer

Table 1. PLD Product Family

PART NUMBER	TYPE	CONFIGURATION
	20-PIN	
PHD16N8	PHD	12-Input/8-Output
PLS153/153A	PLA	18-Input/10-Output – 42-Term
PLUS153B/153D/153–10	PLA	18-Input/10-Output – 42-Term
PLS155-159A	PLS	16-Input/12-Output – 45-Term
PLS155	PLS	4 Registered Outputs
PLS157	PLS	6 Registered Outputs
PLS159A	PLS	8 Registered Outputs
PLHS18P8A/B	PAL-Type	18-Input/10-Output – 72-Term
PLHS16L8A/B	PAL-Type	16-Input/8-Output – 64-Term
PLC18V8Z/I	PAL-Type	18-Input/8-Output – 72-Term
PLQ16L8–5	PAL-Type	16-Input/8-Output – 72-Term
PLQ16R4–5	PAL-Type	16-Input/4-Output, 4 Registers
PLQ16R6–5	PAL-Type	16-Input/6-Output, 6 Registers
PLQ16R8–5	PAL-Type	16-Input/8-Output, 8 Registers
PLUS16L8D/–7	PAL-Type	16-Input/8-Output – 72-Term
PLUS16R4D/–7	PAL-Type	16-Input/4-Output, 4 Registers
PLUS16R6D/–7	PAL-Type	16-Input/6-Output, 6 Registers
PLUS16R8D/–7	PAL-Type	16-Input/8-Output, 8 Registers
	24-PIN	
PLS167/A	PLS	14-Input/6-Output – 48-Term 8-Bit State Registers 6-Output Registers
PLS168/A	PLS	12-Input/8-Output – 48-Term 10-Bit State Registers 8-Output Registers
PLS173/PLUS173B/D/–10	PLA	22-Input/10-Output – 42-Term
PLS179	PLS	22-Input/10-Output – 42-Term
PLC42VA12	PLS	42-Input/12-Output, 10 Registers
PLQ20L8–5	PAL-Type	20-Input/8-Output
PLQ20R4–5	PAL-Type	20-Input/4-Output
PLQ20R6–5	PAL-Type	20-Input/6-Output
PLQ20R8–5	PAL-Type	20-Input/8-Output
PLQ22V10–7	PAL-Type	22-Input – 10 Registered I/O – 132-Term
PLUS20L8D/–7	PAL-Type	20-Input/8-Output
PLUS20R4D/–7	PAL-Type	20-Input/4-Output
PLUS20R6D/–7	PAL-Type	20-Input/6-Output
PLUS20R8D/–7	PAL-Type	20-Input/8-Output
	28-PIN	
PLC415–16	PLS	17-Input/8-Output – 68-Term 8-Bit State and Output Registers
PLS100/101	PLA	16-Input/8-Output – 48-Term
PLS105/105A/PLUS105	PLS	16-Input/8-Output – 48-Term 6-Bit State Register 8-Output Registers
PLUS405	PLS	16-Input/8-Output – 64-Term 8-Bit State and Output Registers
	52-, 68-PIN	
PLHS501	PML	32-Input/24-Output – 116-Term
PLHS502	PML	32-Input/24-Output – 144-Term
PLHS601	PML	40-Input/24-Output – 150-Term
PML2552	PML	53-Input/24-Output – 226-Term
PHD48N22	PHD	48-Input/22-Output

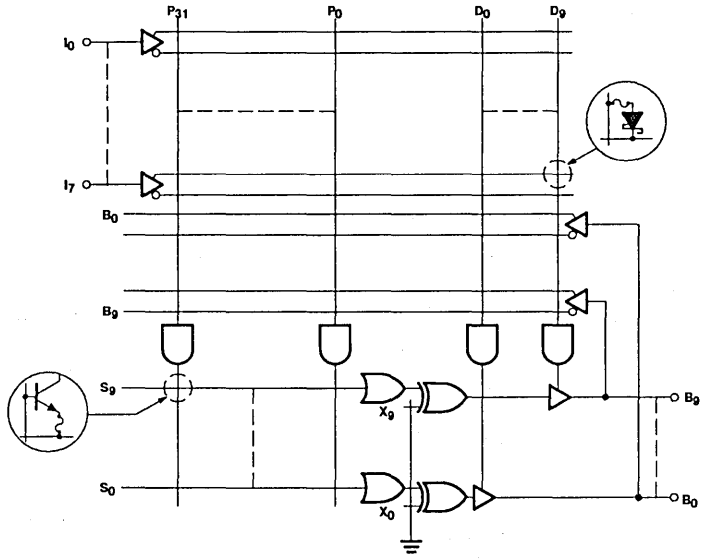


Figure 2. PLUS153 20-Pin Functional Diagram

PLD LOGIC SYNTHESIS

No intermediate step is required to implement Boolean Logic Equations with PLDs. Each term in each equation simply becomes a direct entry into the Logic Program Table. The following example illustrates this straightforward concept:

$X_0 = AB + \overline{C}D + B\overline{D}$

$\overline{X}_1 = \overline{A}B + \overline{C}D + EFG$

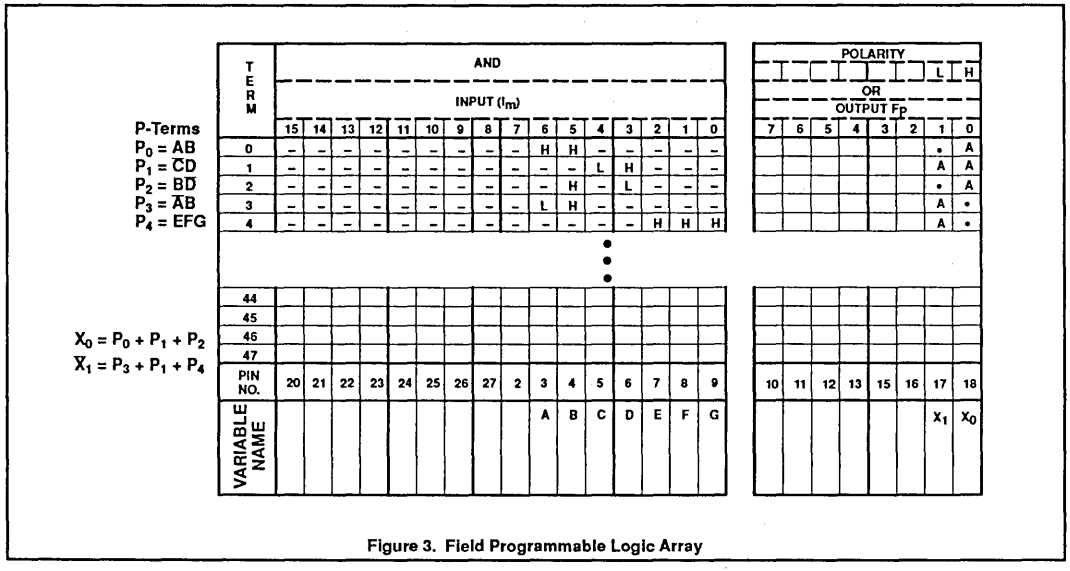


Figure 3. Field Programmable Logic Array

Figure 4. Equivalent Fixed Logic Diagram

In the previous example, the two Boolean Logic equations were broken into Product terms. Each P-term was then programmed into the P-term section of the PLA Program Table. This was accomplished in the following manner:

Step 1

Select which input pins $I_0 - I_{15}$ will correspond to the input variables. In this case A – G are the input variable names. I_6 through I_0 were selected to accept inputs A – G respectively.

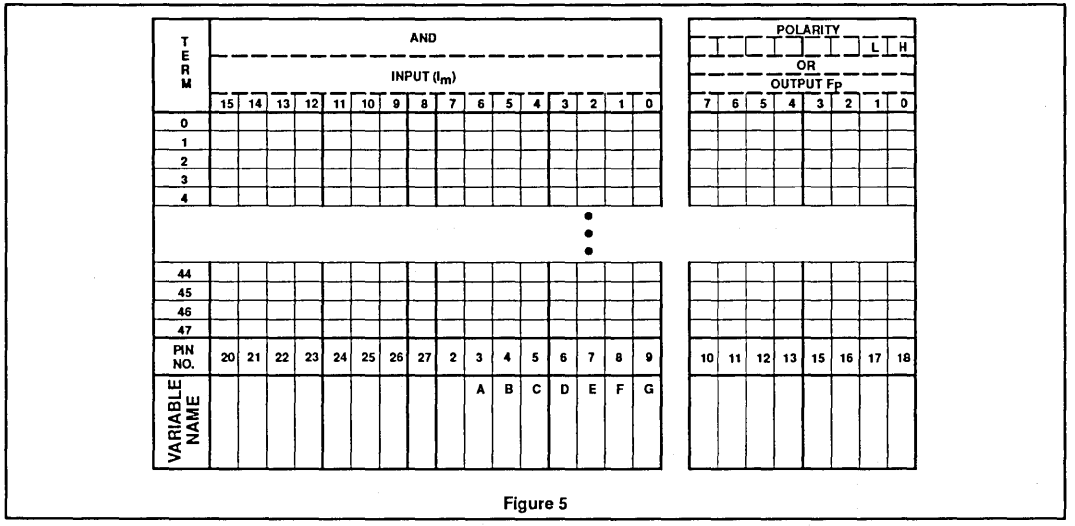


Figure 5

Step 2

Transfer the Boolean Terms to the PLA Program Table. This is done simply by defining each term and entering it on the Program Table.

e.g., $P_0 = AB$

This P-term translates to the Program Table by selecting $A = I_6 = H$ and $B = I_5 = H$ and entering the information in the appropriate column.

$P_1 = \overline{C}D$

This term is defined by selecting $C = I_4 = L$ and $D = I_3 = H$, and entering the data into the Program Table. Continue this operation until all P-terms are entered into the Program Table.

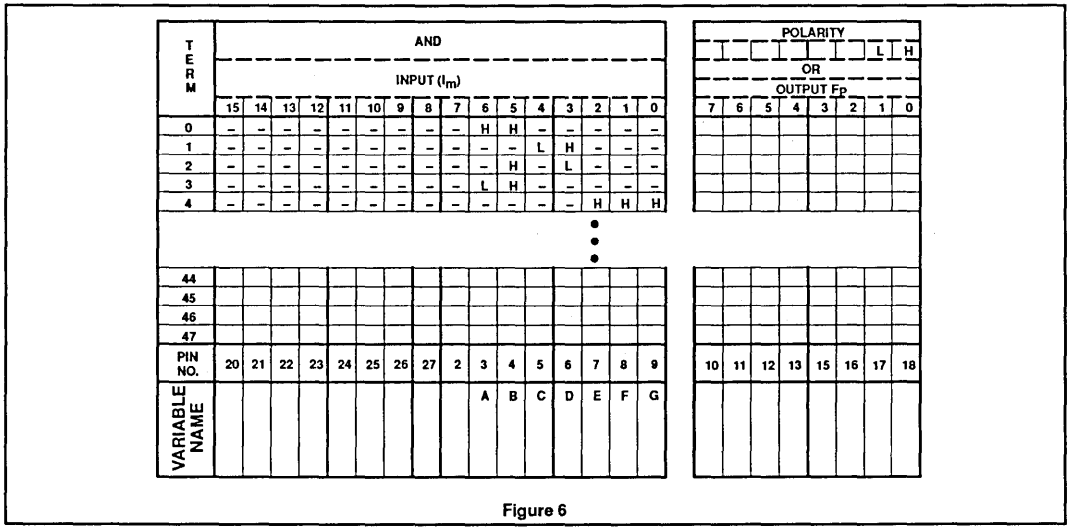


Figure 6

Step 3

Select which output pins correspond to each output function. In this case F_0 = Pin 18 = X_0, and F_1 = Pin 17 = X_1.

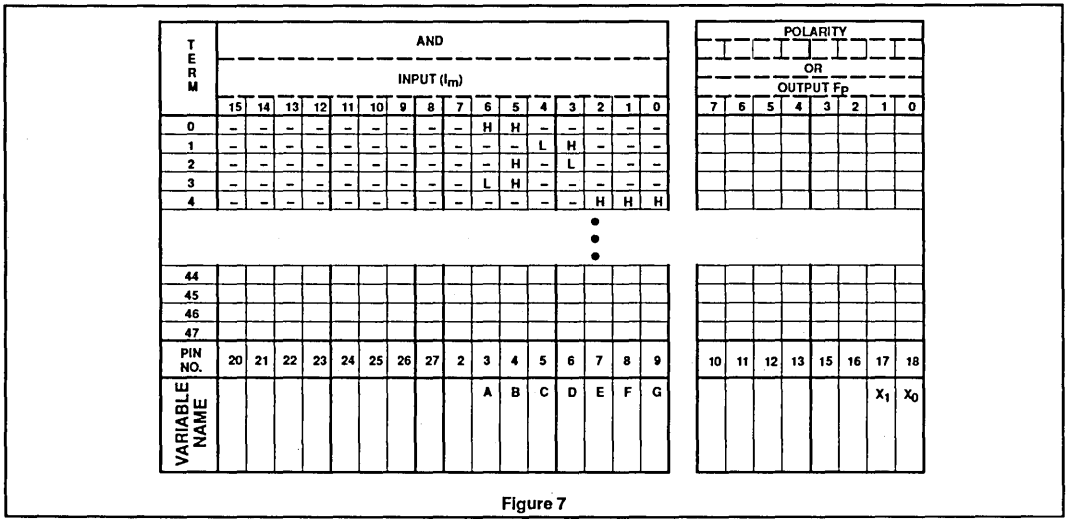


Figure 7

Step 4

Select the Output Active Level desired for each Output Function. For X_0 the active level is high for a positive logic expression of this equation. Therefore, it is only necessary to place an (H) in the Active Level box above Output Function 0, (F_0). Conversely, X_1 can be expressed as $\overline{X}_1$ by placing an (L) in the Active Level box above Output Function 1, (F_1).

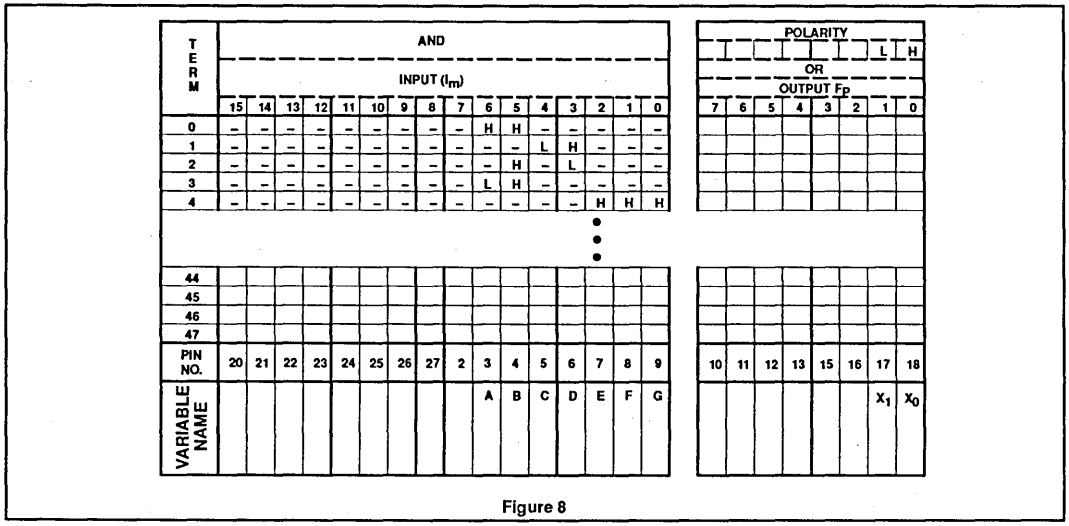


Figure 8

Step 5

Select the P-Terms you wish to make active for each Output Function. In this case $X_0 = P_0 + P_1 + P_2$, so an A has been placed in the intersection box for P_0 and X_0, P_1 and X_0 and P_2 and X_0.

Terms which are not active for a given output are made inactive by placing a (•) in the box under that P-term. Leave all unused P-terms unprogrammed.

Continue this operation until all outputs have been defined in the Program Table.

Step 6

Enter the data into a Signetics approved programmer. The input format is identical to the Signetics Program Table. You specify the P-terms, Output Active Level, and which P-terms are active for each output exactly the way it appears on the Program Table.

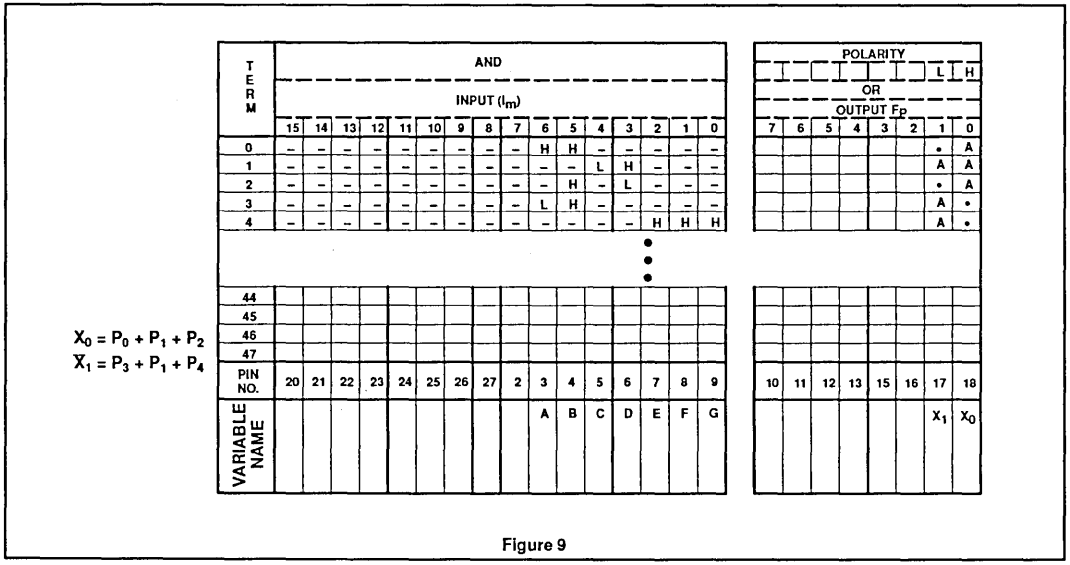


Figure 9

PLD LOGIC SYNTHESIS
(Continued)

When fewer inputs and outputs are required in a logic design and low cost is most important, the Signetics 20-pin PLD should be considered first choice. The PLUS153 is a PLA with 8 inputs, 10 I/O pins, and 42 product terms. The user can configure the device by defining the direction of the I/O pins. This is easily accomplished by using the direction control terms $D_0 - D_9$ to establish the direction of pins $B_0 - B_9$. The D-terms control the 3-State buffers found on the outputs of the Ex-OR gates. Figures 10 and 11 show how the D-term configures each B_X pin.

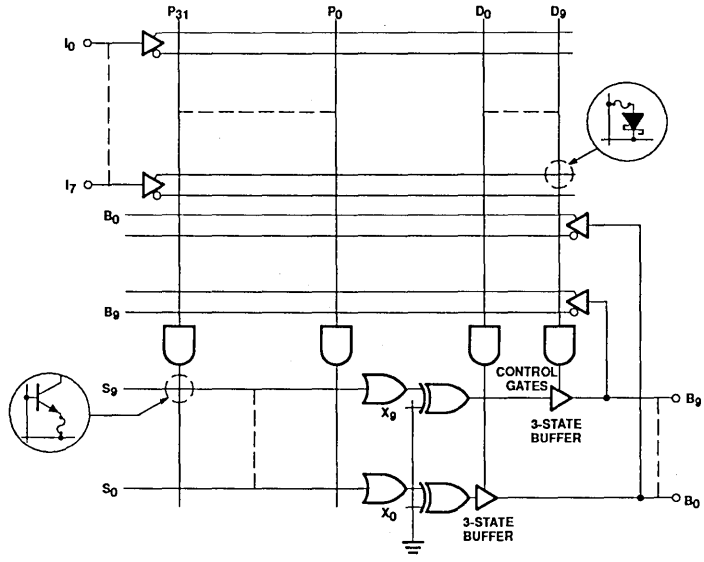
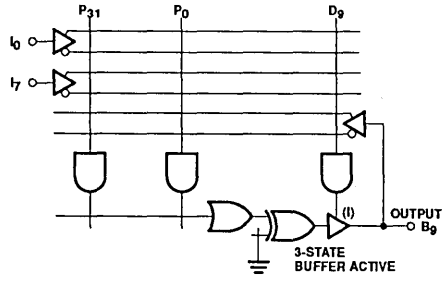
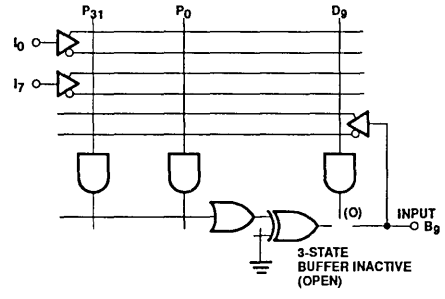


Figure 10. PLUS153 Functional Diagram



a. D_9 Active Makes B_9 Appear as an Output with Feedback



b. D_9 Inactive Makes B_9 Appear as an Input

Figure 11

Programmable Logic — Introduction

To control each D-term, it is necessary to understand that each control gate is a 36-input AND gate. To make the 3-State buffer active (B_x pin an output), the output of the control gate must be at logic HIGH (1). This can be accomplished in one of two ways. A HIGH can be forced on all control gate input nodes, or fuses can be programmed. When a fuse is programmed, that control gate input node is internally pulled up to HIGH (1). See Figure 12 and Figure 13.

Programming the fuse permanently places a HIGH (1) on the input to the control gate. The input pin no longer has any effect on that state.

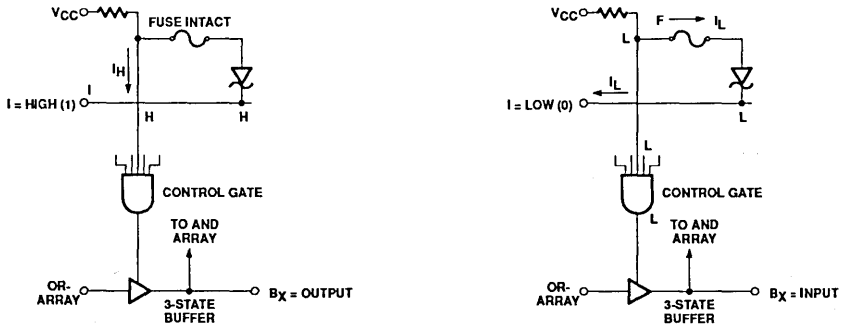


Figure 12. Input Effect on Control Gates (Fuse Intact)

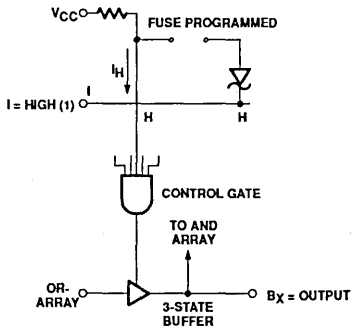


Figure 13. Effect on Control Gate if Fuse is Programmed

DEDICATING B_X PIN DIRECTION

Since each input to the D-terms is true and complement buffered (see Figure 11), when the device is shipped with all fuses intact, all control gates have half of the 36 input lines at logic low (0). The result of this is all Control Gate outputs are low (0) and the 3-State buffers are inactive. This results in all B_X pins being in the input condition. the resultant device is, therefore, an 18-input, 0-output FPLA. While useful as a bit bucket or Write-Only-Memory (WOM), most applications require at least one output. Clearly, the first task is to determine which of the B_X pins are to be outputs. The next step is to condition the control gate to make the 3-State buffer for those gates active. To dedicate B_0 and B_1 as outputs, it is necessary to program all fuses to the inputs to Control Gates D_0 and D_1. This internally pulls all inputs to those gates to HIGH (1) permanently. since all inputs to the Control Gates are HIGH (1), the output is HIGH (1) and the 3-State buffers for B_0 and B_1 are active. This permanently enables B_0 and B_1 as outputs. Note that even though B_0 and B_1 are outputs, the output data is available to the AND array via the internal feedback (see Figure 11a).

To program this data, the PLUS153 Program Table is used as shown in Figure 14.

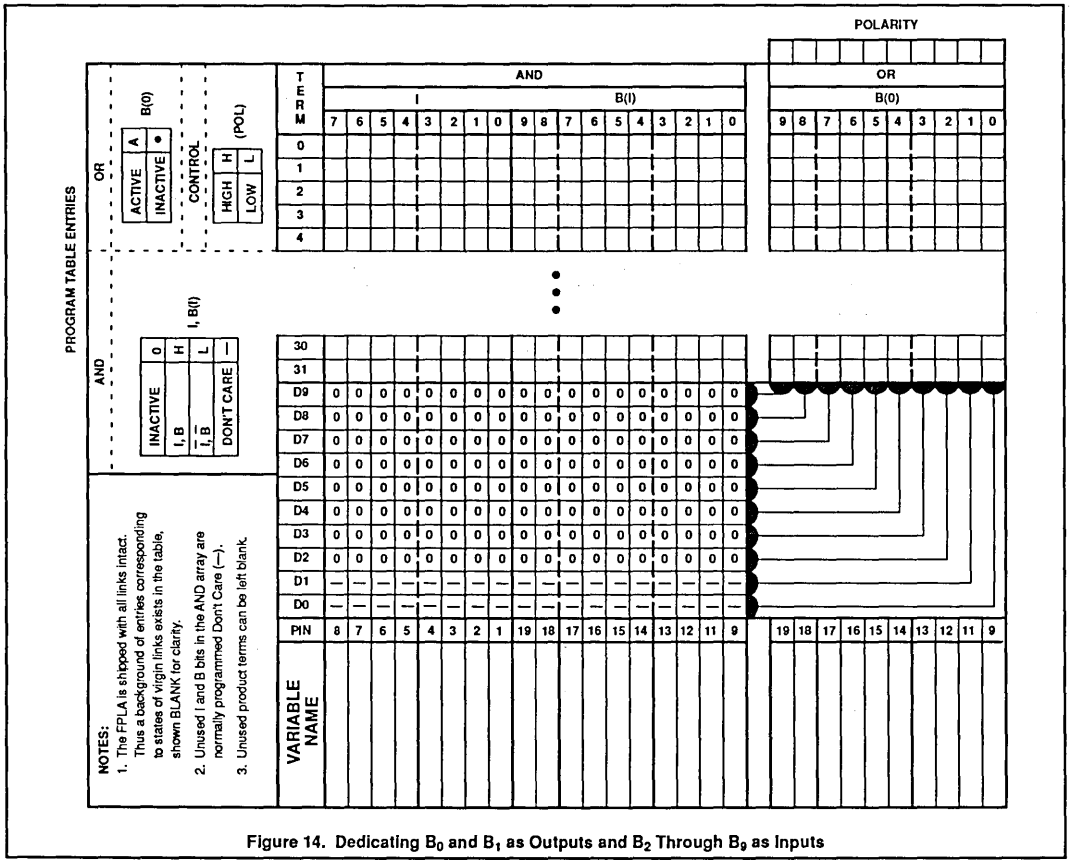


Figure 14. Dedicating B_0 and B_1 as Outputs and B_2 Through B_9 as Inputs

By placing a (—) Don't Care in each input box you are specifying that the True and Complement fuses are programmed on each Control Gate, thus permanently dedicating the B_0 and B_1 pins as outputs. By placing a (0) in all input boxes for B_2 - B_9, you are specifying that both True and Complement fuses are intact. This causes a low (0) to be forced on half of the Control Gate inputs, guaranteeing the output of the Control Gate will be low (0). When the Control Gate outputs are low (0), the 3-State buffer is inactive and the B_2 - B_9 pins are enabled as inputs. All B_X pin directions can be controlled in this manner.

ACTIVE DIRECTION CONTROL

Sometimes it is necessary to be able to actively change the direction of the B_X pins without permanently dedicating them. Some applications which require this include 3-State bus enable, multi-function decoding, etc. This can easily be done by programming the Control Gate to respond to one or more input pins. It is only necessary to select which I_X and B_X pins will control the pin directions and the active level HIGH (H) or LOW (L) that will be used. The PLUS153 Program Table in Figure 15 shows the method of controlling $B_0 - B_9$ with I_7. When I_7 is LOW (L), pins $B_0 - B_9$ are outputs; when I_7 is HIGH (H), pins $B_0 - B_9$ are inputs. Note that by programming all other I_X and B_X pins as DON'T CARE (—), they are permanently disconnected from control of B_X pin direction.

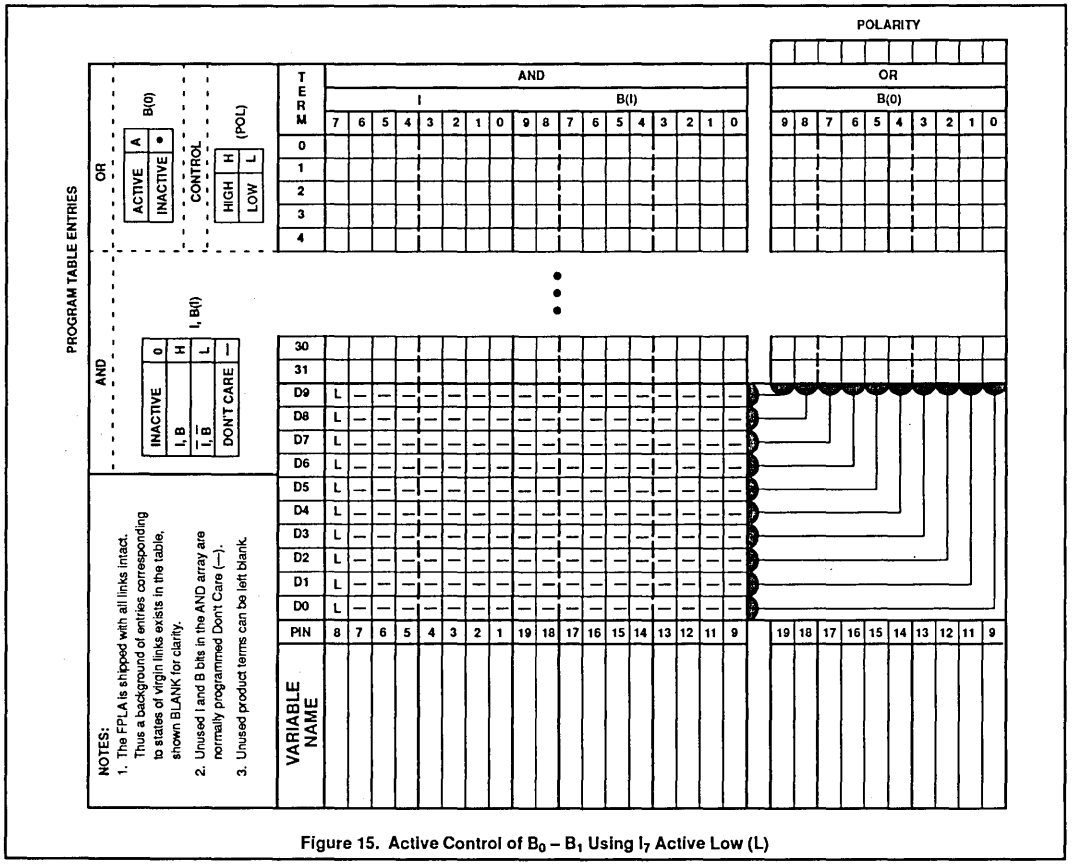


Figure 15. Active Control of $B_0 - B_1$ Using I_7 Active Low (L)

The previous 28-pin logic synthesis example could be done on the PLUS153 as follows:

$X_0 = AB + \overline{C}D + B\overline{D}$

$\overline{X}_1 = \overline{A}B + \overline{C}D + EFG$

Note that B_0 was used as a CHANGE input. When B_0 is HIGH (H) the outputs appear on B_8 and B_9. When B_0 is LOW (L), the outputs appear on B_6 and B_7. B_1 through B_5 are not used and therefore left unprogrammed.

Signetics offers two packages for user-friendly design assistance. The first package, AMAZE, has evolved over 10 years to support Signetics programmable products with logic equation, state equation, and schematic entry. AMAZE can compile designs quite well for Signetics lower density parts. However, to satisfy the needs of Programmable Macro Logic users, Signetics developed an additional software package called SNAP. SNAP expands upon the capabilities of AMAZE in its approach to design implementation, more closely resembling a gate array methodology. Both of these products are described in more depth at a later point in this handbook.

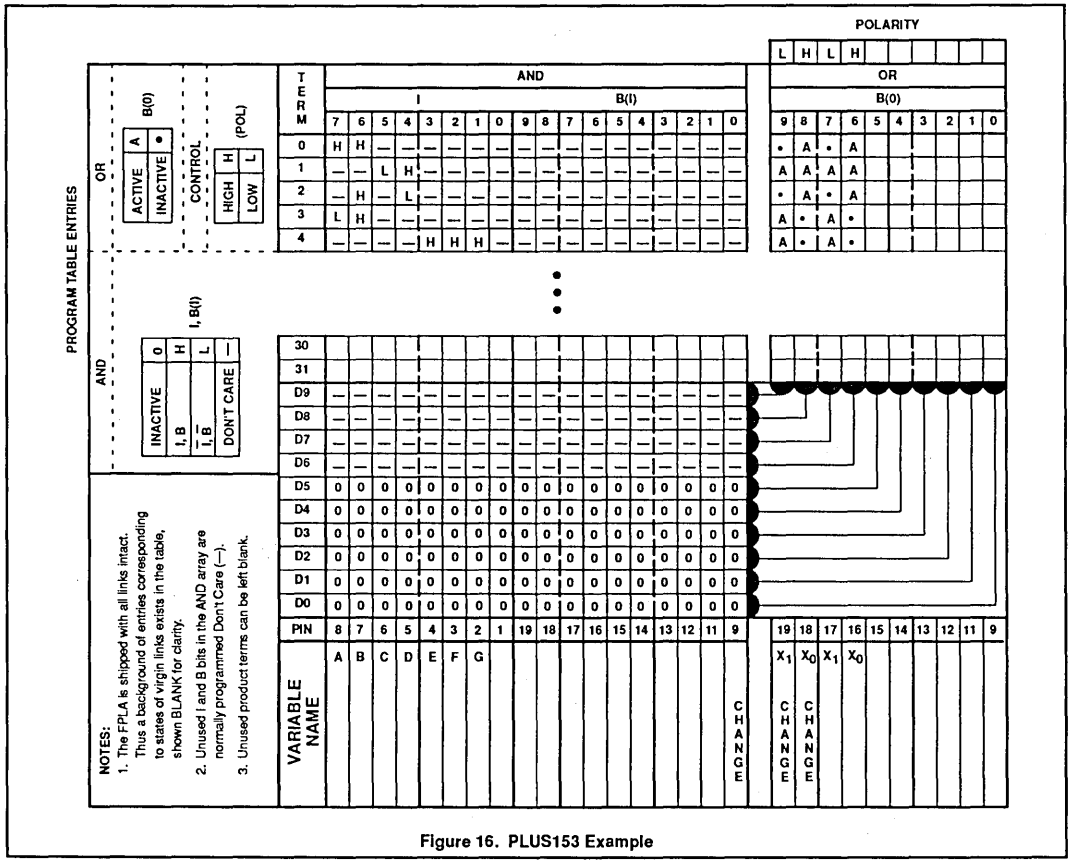


Figure 16. PLUS153 Example

SEQUENTIAL LOGIC CONSIDERATIONS

The PLUS405, PLUS105 and PLC42VA12 represent significant increases in complexity when compared to the combinatorial logic devices previously discussed. By combining the AND/OR combinatorial logic with clock output flip-flops and appropriate feedback, Signetics has created the first family of totally flexible sequential logic machines.

The PLUS405 (Programmable Logic Sequencer) is an example of a high-order machine whose applications are many. Application areas for this device include VRAM, DRAM, Bus and LAN control. The PLUS405 is fully capable of performing fast sequential operations in relatively high-speed processor systems. By placing repetitive sequential operations on the PLUS405, processor overhead is reduced.

The following pages summarize the PLUS405 architecture and features.

Sequencer Architecture

The PLUS405 Logic Sequencer is a programmable state machine, in which the output is a function of the present state and the present input.

With the PLUS405, a user can program any logic sequence expressed as a series of jumps between stable states, triggered by a valid input condition (I) at clock time (t). All stable states are stored in the State Register. The logic output of the machine is also programmable, and is stored in the Output Register. The PLUS105 is a subset of the PLUS405.

Clocked Sequence

A synchronous logic sequence can be represented as a group of circles interconnected with arrows. The circles represent stable states, labeled with an arbitrary numerical code (binary, hex, etc.) corresponding to discrete states of a suitable register. The arrows represent state transitions, labeled with symbols denoting the jump condition and the required **change** in output. The number of states in the sequence depends on the length and complexity of the desired algorithm.

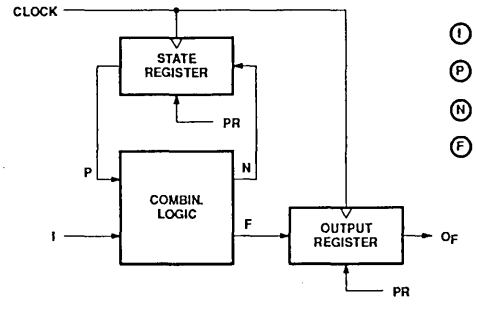


Figure 17. Basic Architecture of PLS105 FPLS. I, P, N, and F are Multi-line Paths Denoting Groups of Binary Variables Programmed by the User.

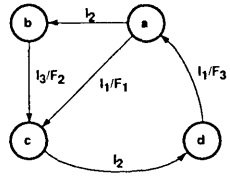


Figure 18. Typical State Diagram. I_{1-3} Are Jump Conditions Which Must Be Satisfied Before Any Transitions Take Place. F_r Are Changes in Output Triggered by I_m, and Stored in the Output Register. State Transitions a → b and c → d Involve No Output Change.

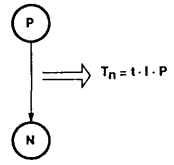


Figure 19. Typical State Transition Between Any Two States of Figure 18. The Arrow Connecting the Two States Gives Rise to a Transition Term T_n. I is the Jump Condition.

State Jumps

The state from which a jump originates is referred to as the Present state (P), and the state to which a jump terminates is defined as the Next state (N). A state jump always causes a change in state, but may or may not cause a change in machine output (F).

State jumps can occur only via "transition terms" T_n. These are logical AND functions of the clock (t), the Present state (P), and a valid input (I). Since the clock is actually applied to the State Register, $T_n = I \bullet P$. When T_n is "true", a control signal is generated and used at clock time (t) to force the contents of the State Register from (P) to (N), and to change the contents of the Output Register (if necessary). The simple state jump in Figure 20, involving 2 inputs, 1 state bit, and 1 output bit, illustrates the equivalence of discrete and programmable logic implementations.

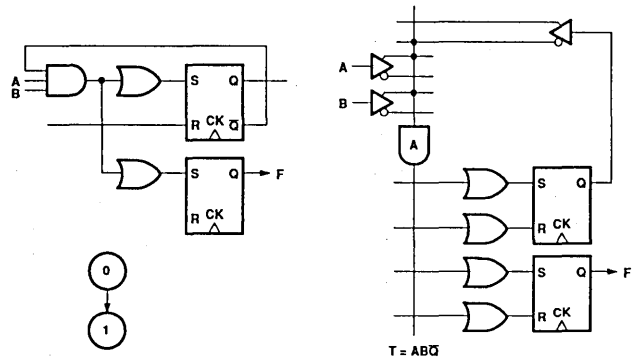


Figure 20. Typical State Jump From State (0) to State (1), If Inputs A = B = "1". The Jump Also Forces F = "1", as Required.

Sequencer Logic Structure

The Sequencer consists of programmable AND and OR gate arrays which control the Set and Reset inputs of a State Register, as well as monitor its output via an internal feedback path. The arrays also control an independent Output Register, added to store output commands generated during state transitions, and to hold the output constant during state sequences involving no output changes. If desired, any number of bits of the Output Register can be used to extend the width of the State Register, via external feedback.

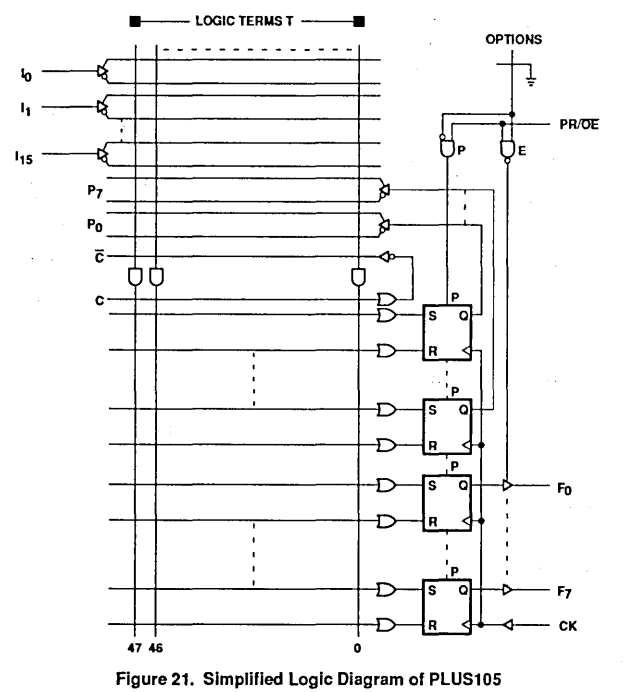


Figure 21. Simplified Logic Diagram of PLUS105

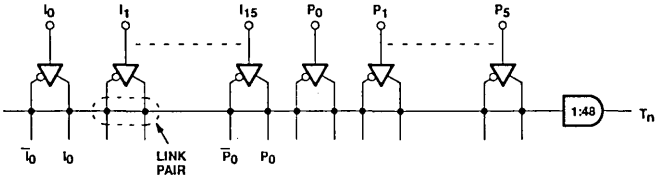


Figure 22. Typical AND Gate Coupled to (I) and (P) Inputs. If at Least One Link Pair Remains Intact, T_n is Unconditionally Forced Low.

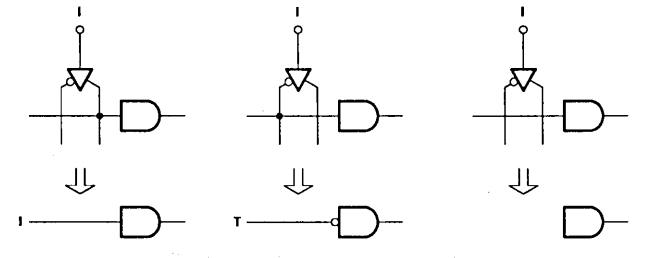


Figure 23. Choice of Input Polarity Coupling to a Typical AND Gate. With Both Links Open, (I) is Logically Don't Care.

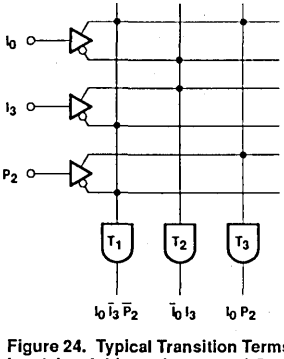


Figure 24. Typical Transition Terms Involving Arbitrary Inputs and State Variables. All Remaining Gate Inputs Are Programmed Don't Care. Note That T_2 Output is State Independent.

Input Buffers

16 external inputs (I_m) and 6 internal inputs (P_s), fed back from the State Register, are combined in the AND array through two sets of True/Complement (T/C) buffers. There are a total of 22 T/C buffers, all connected to multi-input AND gates via fusible links which are initially intact.

Selective fusing of these links allows coupling either True, Complement, or Don't Care values of (I_m) and (P_s).

"AND" Array

State jumps and output changes are triggered at clock time by valid transition terms T_n. These are logical AND functions of the present state (P) and the present input (I).

The PLUS105 AND Array contains a total of 48 AND gates. Each gate has 45 inputs – 44 connected to 22 T/C input buffers, and 1 dedicated to the Complement Array. The outputs of all AND gates are propagated through the OR Array, and used at clock time (t) to force the contents of the State Register from (P) to (N). they are also used to control the Output Register, so that the FPLS 8-bit output F_f is a function of the inputs and the present state. The PLUS405 contains 64 AND gates in its' AND array.

"OR" Array

In general, a clocked sequence will consist of several stable states and transitions, as determined by the complexity of the desired algorithm. All state and output changes in the state diagram imply changes in the contents of State and Output Registers.

Thus, each flip-flop in both registers may need to be conditionally set or reset several times with T_n commands. This is accomplished by selectively ORing through a programmable OR Array all AND gate outputs T_n necessary to activate the proper flip-flop control inputs.

The PLUS105 OR Array consists of 14 pairs of OR gates, controlling the S/R inputs of 14 State and Output Register stages, and a single NOR gate for the Complement Array. All gates have 48 inputs for connecting to all 48 AND gates. The PLUS405 uses 64 input gates.

The PLUS405 contains 16 pairs of OR gates controlling state transitions and output stages and two additional NOR gates for dual complement arrays.

Complement Array

The COMPLEMENT Array provides an asynchronous feedback path from the OR Array back to the AND Array.

This structure enables the sequencer to perform both direct and complement sequential state jumps with a minimum of transition (AND) terms.

Typically direct jumps, such as T_1 and T_2 in Figure 27 require only a single AND gate each.

But a complement jump such as T_3 generally requires many AND gates if implemented as a direct jump. However, by using the Complement Array, the logic requirements for this type of jump can be handled with just one more gate from the AND Array. Because it can be split into separate machines (2 clocks), the PLUS405 incorporates two COMPLEMENT Arrays.

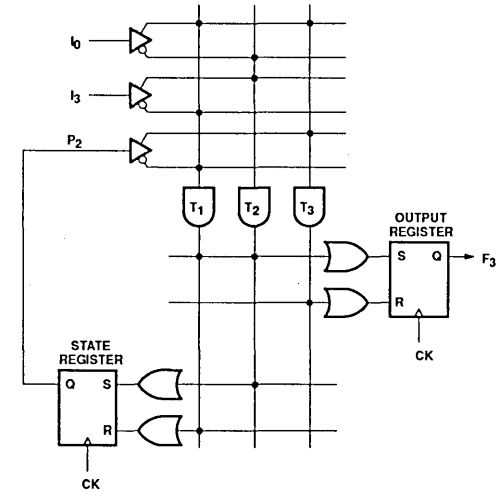


Figure 25. Typical OR Array Gating of Transition Terms $T_{1,2,3}$ Controlling Arbitrary State and Output Register Stages.

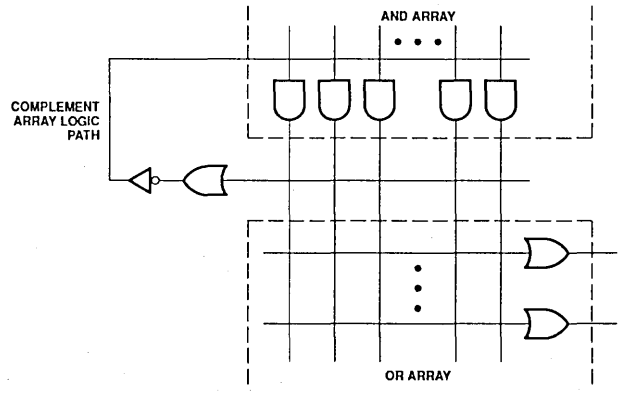
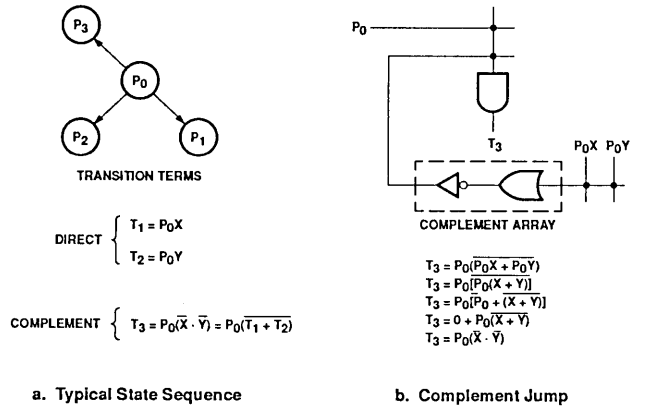


Figure 26. The COMPLEMENT Array is Logically Constructed from a multiple Input Programmable NOR Gate. All AND Terms Coupled to the OR Gate are Complemented at the Inverter Output, and Can be Fed Back as Inputs to the AND Array.



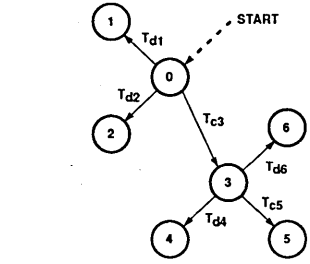
a. Typical State Sequence b. Complement Jump

Figure 27.

a. X and Y Specify the Conditional Logic for Direct Jump Transition Terms T_1 and T_2. The Complement Jump Term T_3 is True Only When Both T_1 and T_2 are False.

b. Note that the Complementary Logic Expression for T_3, $\overline{T_1 + T_2}$, Corresponds Exactly to the Logic Structure of the Complement Array.

As indicated in Figure 28, the single Complement Array gate may be used for many states of the state diagram. This happens because all transition terms linked to the OR gate include the present state as a part of their conditional logic. In any particular state, only those transition terms which are a function of that state are enabled; all other terms coupled to different states are disabled and do not influence the output of the Complement Array. As a general rule of thumb, the Complement Array can be used as many times as there are states.



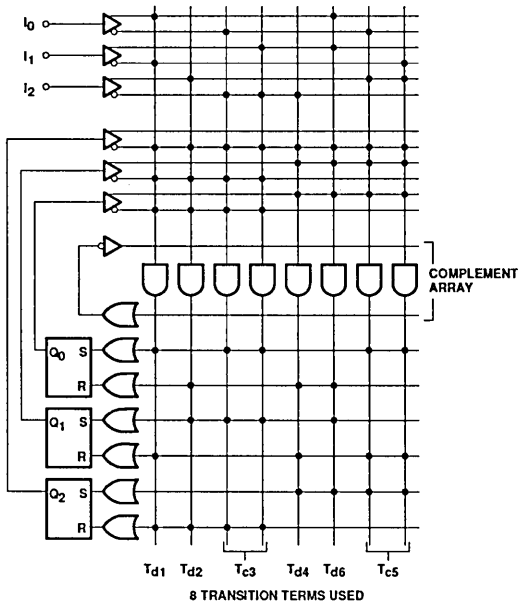
a. State Diagram

$T_{d1} = I_0 I_1 P_0$
$T_{d2} = \overline{I_2} P_0$
$T_{c3} = \overline{(T_{d1} + T_{d2})} P_0 = \overline{(\overline{I_0} \overline{I_1} + \overline{I_2})} P_0$

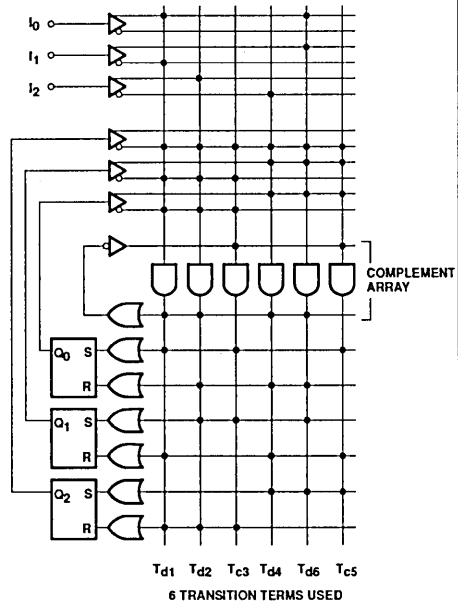
$T_{d4} = I_2 P_3$
$T_{d6} = \overline{I_0 I_1} P_0$
$T_{c5} = \overline{(T_{d4} + T_{d6})} P_3 = \overline{(\overline{I_0} \overline{I_1} + \overline{I_2})} P_3$

T_{cn} = COMPLEMENT STATE TRANSITION TERM
T_{dn} = DIRECT STATE TRANSITION TERM
P_a = PRESENT STATE

b. Logic Definition



c. State Logic without Using the Complement Array



d. State Logic Using the Complement Array

Figure 28. Logic Reduction with the Complement Array. The Logic State Diagram in (a) Includes Complement Jumps T_{C3} and T_{C5} Defined in (b). When Using the Complement Array, a Savings of 2 Transition Terms Results, as Shown in (c) and (d).

Additional features are available depending on a specific part. In particular, the PLC42VA12 has everything mentioned here, and more. More details on PLAs, PAL-Type devices and Sequencers can be found in the application section later in the manual.

Programmable Macro Logic, Signetics very high density logic is fully described in detail in its own section.

Quality and Reliability

Programmable Logic Devices

SIGNETICS PROGRAMMABLE LOGIC QUALITY

Signetics has put together winning processes for manufacturing Programmable Logic. Our standard is zero defects, and current customer quality statistics demonstrate our commitment to this goal.

The PLDs produced in the Standard Products Group must meet rigid criteria as defined by our design rules and as evaluated with a thorough product characterization and quality process. The capabilities of our manufacturing process are measured and the results evaluated and reported through our corporate-wide QA05 data base system. The SURE (Systematic Uniform Reliability Evaluation) program monitors the performance of our product in a variety of accelerated environmental stress conditions. All of these programs and systems are intended to prevent product-related problems and to inform our customers and employees of our progress in achieving zero defects.

RELIABILITY BEGINS WITH THE DESIGN

Quality and reliability must begin with design. No amount of extra testing or inspection will produce reliable ICs from a design that is inherently unreliable. Signetics follows very strict design and layout practices with its circuits. To eliminate the possibility of metal migration, current density in any path cannot exceed 2×10^5 amps/cm^2. Layout rules are followed to minimize the possibility of shorts, circuit anomalies, and SCR type latch-up effects. Numerous ground-to-substrate connections are required to ensure that the entire chip is at the same ground potential, thereby precluding internal noise problems.

PRODUCT CHARACTERIZATION

Before a new design is released, the characterization phase is completed to insure that the distribution of parameters resulting from lot-to-lot variations is well within specified limits. Such extensive characterization data also provides a basis for identifying unique application-related problems which are not part of normal data sheet guarantees. Characterization takes place from –55°C to +125°C and at ± 10% supply voltage.

QUALIFICATION

Formal qualification procedures are required for all new or changed products, processes and facilities. These procedures ensure the high level of product reliability our customers expect. New facilities are qualified by corporate groups as well as by the quality organizations of specific units that will operate in the facility. After qualification, products manufactured by the new facility are subjected to highly accelerated environmental stresses to ensure that they can meet rigorous failure rate requirements. New or changed processes are similarly qualified.

QA05 — QUALITY DATA BASE REPORTING SYSTEM

The QA05 data reporting system collects the results of product assurance testing on all finished lots and feeds this data back to concerned organizations where appropriate action can be taken. The QA05 reports EPQ (Estimated Process Quality) and AOQ (Average Outgoing Quality) results for electrical, visual/mechanical, hermeticity, and documentation audits. Data from this system is available upon request.

THE SURE PROGRAM

The SURE (Systematic Uniform Reliability Evaluation) program audits/monitors products from all Signetics' divisions under a variety of accelerated environmental stress conditions. This program, first introduced in 1964, has evolved to suit changing product complexities and performance requirements.

The SURE program has two major functions: Long-term accelerated stress performance audit and a short-term accelerated stress monitor. In the case of Bipolar Memory and Programmable Logic products, samples are selected that represent all generic product groups in all wafer fabrication and assembly locations.

THE LONG-TERM AUDIT

One-hundred devices from each generic family are subjected to each of the following stresses every eight weeks:

- High Temperature Operating Life: T_J = 150°C, 1000 hours, static biased or dynamic operation, as appropriate (worst case bias configuration is chosen)
- High Temperature Storage: T_J = 150°C, 1000 hours
- Temperature Humidity Biased Life: 85°C, 85% relative humidity, 1000 hours, static biased
- Temperature Cycling (Air-to-Air): –65°C to +150°C, 1000 cycles

THE SHORT-TERM MONITOR

Every other week a 50-piece sample from each generic family is run to 168 hours of pressure pot (15psig, 121°C, 100% saturated steam).

In addition, each Signetics assembly plant performs SURE product monitor stresses weekly on each generic family and molded package by pin count and frame type. Fifty-piece samples are run on each stress, pressure pot to 96 hours, thermal shock to 300 cycles.

SURE REPORTS

The data from these test matrices provides a basic understanding of product capability, an indication of major failure mechanisms and an estimated failure rate resulting from each stress. This data is compiled periodically and is available to customers upon request.

Many customers use this information in lieu of running their own qualification tests, thereby eliminating time-consuming and costly additional testing.

Quality and Reliability

RELIABILITY ENGINEERING

In addition to the product performance monitors encompassed in the Programmable Logic SURE program, Signetics' Corporate and Division Reliability Engineering departments sustain a broad range of evaluation and qualification activities.

Included in the engineering process are:

- Evaluation and qualification of new or changed materials, assembly/wafer-fab processes and equipment, product designs, facilities and subcontractors
- Device or generic group failure rate studies
- Advanced environmental stress development
- Failure mechanism characterization and corrective action/prevention reporting

The environmental stresses utilized in the engineering programs are similar to those utilized for the SURE monitor; however, more highly-accelerated conditions and extended durations typify the engineering projects. Additional stress systems such as biased pressure pot, power-temperature cycling, and cycle-biased temperature-humidity, are also included in the evaluation programs.

FAILURE ANALYSIS

The SURE Program and the Reliability Engineering Program both include failure analysis activities and are complemented by corporate, divisional and plant failure analysis departments. These engineering units provide a service to our customers who desire detailed failure analysis support, who in turn provide Signetics with the technical understanding of the failure modes and mechanisms actually experienced in service. This information is essential in our ongoing effort to accelerate and improve our understanding of product failure mechanisms and their prevention.

ZERO DEFECTS PROGRAM

In recent years, United States industry has increasingly demanded improved product quality. We at Signetics believe that the customer has every right to expect quality products from a supplier. The benefits which are derived from quality products can be summed up in the words, *lower cost of ownership.*

Those of you who invest in costly test equipment and engineering to assure that incoming products meet your specifications have a special understanding of the cost of ownership. And your cost does not end there; you are also burdened with inflated inventories, lengthened lead times and more rework.

SIGNETICS UNDERSTANDS CUSTOMERS' NEEDS

Signetics has long had an organization of quality professionals, inside all operating units, coordinated by a corporate quality department. This broad decentralized organization provides leadership, feedback, and direction fo achieving a high level of quality.

In 1980 we recognized that in order to achieve outgoing levels on the order of 100ppm (parts per million), down from an industry practice of 10,000ppm, we needed to supplement our traditional quality programs with one that encompassed all activities and all levels of the company. Such unprecedented low defect levels could only be achieved by contributions from all employees, from the R and D laboratory to the shipping dock. In short, from a program that would effect a total cultural change within Signetics in our attitude toward quality.

QUALITY PAYS OFF FOR OUR CUSTOMERS

Signetics' dedicated programs in product quality improvement, supplemented by close working relationships with many of our customers, have improved outgoing product quality more than twenty-fold since 1980. Today, many major customers no longer test Signetics circuits. Incoming product moves directly from the receiving dock to the production line, greatly accelerating throughput and reducing inventories. Other customers have pared significantly the amount of sampling done on our products. Others are beginning to adopt these cost-saving practices.

We closely monitor the electrical, visual, and mechanical quality of all our products and review each return to find and correct the cause. Since 1981, over 90% of our customers report a significant improvement in overall quality (see Figure 1).

At Signetics, quality means more than working circuits. It means on-time delivery of the right product at the agreed-upon price. Signetics considers Performance to Customer Request and Performance to Original Schedule Date to be key Quality issues. Employees treat delinquencies as quality defects. They analyze the cause for the delinquency and seek corrective action to prevent future occurrence. Continuous effort is given to try to achieve the ultimate goal of zero delinquencies.

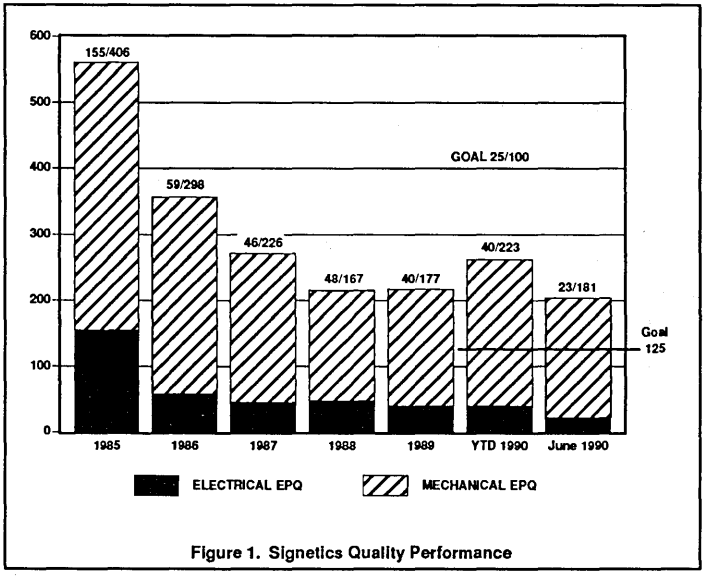


Figure 1. Signetics Quality Performance

Quality and Reliability

ONGOING QUALITY PROGRAM

The quality improvement program at Signetics is based on "Do It Right the First Time". The intent of this innovative program is to change the perception of Signetics' employees that somehow quality is solely a manufacturing issue where some level of defects is inevitable. this attitude has been replaced by one of acceptance of the fact that all errors and defects are preventable, a point of view shared by all technical and administrative functions equally.

This program extends into every area of the company, and more than 40 quality improvement teams throughout the organization drive its ongoing refinement and progress.

Key components of the program are the Quality College, the "Make Certain" Program, Corrective Action Teams, and the Error Cause Removal System.

The core concepts of doing it right the first time are embodied in the four absolutes of quality:

1. The definition of quality is conformance to requirements.
2. The system to achieve quality improvement is prevention.
3. The performance standard is zero defects.
4. The measurement system is continuous improvement.

"MAKING CERTAIN" — ADMINISTRATIVE QUALITY IMPROVEMENT

Signetics' experience has shown that the largest source of errors affecting product and service quality is found in paperwork and in other administrative functions. The "Make Certain" program focuses the attention of management and administrative personnel on error prevention, beginning with each employee's own actions.

This program promotes defect prevention in three ways: by educating employees as to the impact and cost of administrative errors, by changing attitudes from accepting occasional errors to one of accepting a personal work standard of zero defects, and by providing a formal mechanism for the prevention of errors.

CORRECTIVE ACTION TEAMS

Employees with the perspective, knowledge, and necessary skills to solve a problem are formed into ad hoc groups called Corrective Action Teams. These teams, a major force within the company for quality improvement, resolve administrative, technical and manufacturing issues.

ECR SYSTEM (ERROR CAUSE REMOVAL)

The ECR System permits employees to report to management any impediments to doing the job right the first time. Once such an impediment is reported, management is obliged to respond promptly with a corrective program. Doing it right the first time in all company activities produces lower cost of ownership through defect prevention.

PRODUCT QUALITY PROGRAM

To reduce defects in outgoing products, we created the Product Quality Program. This is managed by the Product Engineering Council, composed of the top product engineering and test professionals in the company. this group:

1. Sets aggressive product quality improvement goals;
2. provides corporate-level visibility and focus on problem areas;
3. serves as a corporate resource for any group requiring assistance in quality improvement; and
4. drives quality improvement projects.

As a result of this aggressive program, every major customer who reports back to us on product performance is reporting significant progress.

VENDOR CERTIFICATION PROGRAM

Our vendors are taking ownership of their own product quality by establishing improved process control and inspection systems. They subscribe to the zero defects philosophy. Progress has been excellent.

Through intensive work with vendors, we have improved our lot acceptance rate on incoming materials as shown in Figure 2. Simultaneously, waivers of incoming material have been eliminated.

QUALITY AND RELIABILITY ORGANIZATION

Quality and reliability professionals at the divisional level are involved with all aspects of the product, from design through every step in the manufacturing process, and provide product assurance testing of outgoing product. A separate corporate-level group provides direction and common facilities.

Quality and Reliability Functions

- Manufacturing quality control
- Product assurance testing
- Laboratory facilities – failure analysis, chemical, metallurgy, thin film, oxides
- Environmental stress testing
- Quality and reliability engineering
- Customer liaison

COMMUNICATING WITH EACH OTHER

For information on Signetics' quality programs or for any question concerning product quality, the field salesperson in your area will provide you with the quickest access to answers. Or, write on your letterhead directly to the corporate VP of quality at the corporate address shown at the back of this manual.

We are dedicated to preventing defects. When product problems do occur, we want to know about them so we can eliminate their causes. Here are some ways we can help each other:

- Provide us with one informed contact within your organization. This will establish continuity and build confidence levels.
- Periodic face-to-face exchanges of data and quality improvement ideas between your engineers and ours can help prevent problems before they occur.
- Test correlation data is very useful. Line-pull information and field failure reports also help us improve product performance.
- Provide us with as much specific data on the problem as soon as possible to speed analysis and enable us to take corrective action.
- An advance sample of the devices in question can start us on the problem resolution before physical return of shipment.

This teamwork with you will allow us to achieve our mutual goal of improved product quality.

Quality and Reliability

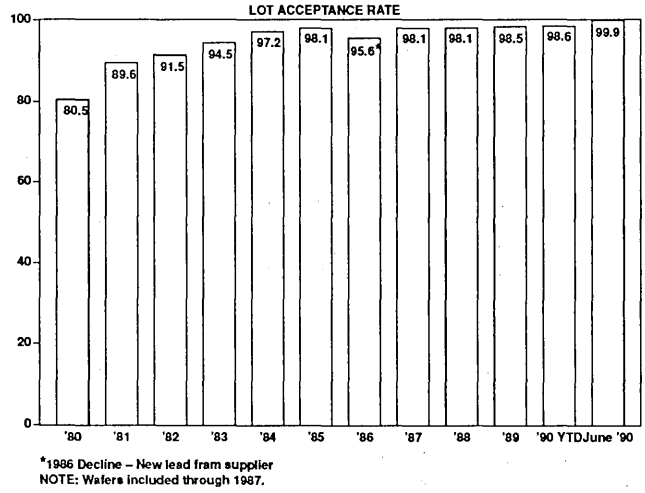


Figure 2. Incoming Direct Materials Quality

MANUFACTURING: DOING IT RIGHT THE FIRST TIME

In dealing with the standard manufacturing flows, it was recognized that significant improvement would be achieved by "doing every job right the first time", a key concept of the quality improvement program. During development of the program many profound changes were made. Figure 3, *Programmable Logic Process Flow,* shows the result. Key changes included such things as implementing 100% temperature testing on all products as well as upgrading test handlers to insure 100% positive binning. Some of the other changes and additions were to tighten the outgoing QA lot acceptance criteria to the tightest in the industry, with zero defect lot acceptance sampling across all three temperatures.

The achievements resulting from the improved process flow have helped Signetics to be recognized as the leading Quality supplier of Programmable Logic. These achievements have also led to our participation in several Ship-to-Stock programs, which our customers use to eliminate incoming inspection. such programs reduce the user *cost of ownership* by saving both time and money.

OUR GOAL: 100% PROGRAMMING YIELD

Our original goal back in the early 1970s was to develop a broad line of programmable products which would be recognized as having the best programming yield in the industry. Within the framework of a formal quality program, our efforts to improve circuit designs and refine manufacturing controls have resulted in major advances toward that goal.

Also within the framework of our formal quality program we have now established a stated goal of 100% programming yield. through the increasing effectiveness of a quality attitude of "Do It Right the First Time" we're moving ever closer to that target.

Signetics PLD programming yields have been shown in collected data from internal audits and customer reporting to be consistently higher than comparable devices produced by our competition. We use systematic methods involving publication of exacting specifications of our programming algorithms, and through evaluation of those algorithms as implemented in industry standard programming equipment. Because of this we can assure our customers who program Signetics PLDs on such qualified equipment they will see consistently high yields. Our data base shows that average lot programming yield exceeds 97%.

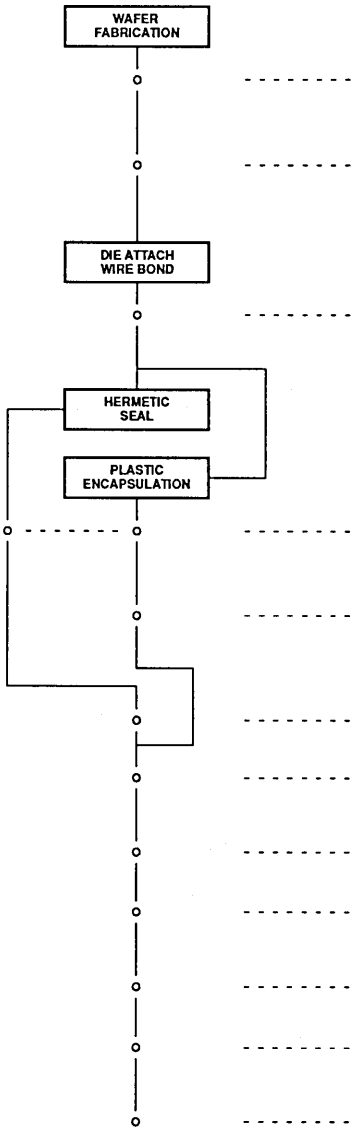
As time goes on the drive for a product line that has Zero Defects will grow in intensity. These efforts will provide both Signetics and our customers with the ability to achieve the mutual goal of improved product quality.

The Customer Specific Quality Assurance department has monitored PPM progress. We are pleased with the progress that has been made, and expect to achieve even more impressive results as the procedures for accomplishing these tasks are fine tuned.

The Customer Specific Reliability Department has established an ongoing Infant Mortality Monitor. This monitor is used to determine and drive ongoing Corrective Action for the purposes of continuously improving product reliability.

The real measure of any quality improvement program is the result that our customers see. The meaning of *Quality* is more than just working circuits. It means commitment to *On Time Delivery* at the *Right Place* of the *Right Quantity* of the *Right Product* at the *Agreed Upon Price.*

Quality and Reliability



SCANNING ELECTRON MICROSCOPE CONTROL

Wafers are sampled daily by the Quality Control Laboratory from each fabrication area and subjected to SEM analysis. This process control reveals manufacturing defects such as contact and oxide step coverage in the metalization process which may result in early failures.

DIE SORT VISUAL ACCEPTANCE

Product is inspected for defects caused during fabrication, wafer testing, or the mechanical scribe and break operation. Defects such as scratches, smears and glassivated bonding pads are included in the lot acceptance criteria.

PRE-SEAL VISUAL ACCEPTANCE

Product is inspected to detect any damage incurred at the die attach and wire bonding stations. Defects such as scratches, contamination and smeared ball bonds are included in the lot acceptance criteria.

STABILIZATION BAKE PRECONDITIONING

Plastic molded devices are baked to stress wire and die bonds and help eliminate marginal devices. It also ensures an optimum plastic seal to enhance moisture resistance. Hermetic devices are baked to ensure the elimination of any remaining cavity moisture.

TEMPERATURE CYCLING (Monitor)

Using MIL–STD–883 Method 1010.1 Condition C devices are cycled from cold to hot temperatures imposing a very severe stress which weakens loose bonds and package seals without degrading good devices. A loosened bond is later detected during 100% electrical tests.

SEAL TESTS

Hermetic package seal integrity is ensured by 100% gross leak testing.

100% PRODUCTION ELECTRICAL TEST

Every device is tested for functional and AC/DC parameters at high temperature with guard bands to assure performance to data sheet limits.

BURN IN (SUPR II LEVEL B OPTION)

Devices are burned in for 21 hours at 155°C maximum junction temperature.

SYMBOL

Devices are marked with the Signetics logo, device number and the date code or custom symbol per individual specification requirements.

VISUAL

All products are 100% visually inspected per the requirements specified in Signetics or customer documents.

100% PRODUCTION ELECTRICAL TESTING

Every device is tested for functional and AC/DC parameters at 25°C with guard bands to assure performance to data sheet limits.

FINAL QUALITY ASSURANCE GATE

The final QA inspection step guarantees that the specified mechanical and electrical tests have met Signetics Zero Defect Requirements. Every shipment is sealed and identified by QA personnel.

Figure 3. Customer Specific Products Programmable Logic Process Flow

Quality and Reliability

CMOS RELIABILITY INFORMATION

All Signetics' EPROM die are designed as low power UV light erasable and electrically programmable read only memories. They have been designed to perform over military and commercial temperature ranges. These die are assembled in EPROM packages that comply with industry standard packages: CERDIP (Quartz window), Plastic DIP (One Time Programmable) and Plastic Leaded Chip Carrier (One Time Programmable).

The following descriptions are of the tests and calculations performed on each device organization and package type to validate the quality and reliability of the CMOS design and technology. All described tests are performed on each package type, with the exception of the 'Program-erase cycling' test for the One Time Programmable devices.

ELECTROSTATIC DISCHARGE PROTECTION (ESD)

This test is performed to validate the product's tolerance to electrostatic discharge damage.

Both MIL–STD–883 criteria (human body model) and mechanical model charged device test are performed.

HIGH TEMPERATURE STORAGE LIFE TEST (HTSL)

Another popular name for this test is data retention bake. This process is used to thermally accelerate charge loss from the floating gate. The test is performed by subjecting devices that contain a 100% programmed data pattern to a 250°C bake with no applied electrical bias or clocks.

In addition to charge loss, this test is used to detect mechanical reliability (i.e., bond integrity) and process instability.

DYNAMIC LOW TEMPERATURE LIFE TEST (DLTL)

This test is performed at –10°C to detect the effects of hot electron injection into the gate oxide as well as package-related failures (i.e., metal corrosion). The biasing and clocking conditions for this test are identical to the DHTL #1 test.

TEMPERATURE CYCLE (TMCL)

This test consists of performing 200 cycles of ambient air temperature of the chamber and housing the unbiased subject devices from –65°C to +150°C and back. The 200 cycles are performed at 20 minutes per cycle.

DYNAMIC HIGH TEMPERATURE LIFE TEST (DHTL #1)

This test is used to accelerate failure mechanisms by operating the devices at 125°C ambient temperature with worst-case specified power supply voltages of V_{CC} and V_{PP} at 5.5V. The memory is sequentially addressed to exercise the fully-loaded outputs. A checkerboard complement data pattern is used to simulate random patterns expected during actual use.

DYNAMIC HIGH TEMPERATURE LIFE TEST (DHTL #2)

This test is used to accelerate oxide breakdown failures and to further accelerate the failure mechanisms of DHTL #1. The test setup is identical to the one used for the DHTL #1 test except the temperature is 150°C and the V_{CC} and V_{PP} power supply voltages are 6.5V, resulting in a 20% increase over the specified operational electrical field across the gate oxides of the device (1.25mV/cm for 325Å oxide thickness). This represents a 55 × electrical field induced acceleration in addition to the thermal acceleration at 150°C.

PROGRAM-ERASE CYCLING AND PROGRAMMABILITY

All four power supply voltage combinations for V_{CC} and V_{PP} are tested for programmability (V_{CC} = 6.0V ± 0.25V and V_{PP} = 12.5V ± 0.5V in program mode). The number of possible program/erase cycles is then tested to establish program-erase cycling expectations.

FAILURE RATE PREDICTIONS

In preparation for the various life tests, a 168 hour, 125°C, 5.5V production burn-in is performed on the devices. The infant mortality rejects are removed from the population in order to develop long-term failure rate information during the random failure rate portion of the device life cycle.

The failure rate calculation combines all failure mechanisms by activation energies and associated device hours for the 125°C, 5.5V Dynamic Life Test (DHTL #1), the 150°C, 6.5V Dynamic Life Test (DHTL #2), the 150°C, 7.5V Static Life Test and the 250°C Bake.

The activation energies for the various EPROM failure mechanisms are:

Defective bit	0.6eV charge gain/loss (electron hopping conduction)
Oxide breakdown	0.3eV
Silicon defects	0.3eV
Contamination	1.0 – 1.2eV
Intrinsic charge loss	1.4eV

NOTE:
The combined failure rate for the stresses is the sum of failure rates by activation energies.

Quality and Reliability

METHODS OF FAILURE RATE CALCULATIONS

Actual Device Hours = Number of Devices × Number of Hours. In order to determine the Equivalent Hours derated to a given operation temperature, the junction temperatures of the devices should be calculated using the known thermal resistance of the package (θ_{JA}) and the power dissipation of the devices:

$$T_{1,2} = \theta_{JA}(IV)_{1,2} + T_{A1,2} \quad (1)$$

Using the Arrhenius relation, the test temperature and the derated operation temperature will yield the thermal acceleration factor from T_1 to T_2:

$$\frac{R_1}{R_2} = \frac{A \cdot \exp\left[\frac{E_A}{kT_1}\right]}{A \cdot \exp\left[\frac{E_A}{kT_2}\right]} = \exp\left[\frac{E_A}{k}\right]\left[\frac{1}{T_1} - \frac{1}{T_2}\right] \quad (2)$$

k = 8.617×10^{-5} eV/Kelvin (Boltzmann's constant)
A = Proportionality constant for a given failure mechanism
R_1 = mean time to failure @ T_1
R_2 = mean time to failure @ T_2
E_A = activation energy for the failure mechanism
T_1 = operating temperature
T_2 = life test temperature

An additional 55 × acceleration factor should be added for the 150°C/6.5V dynamic life test due to the time-dependent oxide failure acceleration (20% higher than specified power supply voltage).

Multiplying the actual device hours by the acceleration factor for each failure mechanism will result in the equivalent hours.

Poisson statistics are applied to estimate the performance of the population from the life test results of a sample test. This is useful when the probability of failures is small and the failures occur randomly in time. A commonly used formula for estimating the failure rate is the "chi-squared" equation:

$$F_C = \frac{\chi^2}{2nt} \times 100\% \quad (3)$$

F_C = calculated failure rate estimate (in %/1000 hrs) at upper confidence limit
χ^2 = "chi-squared" value for $2F_A$ + 2 degrees of freedom for $\propto$ where F_A is the number of actual failures (χ^2 comes from available tables for a known $\propto$)
$\propto$ = 1–B, where B is the confidence limit (B is stated in %).
n = number of units in test
t = test time in thousands of hours (equivalent)

Equation 3 will calculate the estimated failure rates/1000 hrs for 60% confidence level (industry standard) for each failure mechanism.

THE SURE PROGRAM

The SURE (Systematic Uniform Reliability Evaluation) program audits/monitors products from all Signetics' divisions under a variety of accelerated environmental stress conditions. this program, first introduced in 1964, has evolved to suit changing product complexities and performance requirements.

The SURE program has two major functions: long-term accelerated stress performance audit and a short-term accelerated stress monitor. In the case of Memory products, samples are selected that represent product groups from all wafer fabrication and assembly locations.

SURE REPORTS

The data from these test matrices provides a basic understanding of product capability, an indication of major failure mechanisms and an estimated failure rate resulting from each stress. This data is compiled periodically and is available to customers upon request.

Section 3
PAL®-Type Device Data Sheets

INDEX

Document No.	853–0960
ECN No.	97887
Date of Issue	October 16, 1989
Status	Product Specification
Programmable Logic Devices	

PLHS16L8A/B

Programmable AND array logic

(16 × 64 × 8)

DESCRIPTION

The PLHS16L8A is a high-speed "A" version, and the PLHS16L8B is a very high-speed "B" version PAL®-type device. The sum of products (AND-OR) architecture is comprised of 64 AND gates and 8 OR gates. The Signetics PLHS16L8A/B devices offer 100% functional compatibility with other PAL 16L8 devices. Specified at a t_{PD} of 20ns (maximum), the PLHS16L8A is 20% faster than other "A" version PAL 16L8 devices, and consumes 20% less power than most other "A" speed 16L8 devices. The PLHS16L8B, specified at 155mA I_{CC} (maximum), consumes 20% less power than other "B" version PAL 16L8 devices.

All AND gates are linked to 10 dedicated inputs, 6 bidirectional I/O and 2 dedicated outputs. On-chip buffers couple either true (I, B) or complement ($\overline{I}$, $\overline{B}$) input polarities to all AND gates. The 64 AND gates are separated into eight groups of eight product terms each. Within each group, seven of the AND terms are OR'ed together, while the eighth is used to control the 3-State function of the bidirectional I/O. All outputs (bidirectional and dedicated) are inverting.

In the virgin state, the AND array fuses are back-to-back CB-EB diode pairs which act as open connections. Current is avalanched across individual diode pairs during fusing, which essentially short circuits the EB diode and provides the connection for the associated product term.

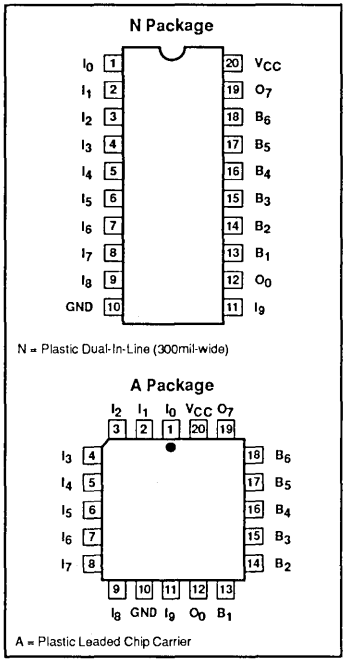
The PLHS16L8A/B is field-programmable, allowing the user to quickly generate custom patterns using standard programming equipment.

Order codes are listed in the Ordering Information Table.

FEATURES

- "A" version 100% functionally and pin-for-pin compatible with AmPAL16L8A, MMI PAL16L8A, TIBPAL16L8-25, and NSC PAL16L8A devices
  - 20% faster than other "A" version PAL devices
  - t_{PD} = 20ns (max)
- "B" version 100% functionally and pin-for-pin compatible with AmPAL16L8B, MMI PAL16L8B, TIBPAL16L8-15 and NSC PAL16L8B devices
  - Consumes 20% less power than other "B" version PAL devices
  - 155mA I_{CC} (worst case)
  - I/O propagation delay: 15ns (max) ("B" version)
- Field-programmable
- 10 dedicated inputs
- 8 outputs
  - 6 bidirectional I/O
  - 2 dedicated outputs
- Individual 3-State control of all outputs
- 64 AND gates/product terms
- Security fuse

PIN CONFIGURATIONS



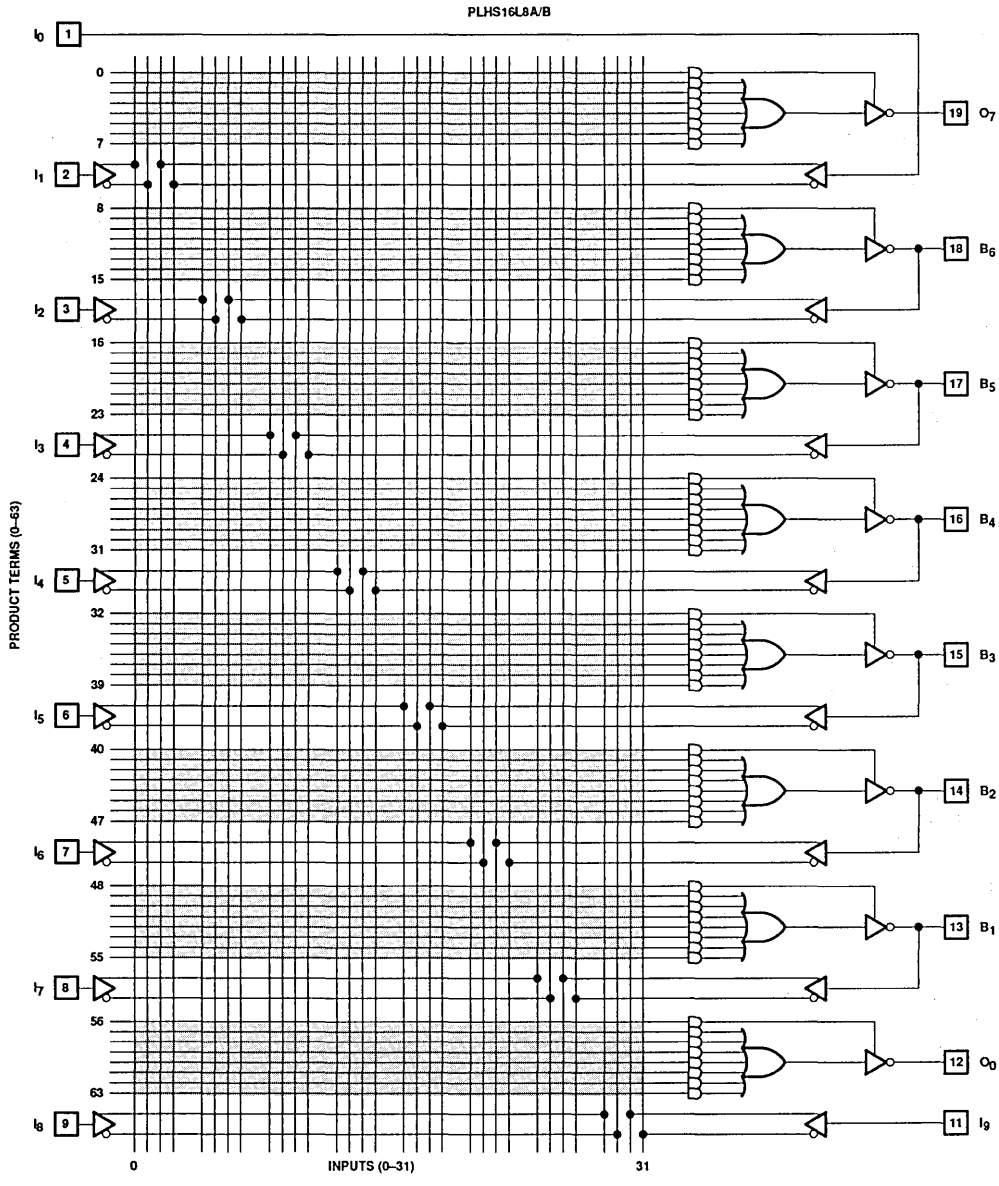
APPLICATIONS

- 100% functional replacement for 20-pin 16L8 combinatorial PAL devices
- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping/decoding
- Multiplexing

®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

Programmable AND array logic (16 × 64 × 8) PLHS16L8A/B

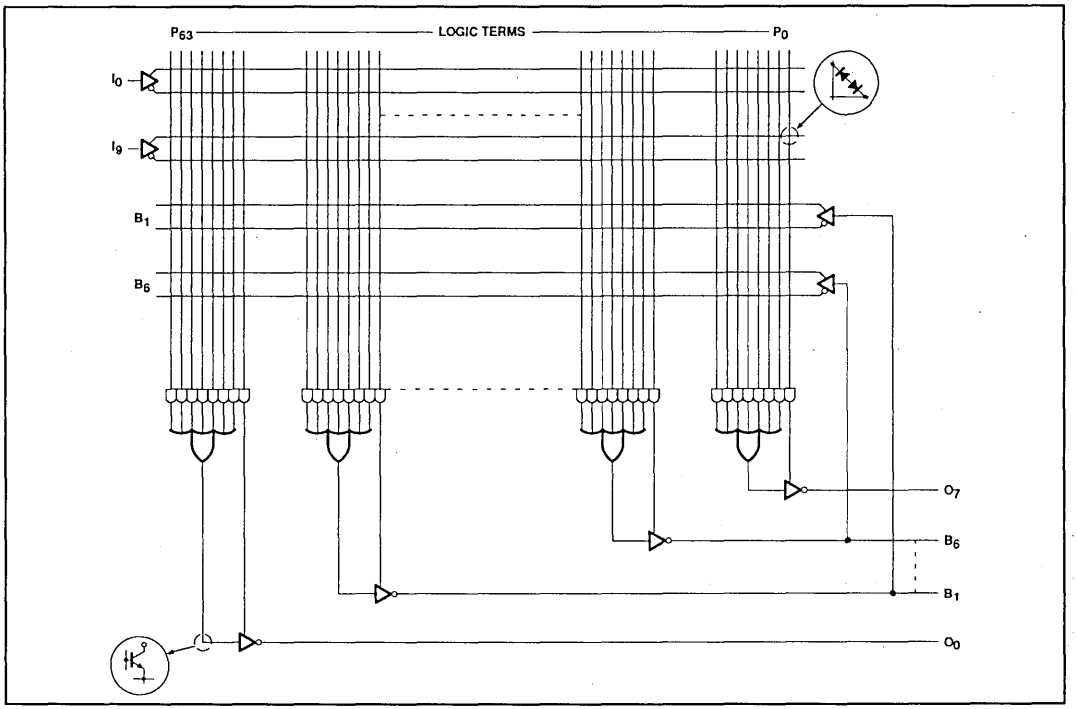
PLA LOGIC DIAGRAM



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "1".
2. ▒ Programmable connections.

Programmable AND array logic (16 × 64 × 8) PLHS16L8A/B

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic Dual In-Line (300mil-wide)	PLHS16L8AN, PLHS16L8BN
20-Pin Plastic Leaded Chip Carrier	PLHS16L8AA, PLHS16L8BA

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	−0.5 to +7	V_{DC}
V_{IN}	Input voltage	−0.5 to +5.5	V_{DC}
V_{OUT}	Output voltage	−0.5 to V_{CC} MAX	V_{DC}
V_{OUTPRG}	Output voltage (programming)	+21	V_{DC}
I_{IN}	Input current	−30 to +5	mA
I_{OUT}	Output current	+100	mA
I_{OUTPRG}	Output current (programming)	+170	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	−65 to +150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			+0.8	V
V_{IH}	High	V_{CC} = MAX	+2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.9	–1.2	V
Output voltage						
V_{OL}	Low	V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL} I_{OL} = +24mA			+0.50	V
V_{OH}	High	I_{OH} = –3.2mA	+2.4	+3.5		V
Input current						
I_{IL}	Low	V_{CC} = MAX V_{IN} = +0.40V		–20	–100	µA
I_{IH}	High	V_{IN} = +2.7V			+25	µA
I_I	High	V_{IN} = +5.5V			+1.0	mA
Output current						
I_{OZH}	Output leakage	V_{CC} = MAX, V_{IL} = 0.8V, V_{IH} = 2.0V V_{OUT} = +2.7V			+100	µA
I_{OZL}	Output leakage	V_{OUT} = +0.40V			–50	µA
I_{OS}	Short circuit [3]	V_{OUT} = +0.5V	–30	–60	–90	mA
I_{CC}	V_{CC} current	V_{CC} = MAX, All inputs = GND		100	155	mA
Capacitance[4]						
C_{IN}	Input	V_{CC} = +5V V_{IN} = 2.0V @ f = 1MHz		6		pF
C_{OUT}	I/O	V_{OUT} = 2.0V @ f = 1MHz		9		pF

NOTES:

1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
3. Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second. V_{OUT} = 0.5V has been chosen to avoid test problems caused by tester ground degradation.
4. These parameters are not 100% tested, but are periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 200\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS	LIMITS						UNIT
					PLHS16L8A			PLHS16L8B			
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PD}	Propagation delay	Output ±	Input ±	C_L = 50pF		14	20		12	15	ns
t_{EA}[3]	Output enable	Output –	Input ±	C_L = 50pF		14	20		12	15	ns
t_{ER}[3]	Output disable	Output +	Input ±	C_L = 5pF		14	20		12	15	ns

NOTES:
1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. t_{PD} is tested with switch S_1 closed and C_L = 50pF.
3. For 3-State outputs; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

VIRGIN STATE

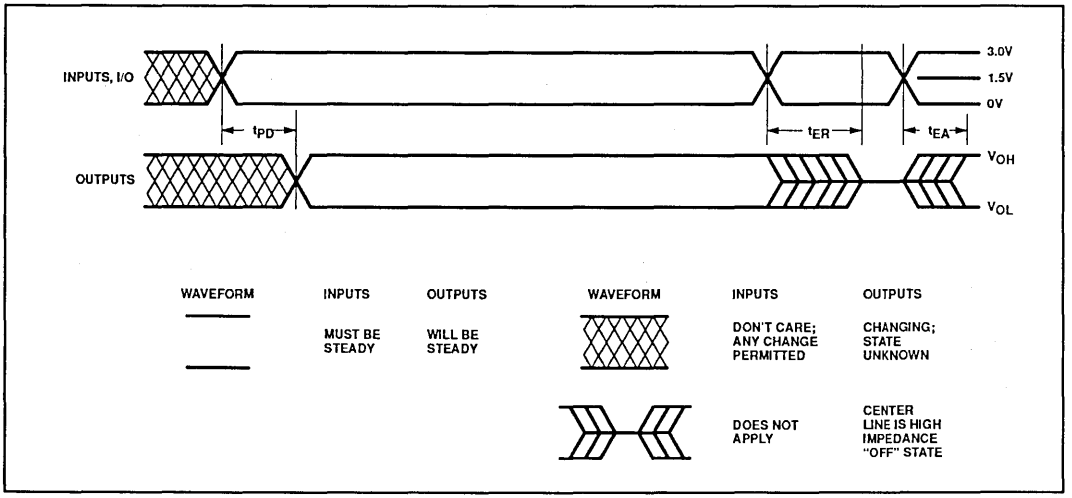
A factory shipped virgin device contains all fusible links open, such that:

1. All outputs are enabled.
2. All p-terms are enabled in the AND array.

TIMING DEFINITIONS

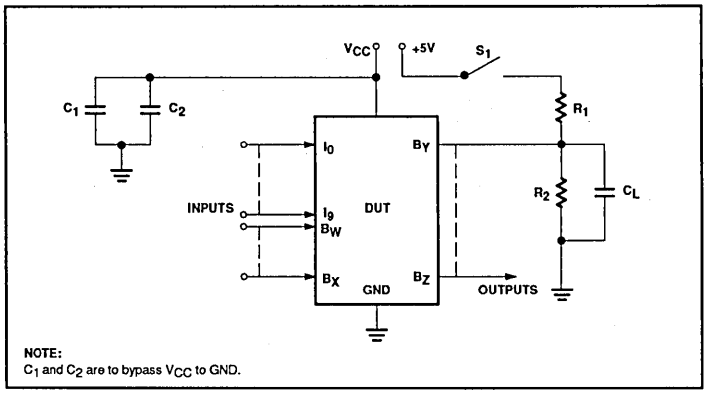
SYMBOL	PARAMETER
t_{PD}	Input to output propagation delay.
t_{EA}	Input to Output Enable delay (Output Enable).
t_{ER}	Input to output disable (3-State) delay (Output Disable).

TIMING DIAGRAM



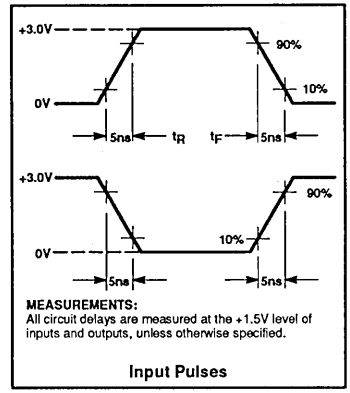
Programmable AND array logic (16 × 64 × 8) PLHS16L8A/B

AC TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

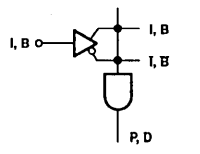
The PLHS16L8A/B is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE design software package. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLHS16L8A/B architecture.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematics capture format.

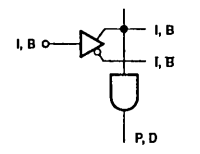
PLHS16L8A/B designs can also be generated using the program table format, detailed on the following page. This program table entry (PTE) format is supported on the Signetics AMAZE PLD design software. AMAZE is available free of charge to qualified users.

To implement the desired logic functions, each logic variable (I, B, P and D) from the logic equations is assigned a symbol. TRUE (High), COMPLEMENT (Low), DON'T CARE and INACTIVE symbols are defined below.

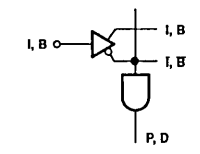
"AND" ARRAY – (I, B)



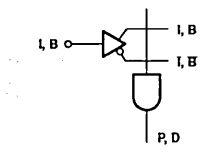
STATE	CODE
INACTIVE[2]	O



STATE	CODE
TRUE	H



STATE	CODE
COMPLEMENT	L



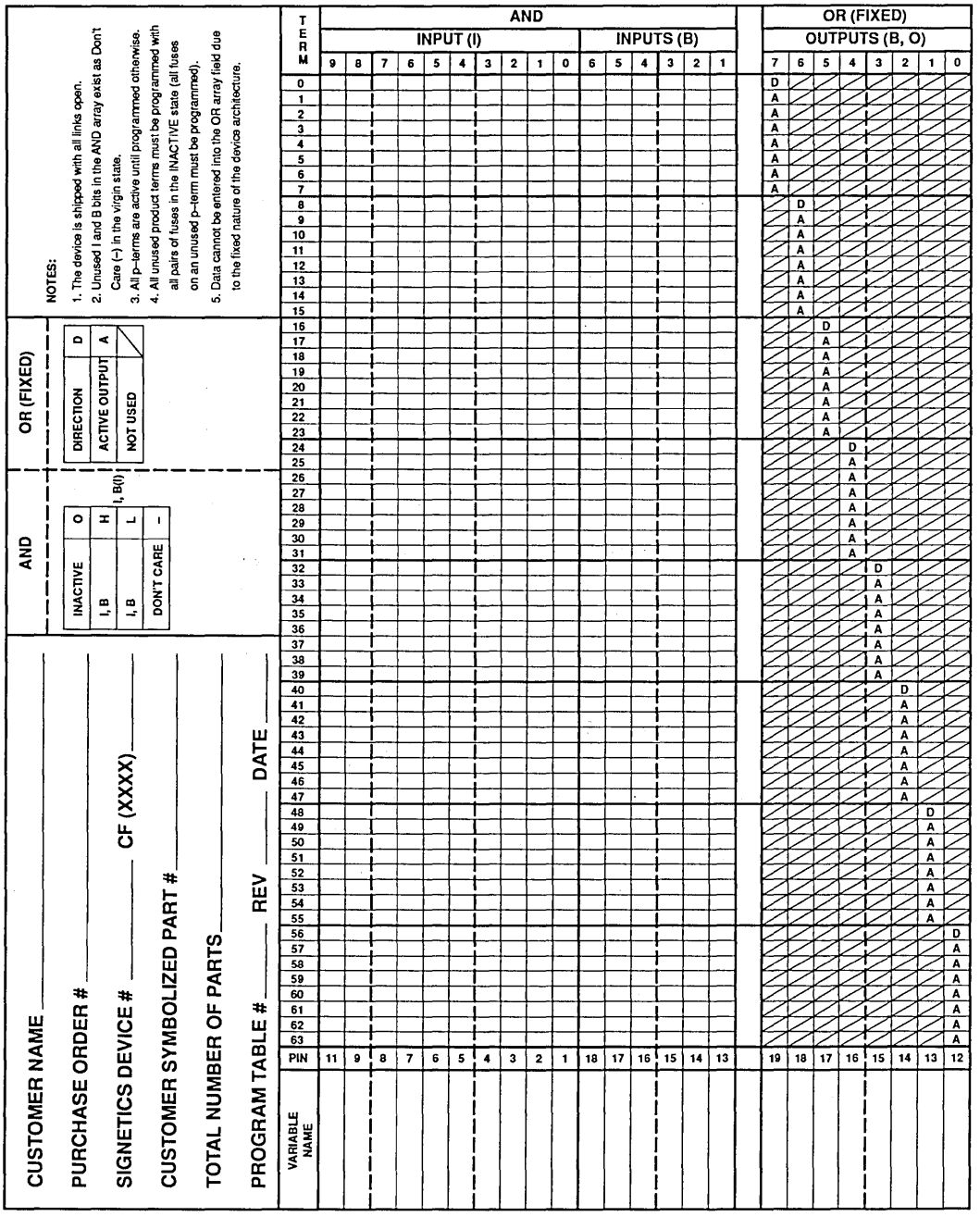
STATE	CODE
DON'T CARE[1]	–

NOTE:
1. This is the initial state of all diodes pairs.
2. All unused product terms must be programmed with all pairs of diodes in the INACTIVE state (all fuses on an unused p-term must be programmed).

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices Inc.
PALASM is a registered trademark of AMD Corp.

Programmable AND array logic (16 × 64 × 8) PLHS16L8A/B

PROGRAM TABLE



Document No.	
ECN No.	
Date of Issue	June 1990
Status	Preliminary Specification
Programmable Logic Devices	

PLQ16R8–5 Series

PAL®-type devices

16L8, 16R8, 16R6, 16R4

FEATURES

- Ultra high-speed
  - t_{PD} = 5ns and f_{MAX} = 118MHz
- 100% functionally and pin-for-pin compatible with industry standard 20-pin PAL ICs
- Power-up reset function to enhance state machine design and testability
- Design support provided via AMAZE and other CAD tools for Series 20 PAL devices
- Field-programmable on industry standard programmers
- Security fuse
- Individual 3-State control of all outputs
- Register Preload for testability
- Power-up 3-State
- 20-Pin DIP and 20-Pin PLCC

DESCRIPTION

The Signetics PLQ16XX family consists of ultra high-speed 5ns versions of Series 20 PAL devices.

The PLQ16XX family is 100% functional and pin-compatible with the 16L8, 16R8, 16R6, and 16R4 Series devices.

The sum of products (AND-OR) architecture is comprised of 64 programmable AND gates and 8 fixed OR gates. Multiple bidirectional pins provide variable input/output pin ratios. Individual 3-State control of all outputs and registers with feedback (R8, R6, R4) is also provided. Proprietary designs can be protected by programming the security fuse.

The PLQ16R8, R6, and R4 have D-type flip-flops which are loaded on the Low-to-High transition of the clock input.

In order to facilitate state machine design and testing, a power-up reset function has been incorporated into these devices to reset all internal registers to active-Low after a specific period of time.

The Signetics State-of-the-Art BiCMOS process, known as QUBiC, has been employed to achieve higher levels of operating performance for the PLQ16XX family of PLDs. The QUBiC transistors have been optimized to provide two-thirds more speed at less than half the power consumed from products using our last generation of bipolar technology. QUBiC reduces on-chip delays and provides high output drive currents while consuming power at very low levels.

The PLQ16XX family of devices are field programmable, enabling the user to quickly generate custom patterns using standard programming equipment. See the programmer chart for qualified programmers.

The AMAZE software package from Signetics supports easy design entry for the PLQ16XX series as well as other PLD devices from Signetics. The PLQ16XX series are also supported by other standard CAD tools for PAL-type devices.

Order codes are listed in the Ordering Information table.

DEVICE NUMBER	DEDICATED INPUTS	COMBINATORIAL OUTPUTS	REGISTERED OUTPUTS
PLQ16L8	10	8 (6 I/O)	0
PLQ16R8	8	0	8
PLQ16R6	8	2 I/O	6
PLQ16R4	8	4 I/O	4

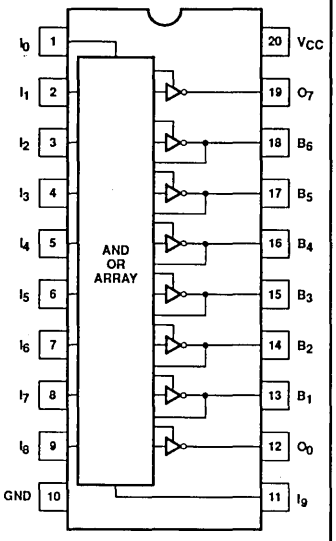
®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

PAL-type devices
16L8, 16R8, 16R6, 16R4

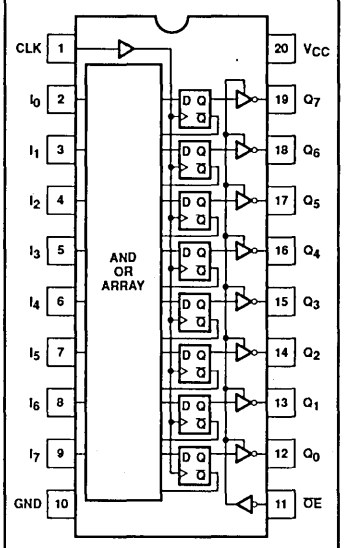
PLQ16R8–5 Series

PIN CONFIGURATIONS

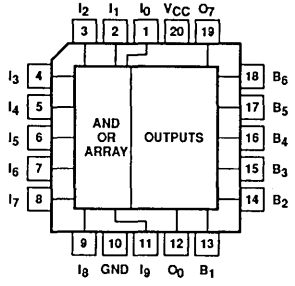
PLQ16L8



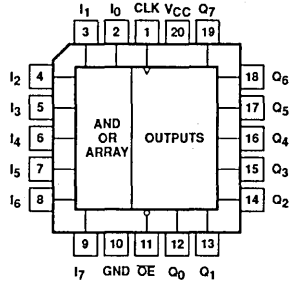
PLQ16R8



PLQ16L8



PLQ16R8



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

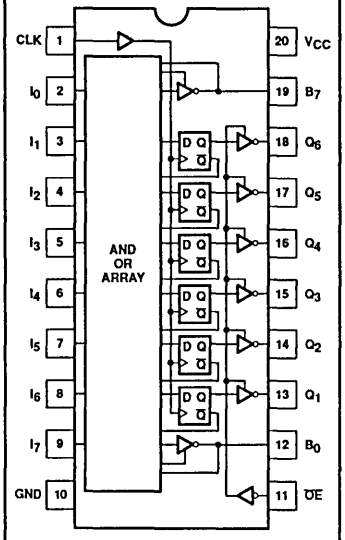
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

PAL-type devices
16L8, 16R8, 16R6, 16R4

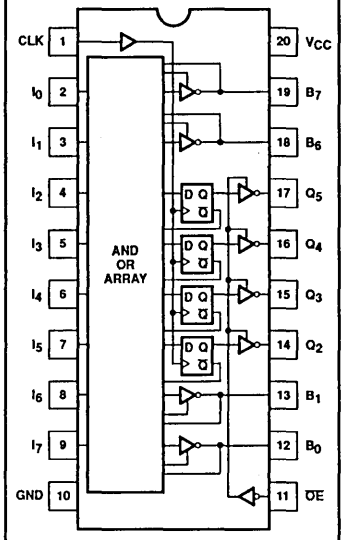
PLQ16R8–5 Series

PIN CONFIGURATIONS

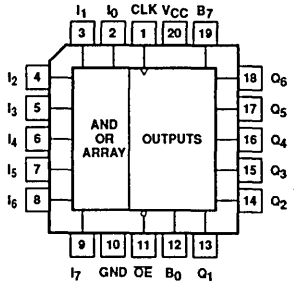
PLQ16R6



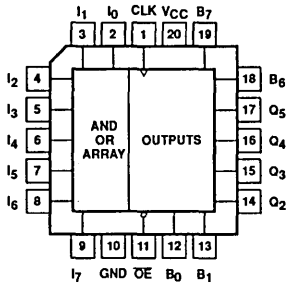
PLQ16R4



PLQ16R6



PLQ16R4



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock Input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

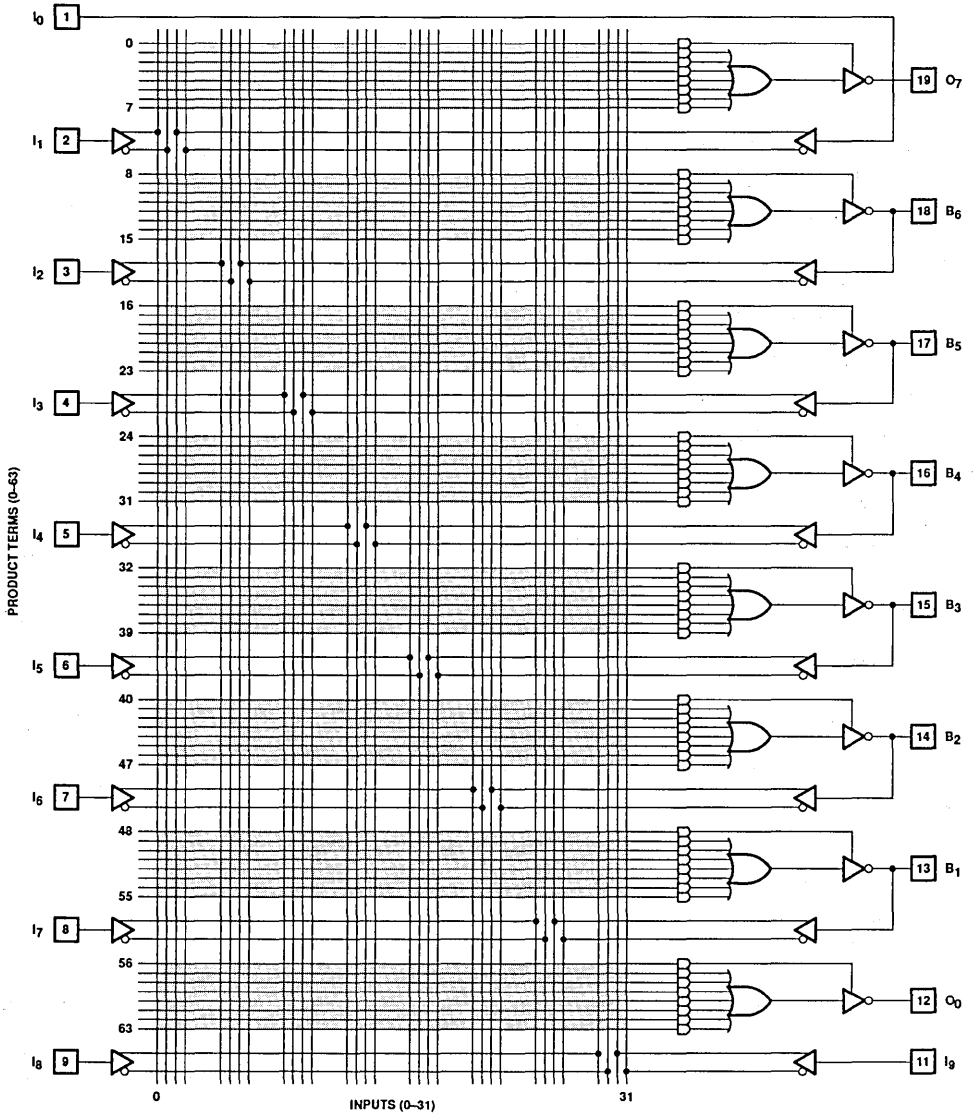
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock Input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLQ16R8–5 Series

LOGIC DIAGRAM

PLQ16L8



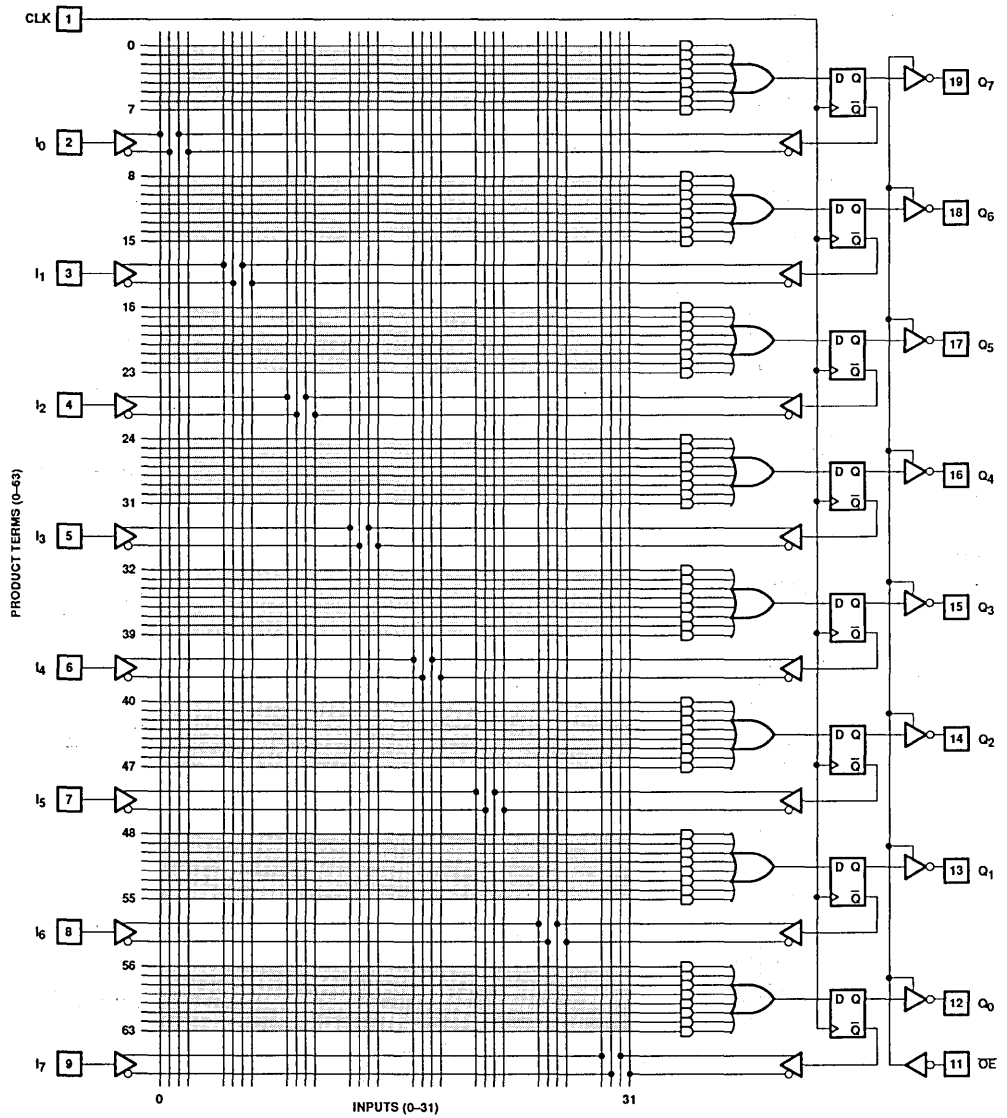
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLQ16R8–5 Series

LOGIC DIAGRAM

PLQ16R8

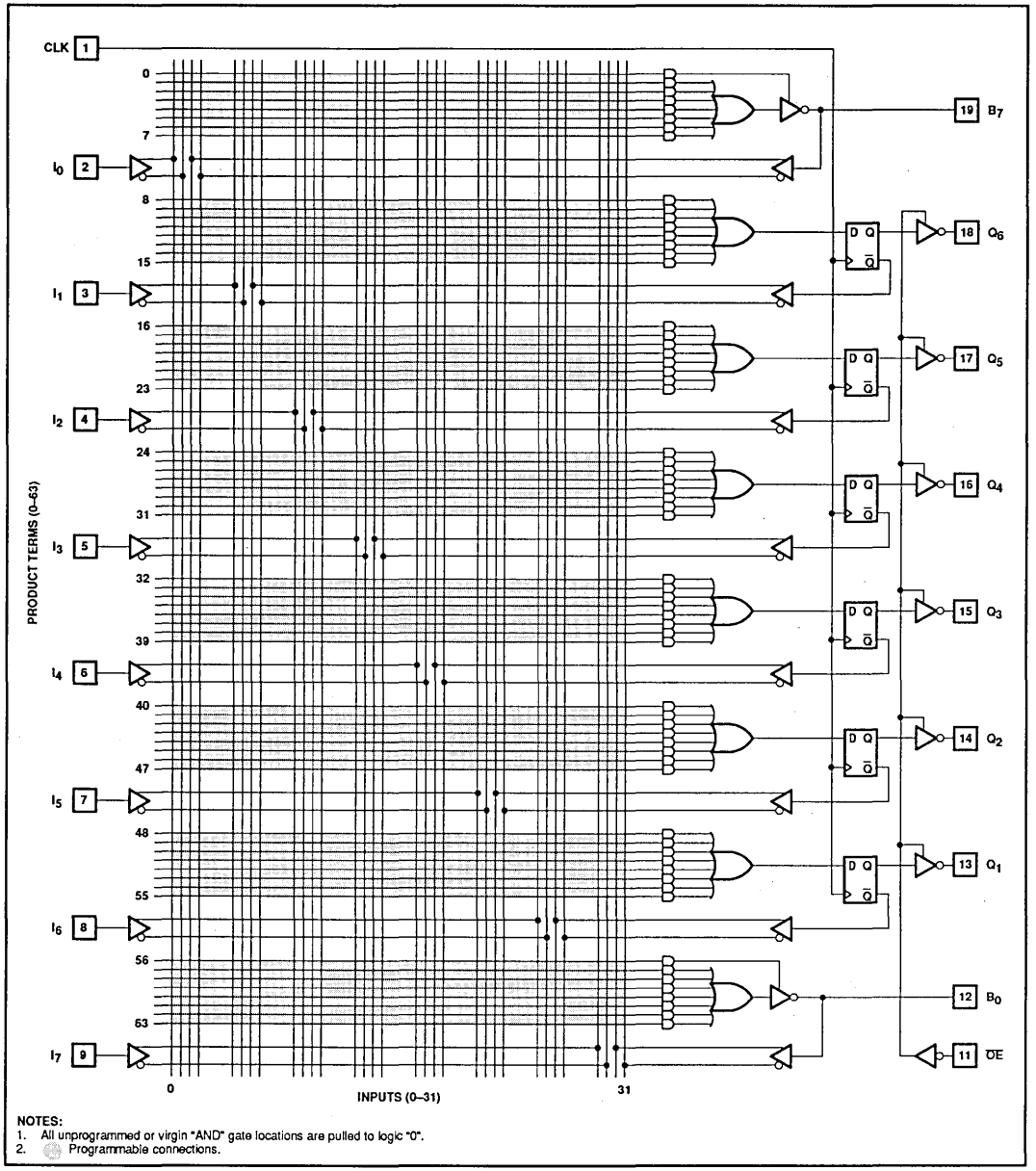


NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ⊙ Programmable connections.

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLQ16R8–5 Series

LOGIC DIAGRAM PLQ16R6



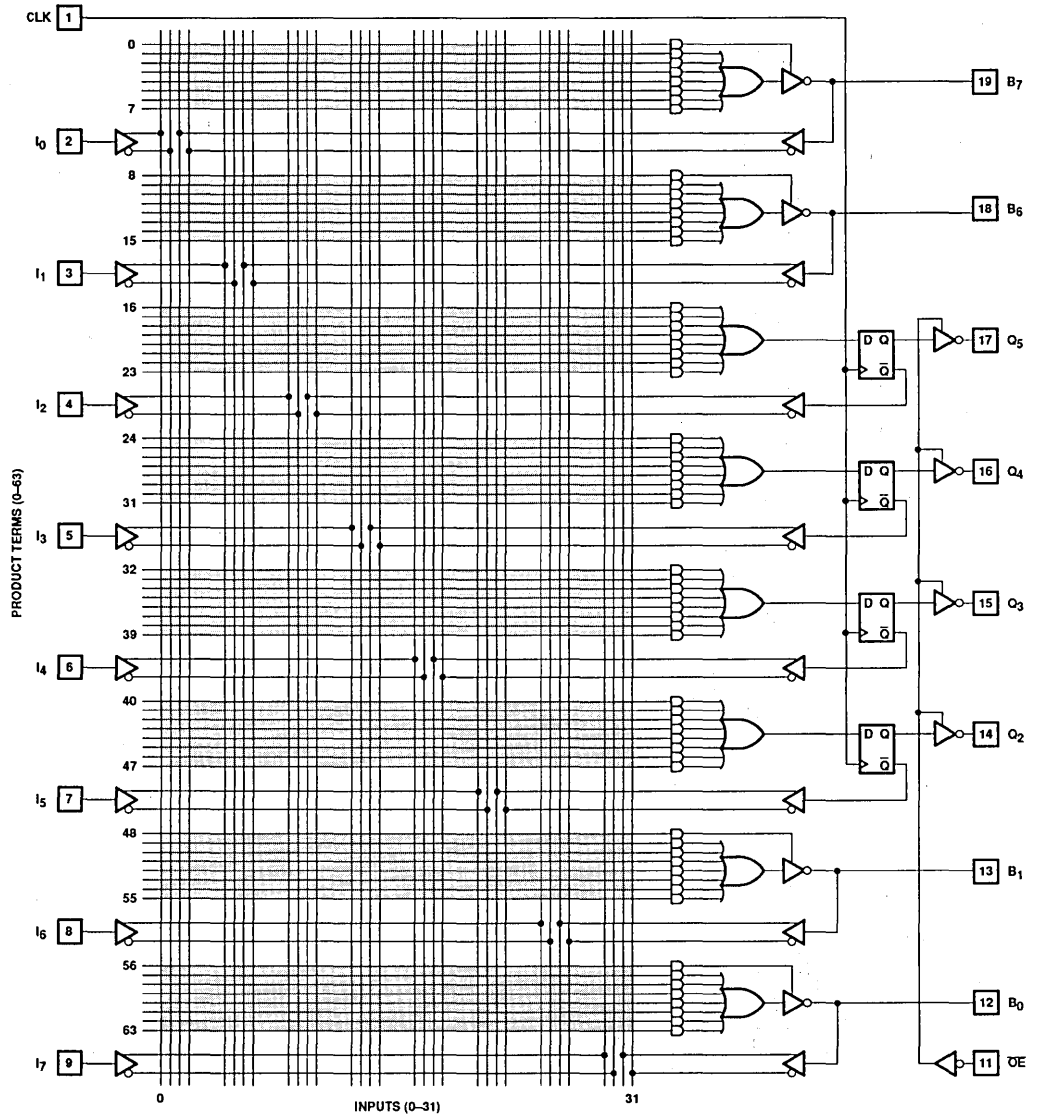
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. Programmable connections.

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLQ16R8–5 Series

LOGIC DIAGRAM

PLQ16R4



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ⊗ Programmable connections.

PAL-type devices 16L8, 16R8, 16R6, 16R4

PLQ16R8–5 Series

FUNCTIONAL DESCRIPTIONS

The PLQ16XX series utilizes the familiar sum-of-products implementation consisting of a programmable AND array and a fixed OR array. These devices are capable of replacing an equivalent of four or more SSI/MSI integrated circuits to reduce package count and board area occupancy, consequently improving reliability and design cycle over Standard Cell or gate array options. By programming the security fuse, proprietary designs can be protected from duplication.

The PLQ16XX series consists of four PAL-type devices. Depending on the particular device type, there are a variable number of combinatorial and registered outputs available to the designer. The PLQ16L8 is a combinatorial part with 8 user configurable outputs (6 bidirectional), while the other three devices, PLQ16R8, PLQ16R6, PLQ16R4, have respectively 8, 6, and 4 output registers.

3-State Outputs

The PLQ16XX series devices also feature 3-State output buffers on each output pin which can be programmed for individual control of all outputs. The registered outputs (Qn) are controlled by an external input (/OE), and the combinatorial outputs (On, Bn) use a product term to control the enable function.

Programmable Bidirectional Pins

The PLQ16XX products feature variable Input/Output ratios. In addition to 8 dedicated inputs, each combinatorial output pin of the registered devices can be individually programmed as an input or output. The PLQ16L8 provides 10 dedicated inputs and 6 Bidirectional I/O lines that can be individually configured as inputs or outputs.

Output Registers

The PLQ16R8 has 8 output registers, the 16R6 has 6, and the 16R4 has 4. Each output register is a D-type flip-flop which is loaded on the Low-to-High transition of the clock input. These output registers are capable of feeding the outputs of the registers back into the array to facilitate design of synchronous state machines.

Power-up Reset

By resetting all flip-flops to a logic Low, as the power is turned on, the PLQ16R8, R6, R4 enhance state machine design and initialization capability.

Register Preload

Preload function allows the register to be loaded from the output pins. This feature allows functional testing of sequential patterns by loading output states.

Power-up 3-State

All outputs will be disabled when V_{CC} is 3.0V ± 20% (25°C). This special feature keeps outputs 3-Stated during power-up. Only when V_{CC} reaches its normal operating range will device function normally.

Software Support

Like other Programmable Logic Devices from Signetics, the PLQ16XX series are supported by AMAZE, the PC-based software development tool from Signetics. The PLQ16XX family of devices are also supported by standard CAD tools for PAL devices, including ABEL and CUPL.

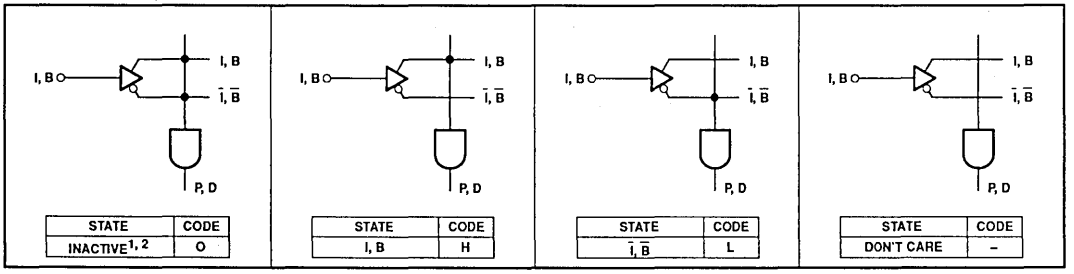
AMAZE is available free of charge to qualified users.

Logic Programming

The PLQ16XX series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLQ16XX architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

AND ARRAY – (I, B)



STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
$\bar{I}, \bar{B}$	L

STATE	CODE
DON'T CARE	–

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All P_n terms are disabled.
2. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic Dual-In-Line 300mil-wide	PLQ16R8–5N PLQ16R6–5N PLQ16R4–5N PLQ16L8–5N
20-Pin Plastic Leaded Chip Carrier (PLCC)	PLQ16R8–5A PLQ16R6–5A PLQ16R4–5A PLQ16L8–5A

NOTE:
The PLQ16XX series of devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Signetics Military Data Handbook.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7.0	V_{DC}
V_{IN}	Input voltage	–1.2	+7.0	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = 24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2 mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low[3]	V_{IN} = 0.40V	–250			µA
I_{IH}	High[3]	V_{IN} = 2.7V			25	µA
I_I	Maximum input current	V_{IN} = 5.5V, V_{CC} = MAX			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage	V_{OUT} = 2.7V			100	µA
I_{OZL}	Output leakage	V_{OUT} = 0.4V	–100			µA
I_{OS}	Short circuit[4, 5]	V_{OUT} = 0.5V	–30		–130	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX			180	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5V				
		V_{OUT} = 2.0V		8		pF
C_B	I/O (B)	V_{OUT} = 2V, f = 1MHz		8		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Test one at a time.
5. Duration of short circuit should not exceed 1 second.
6. These parameters are not 100% tested but periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 200\Omega$, $R_2 = 390\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

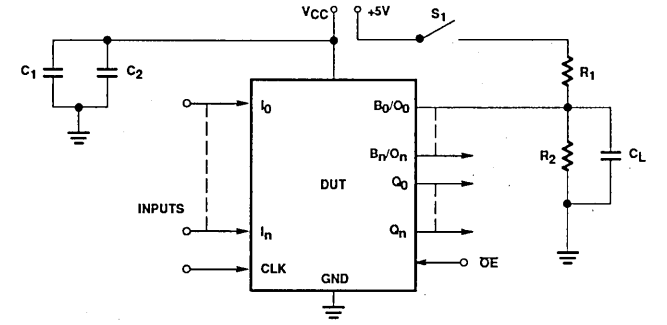
SYMBOL	PARAMETER	FROM	TO	LIMITS MIN[1]	LIMITS MAX	UNIT
Pulse Width						
t_{CKH}	Clock High	CLK+	CLK–	3.0		ns
t_{CKL}	Clock Low	CLK–	CLK+	3.0		ns
t_{CKP}	Period	CLK+	CLK+	6.0		ns
Setup & Hold time						
t_{IS}	Input	Input or feedback	CLK+	4.0		ns
t_{IH}	Input	CLK+	Input or feedback	0		ns
Propagation delay						
t_{CKO}	Clock	CLK±	Q±		4.5	ns
t_{CKF}	Clock[3]	CLK±	$\overline{Q}$		2.5	ns
t_{PD}	Output (16L8, R6, R4)[2]	I, B	Output		5.0	ns
t_{OE1}	Output enable[4]	$\overline{OE}$	Output enable		6.0	ns
t_{OE2}	Output enable[4,5]	I	Output enable		8.0	ns
t_{OD1}	Output disable[4]	$\overline{OE}$	Output disable		6.0	ns
t_{OD2}	Output disable[4,5]	I	Output disable		8.0	ns
t_{SKW}	Output	Q	Q		1.0	ns
t_{PPR}	Power-Up Reset	V_{CC}+	Q+		8.0	ns
Frequency (16R8, R6, R4)						
f_{MAX}	No feedback $1/(t_{CKL} + t_{CKH})$[6]				167	MHz
	Internal feedback $1/(t_{IS} + t_{CKF})$[6]				154	MHz
	External feedback $1/(t_{IS} + t_{CKO})$[6]				118	MHz

* For definitions of the terms, please refer to the Timing/Frequency Definitions tables.

NOTES:

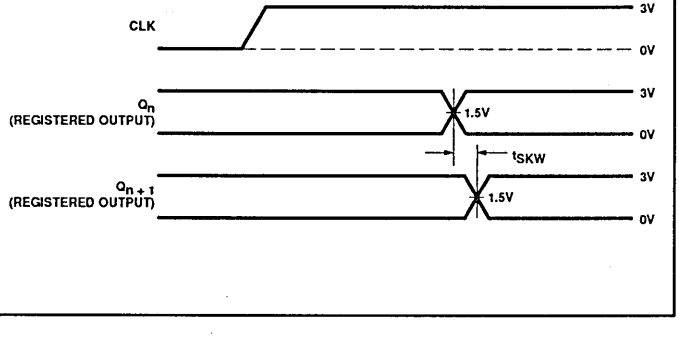
1. CL = 0pF while measuring minimum output delays.
2. t_{PD} test conditions: C_L = 50pF (with jig and scope capacitance), V_{IH} = 3V, V_{IL} = 0V, $V_{OH} = V_{OL}$ = 1.5V.
3. t_{CKF} was calculated from measured internal f_{MAX}.
4. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
5. Same function as t_{OE1} and t_{OD1}, with the difference of using product term control.
6. Not 100% tested, but calculated at initial characterization and at any time a modification in design takes place which may affect the frequency.

TEST LOAD CIRCUIT

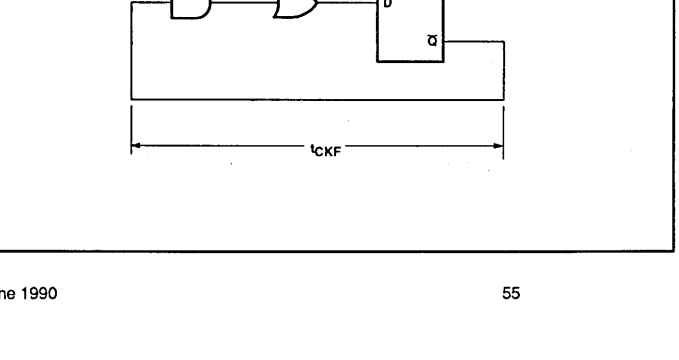


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

OUTPUT REGISTER SKEW



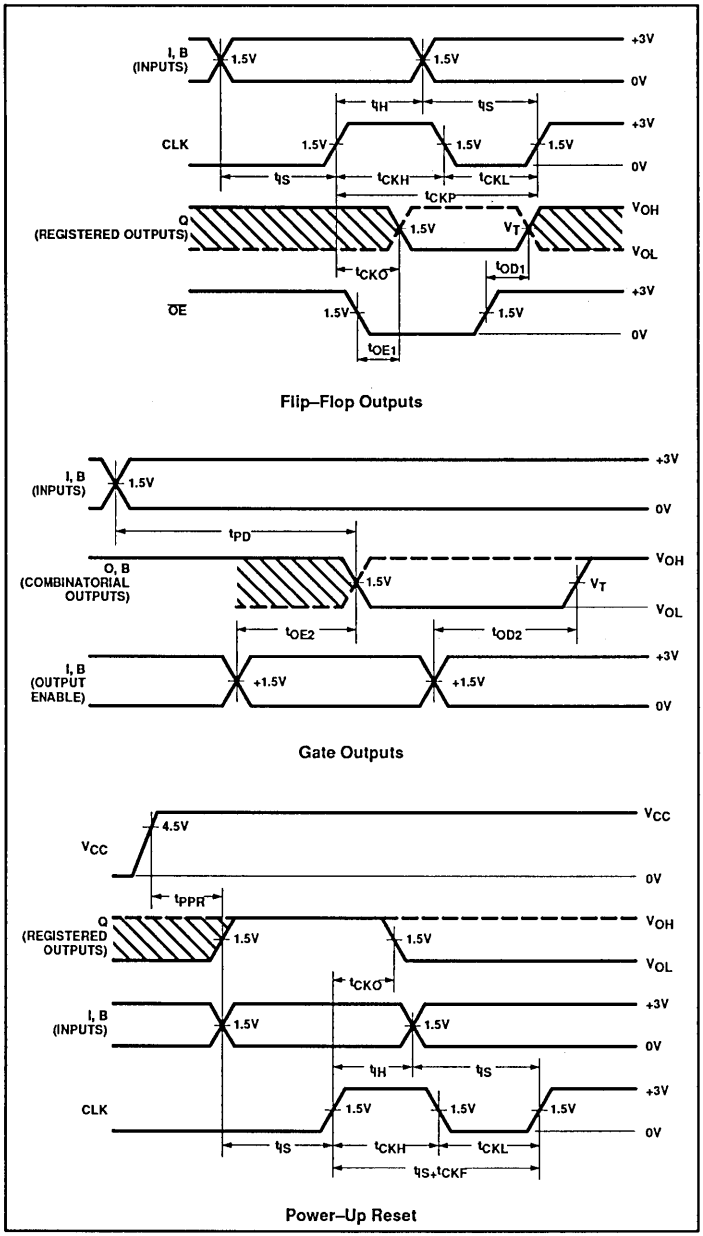
CLOCK TO FEEDBACK PATH



PAL-type devices
16L8, 16R8, 16R6, 16R4

PLQ16R8–5 Series

TIMING DIAGRAMS[1, 2]



TIMING DEFINITIONS

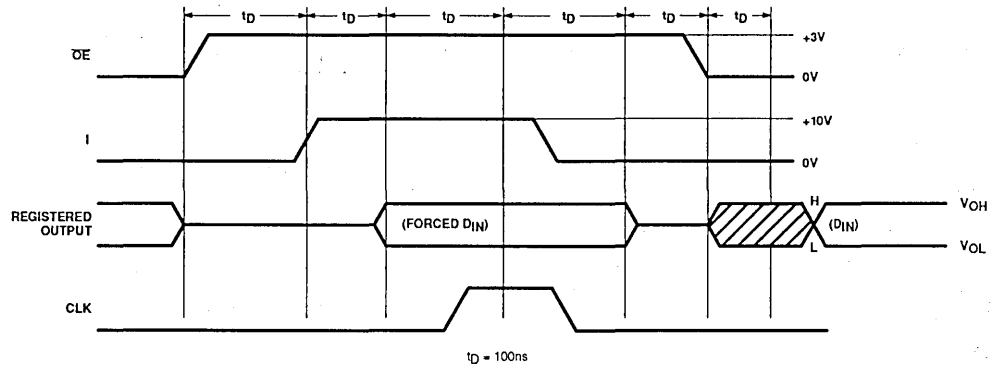
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{IS}	Required delay between beginning of valid input and positive transition of clock.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKF}	Delay between positive transition of clock and when internal $\overline{Q}$ output of flip-flop becomes valid.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the Off-State.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the Off-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.
t_D	Delay between each input change.

FREQUENCY DEFINITIONS

f_{MAX}	**No feedback:** Determined by the minimum clock period, $1/(t_{CKL} + t_{CKH})$. **Internal feedback:** Determined by the internal delay from flip-flop outputs through the internal feedback and array to the flip-flop inputs, $1/(t_{IS} + t_{CKF})$. **External feedback:** Determined by clock-to-output delay and input setup time, $1/(t_{IS} + t_{CKO})$.

NOTES:
1. Input pulse amplitude is 0V to 3V.
2. Input rise and fall times are 2.0ns typical.

TIMING DIAGRAMS (Continued)



Output Register Preload

PROGRAMMING

The PLQ16XX Series are programmable on conventional programmers for 20-pin PAL® devices. Refer to the following charts for qualified manufacturers of programmers and software tools:

PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	SYSTEM 29B, LogicPak™ 303A–V04 ADAPTER 303A–011A–V08 303A–011B–V04 UNISITE 40/48, V2.3 (DIP) V2.5 (PLCC) MODEL 60, 60A/H, V.13	16L8–7/16L8D : 1B/17 16R8–7/16R8D : 1B/24 16R6–7/16R6D : 1B/24 16R4–7/16R4D : 1B/24
STAG MICROSYSTEMS, INC. 1600 WYATT DRIVE SUITE 3 SANTA CLARA, CALIFORNIA 95054 (408)988–1118	ZL30/30A PROGRAMMER REV. 30A31 PPZ PROGRAMMER TBA	16L8–7/16L8D : 11/29 16R8–7/16R8D : 11/30 16R6–7/16R6D : 11/30 16R4–7/16R4D : 11/30

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
SIGNETICS COMPANY 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	AMAZE SOFTWARE REV. 1.7 AND LATER
DATA I/O 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	ABEL™ SOFTWARE REV. 1.0 AND LATER
LOGICAL DEVICES, INC. 1201 NORTHWEST 65TH PLACE FORT LAUDERDALE, FLORIDA 33309 (800)331–7766	CUPL™ SOFTWARE REV. 1.01 AND LATER

Document No.	853–1358
ECN No.	99790
Date of Issue	June 14, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS16R8D/–7 Series

PAL®-type devices

16L8, 16R8, 16R6, 16R4

FEATURES

- Ultra high-speed
  - t_{PD} = 7.5ns and f_{MAX} = 74MHz for the PLUS16R8-7 Series
  - t_{PD} = 10ns and f_{MAX} = 60 MHz for the PLUS16R8D Series
- 100% functionally and pin-for-pin compatible with industry standard 20-pin PAL ICs
- Power-up reset function to enhance state machine design and testability
- Design support provided via AMAZE and other CAD tools for Series 20 PAL devices
- Field-programmable on industry standard programmers
- Security fuse
- Individual 3-State control of all outputs

DESCRIPTION

The Signetics PLUS16XX family consists of ultra high-speed 7.5ns and 10ns versions of Series 20 PAL devices.

The PLUS16XX family is 100% functional and pin-compatible with the 16L8, 16R8, 16R6, and 16R4 Series devices.

The sum of products (AND-OR) architecture is comprised of 64 programmable AND gates and 8 fixed OR gates. Multiple bidirectional pins provide variable input/output pin ratios. Individual 3-State control of all outputs and registers with feedback (R8, R6, R4) is also provided. Proprietary designs can be protected by programming the security fuse.

The PLUS16R8, R6, and R4 have D-type flip-flops which are loaded on the Low-to-High transition of the clock input.

In order to facilitate state machine design and testing, a power-up reset function has been incorporated into these devices to reset all internal registers to Active-Low after a specific period of time.

The Signetics State-of-the-Art oxide isolation Bipolar fabrication process is employed to achieve high-performance operation.

The PLUS16XX family of devices are field programmable, enabling the user to quickly generate custom patterns using standard programming equipment. See the programmer chart for qualified programmers.

The AMAZE software package from Signetics supports easy design entry for the PLUS16XX series as well as other PLD devices from Signetics. The PLUS16XX series are also supported by other standard CAD tools for PAL-type devices.

Order codes are listed in the Ordering Information table.

DEVICE NUMBER	DEDICATED INPUTS	COMBINATORIAL OUTPUTS	REGISTERED OUTPUTS
PLQ16L8	10	8 (6 I/O)	0
PLQ16R8	8	0	8
PLQ16R6	8	2 I/O	6
PLQ16R4	8	4 I/O	4

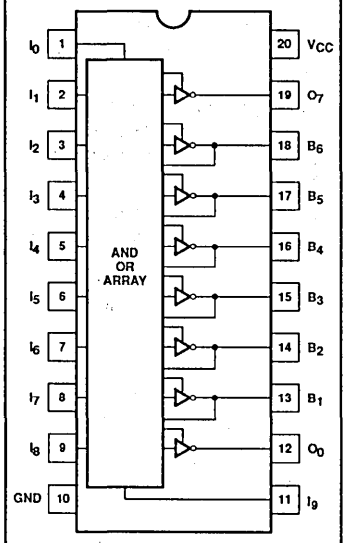
®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

PAL-type devices
16L8, 16R8, 16R6, 16R4

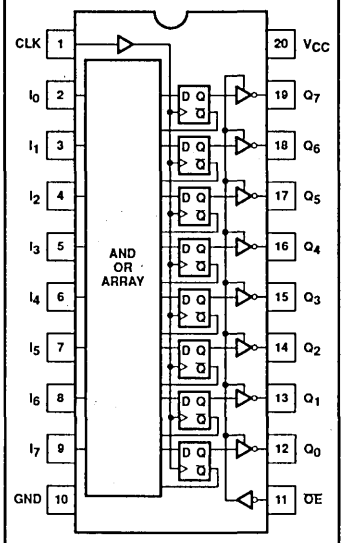
PLUS16R8D/–7 Series

PIN CONFIGURATIONS

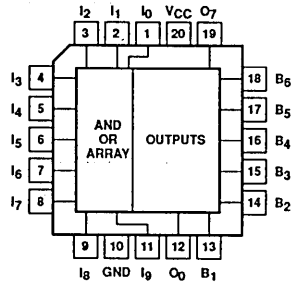
PLUS16L8



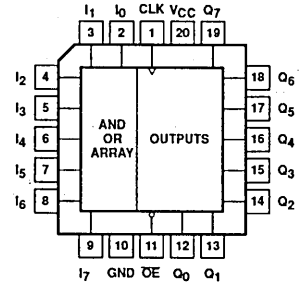
PLUS16R8



PLUS16L8



PLUS16R8



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

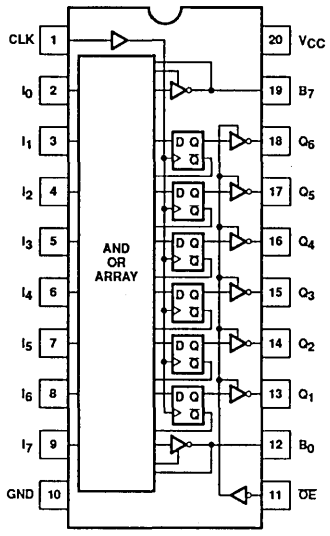
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

PAL-type devices
16L8, 16R8, 16R6, 16R4

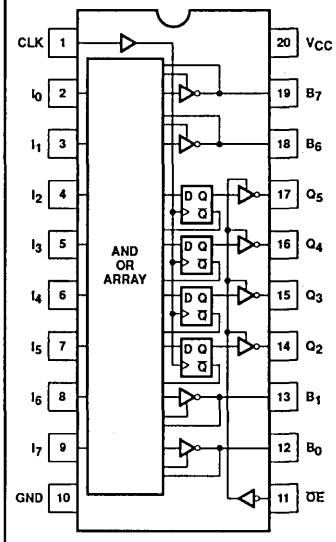
PLUS16R8D/–7 Series

PIN CONFIGURATIONS

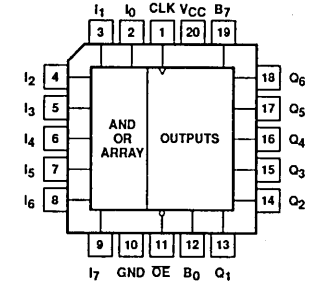
PLUS16R6



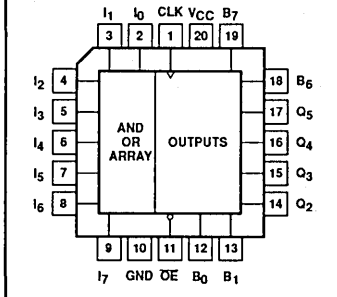
PLUS16R4



PLUS16R6



PLUS16R4



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

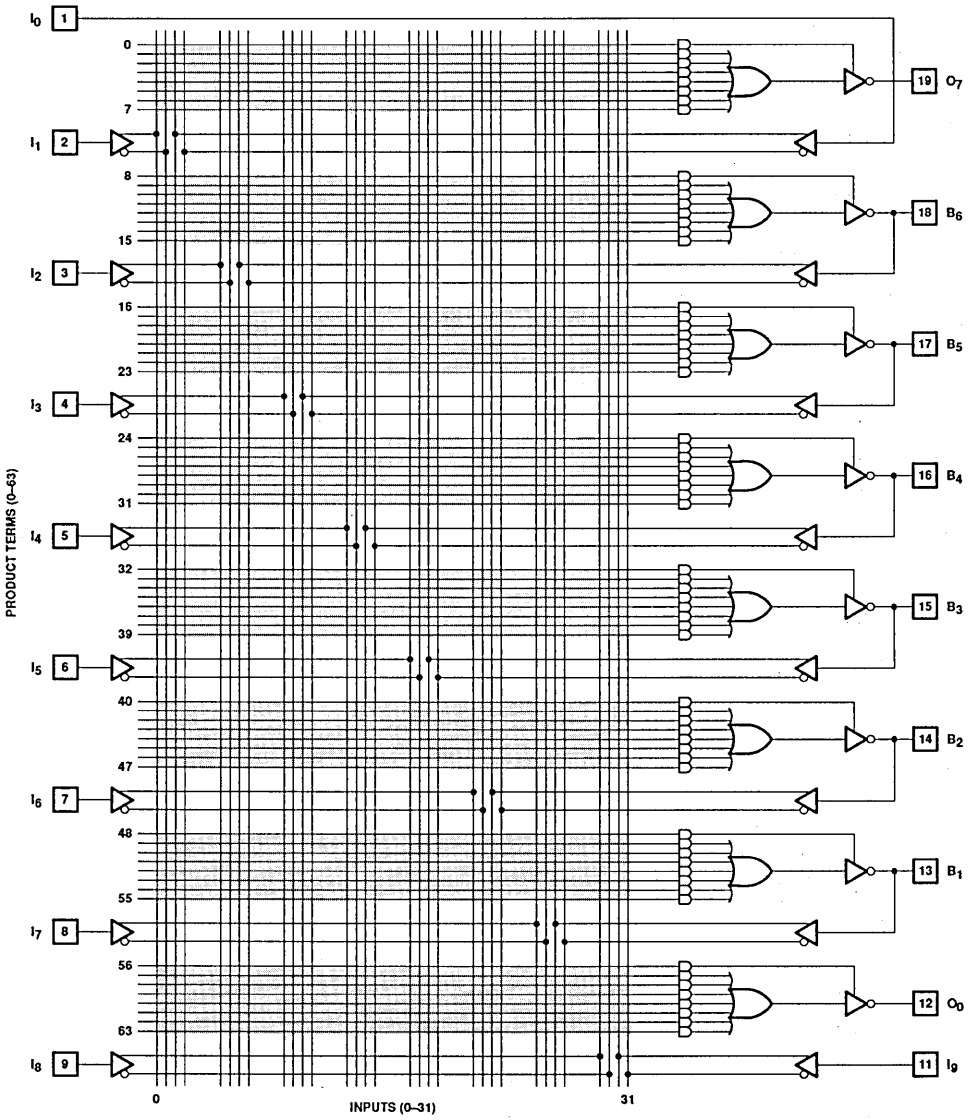
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLUS16R8D/–7 Series

LOGIC DIAGRAM

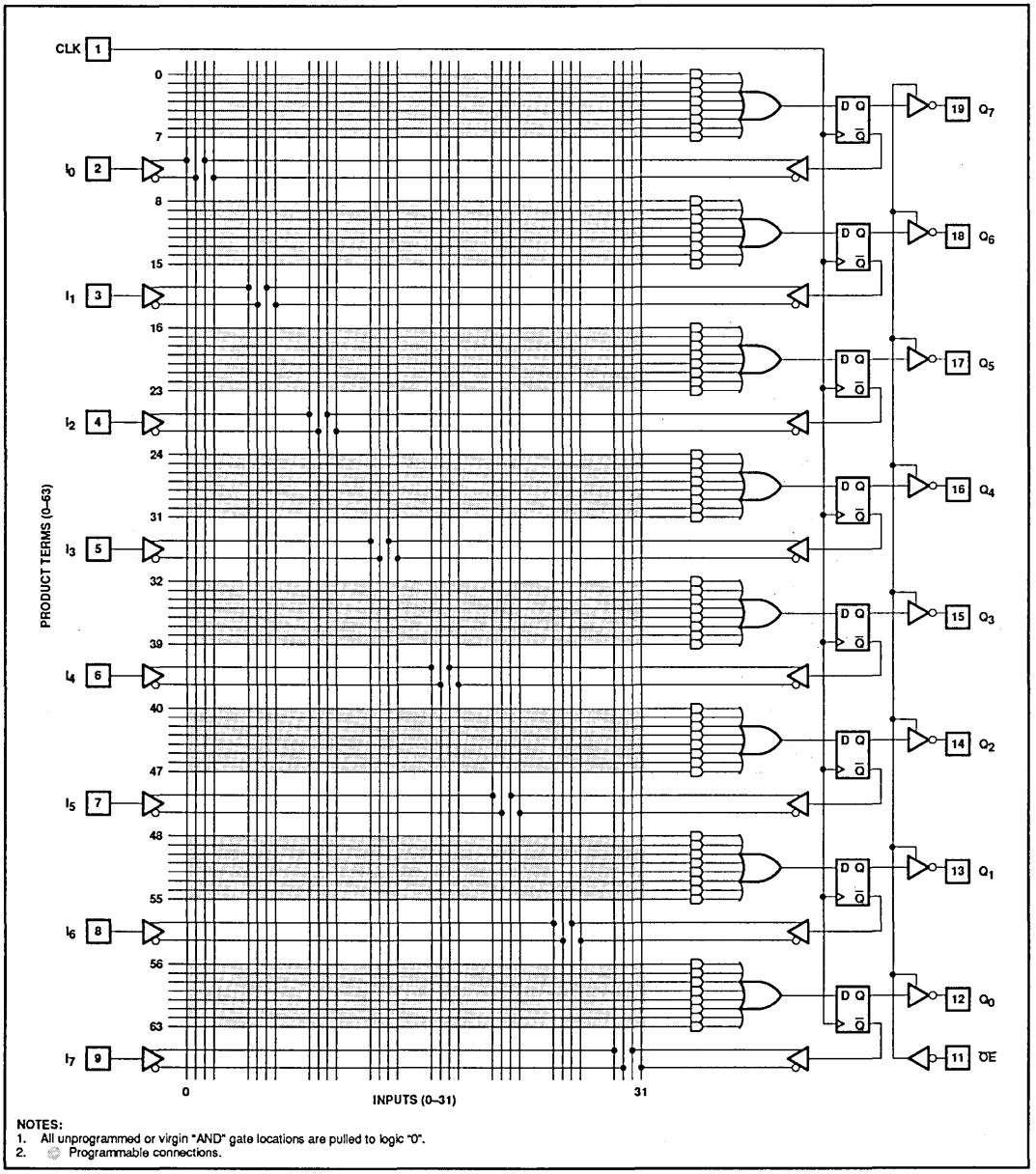
PLUS16L8



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ◌ Programmable connections.

LOGIC DIAGRAM

PLUS16R8



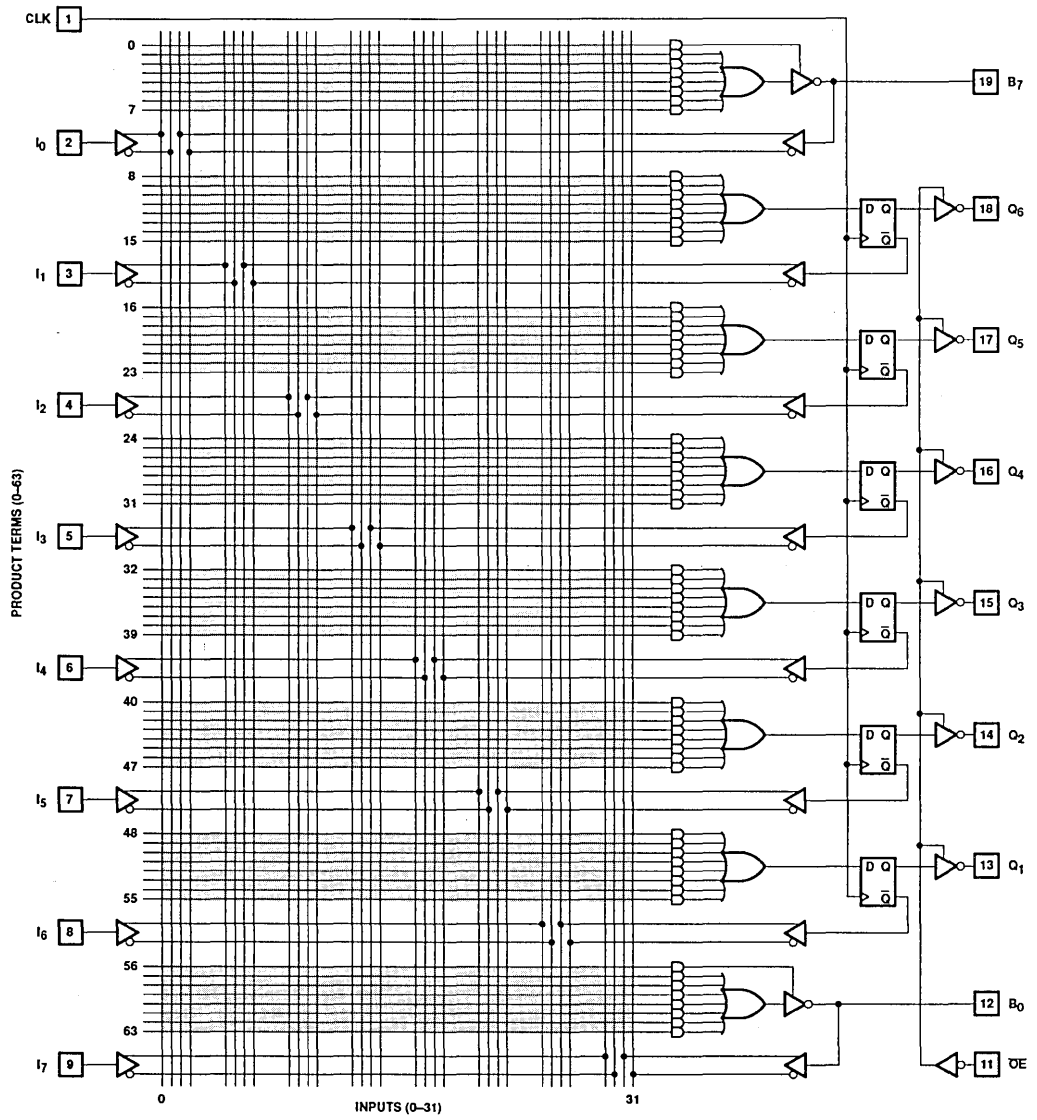
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ⊗ Programmable connections.

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLUS16R8D/–7 Series

LOGIC DIAGRAM

PLUS16R6



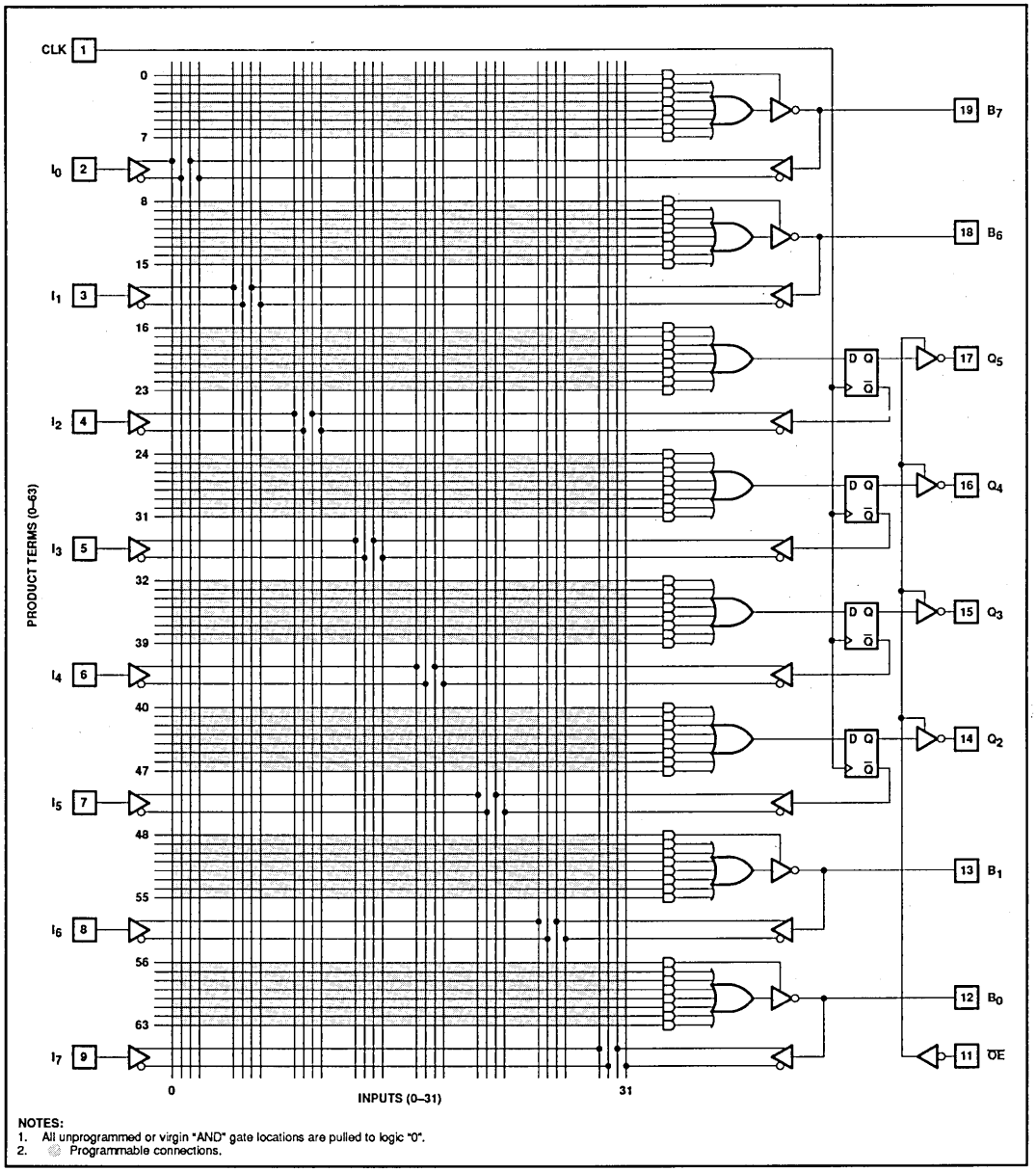
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ◎ Programmable connections.

PAL-type devices
16L8, 16R8, 16R6, 16R4

PLUS16R8D/–7 Series

LOGIC DIAGRAM

PLUS16R4



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

PAL-type devices 16L8, 16R8, 16R6, 16R4

PLUS16R8D/–7 Series

FUNCTIONAL DESCRIPTIONS

The PLUS16XX series utilizes the familiar sum-of-products implementation consisting of a programmable AND array and a fixed OR array. These devices are capable of replacing an equivalent of four or more SSI/MSI integrated circuits to reduce package count and board area occupancy, consequently improving reliability and design cycle over Standard Cell or gate array options. By programming the security fuse, proprietary designs can be protected from duplication.

The PLUS16XX series consists of four PAL-type devices. Depending on the particular device type, there are a variable number of combinatorial and registered outputs available to the designer. The PLUS16L8 is a combinatorial part with 8 user configurable outputs (6 bidirectional), while the other three devices, PLUS16R8, PLUS16R6, PLUS16R4, have respectively 8, 6, and 4 output registers.

3-State Outputs

The PLUS16XX series devices also feature 3-State output buffers on each output pin which can be programmed for individual control of all outputs. The registered outputs (Qn) are controlled by an external input (/OE), and the combinatorial outputs (On, Bn) use a product term to control the enable function.

Programmable Bidirectional Pins

The PLUS16XX products feature variable Input/Output ratios. In addition to 8 dedicated inputs, each combinatorial output pin of the registered devices can be individually programmed as an input or output. The PLUS16L8 provides 10 dedicated inputs and 6 Bidirectional I/O lines that can be individually configured as inputs or outputs.

Output Registers

The PLUS16R8 has 8 output registers, the 16R6 has 6, and the 16R4 has 4. Each output register is a D-type flip-flop which is loaded on the Low-to-High transition of the clock input. These output registers are capable of feeding the outputs of the registers back into the array to facilitate design of synchronous state machines.

Power-up Reset

By resetting all flip-flops to a logic Low, as the power is turned on, the PLUS16R8, R6, R4 enhance state machine design and initialization capability.

Software Support

Like other Programmable Logic Devices from Signetics, the PLUS16XX series are supported by AMAZE, the PC-based software development tool from Signetics. The PLUS16XX family of devices are also supported by standard CAD tools for PAL devices, including ABEL and CUPL.

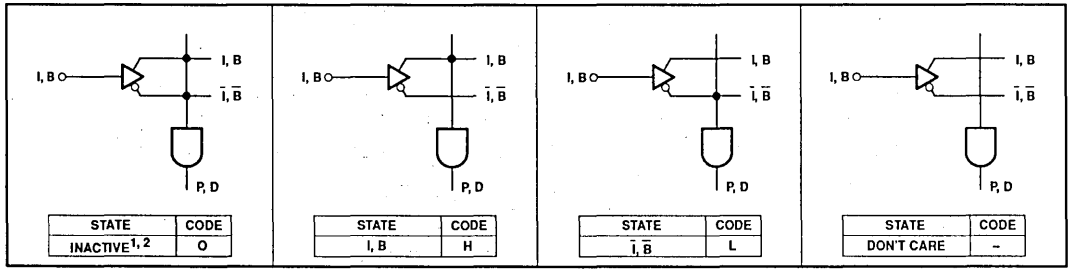
AMAZE is available free of charge to qualified users.

Logic Programming

The PLUS16XX series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLUS16XX architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

AND ARRAY – (I, B)



STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
$\overline{I}$, $\overline{B}$	L

STATE	CODE
DON'T CARE	–

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic Dual-In-Line 300mil-wide	PLUS16R8DN PLUS16R6DN PLUS16R4DN PLUS16L8DN PLUS16R8–7N PLUS16R6–7N PLUS16R4–7N PLUS16L8–7N
20-Pin Plastic Leaded Chip Carrier (PLCC)	PLUS16R8DA PLUS16R6DA PLUS16R4DA PLUS16L8DA PLUS16R8–7A PLUS16R6–7A PLUS16R4–7A PLUS16L8–7A

NOTE:
The PLUS16XX series of devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Signetics Military Data Book.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–1.2	+8.0	V_{DC}
V_{OUT}	Output voltage	–0.5	V_{CC} + 0.5V	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free–air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = 24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2 mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low[3]	V_{IN} = 0.40V	–250			µA
I_{IH}	High[3]	V_{IN} = 2.7V			25	µA
I_I	Maximum input current	V_{IN} = V_{CC} = V_{CCMAX}			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage	V_{OUT} = 2.7V			100	µA
I_{OZL}	Output leakage	V_{OUT} = 0.4V	–100			µA
I_{OS}	Short circuit [4, 5]	V_{OUT} = 0V	–30		–90	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX			180	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5V				
		V_{OUT} = 2.0V		8		pF
C_B	I/O (B)	V_{OUT} = 2V, f = 1MHz		8		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Test one at a time.
5. Duration of short circuit should not exceed 1 second.
6. These parameters are not 100% tested but periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 200\Omega$, $R_2 = 390\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

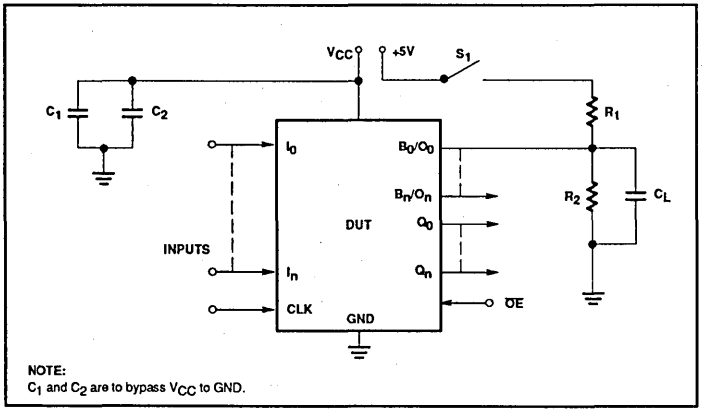
SYMBOL	PARAMETER	FROM	TO	LIMITS					UNIT
				–7			D		
				MIN[1]	TYP	MAX	MIN[1]	MAX	
Pulse Width									
t_{CKH}	Clock High	CK+	CK–	5			7		ns
t_{CKL}	Clock Low	CK–	CK+	5			7		ns
t_{CKP}	Period	CK+	CK+	10			14		ns
Setup & Hold time									
t_{IS}	Input	Input or feedback	CK+	7			9		ns
t_{IH}	Input	CK+	Input or feedback	0			0		ns
Propagation delay									
t_{CKO}	Clock	CK±	Q±	3		6.5	3	7.5	ns
t_{CKF}	Clock[3]	CK±	$\overline{Q}$			3		6.5	ns
t_{PD}	Output (16L8, R6, R4)[2]	I, B	Output	3		7.5	3	10	ns
t_{OE1}	Output enable[4]	$\overline{OE}$	Output enable	3		8	3	10	ns
t_{OE2}	Output enable[4,5]	I	Output enable	3		10	3	10	ns
t_{OD1}	Output disable[4]	$\overline{OE}$	Output disable	3		8	3	10	ns
t_{OD2}	Output disable[4,5]	I	Output disable	3		10	3	10	ns
t_{SKW}	Output	Q	Q			1		1	ns
t_{PPR}	Power–Up Reset	V_{CC}+	Q+			10		10	ns
Frequency (16R8, R6, R4)									
f_{MAX}	No feedback 1/ $(t_{CKL} + t_{CKH})$[6]				100		71.4		MHz
	Internal feedback 1/ $(t_{IS} + t_{CKF})$[6]				90		64.5		MHz
	External feedback 1/ $(t_{IS} + t_{CKO})$[6]				74		60.6		MHz

* For definitions of the terms, please refer to the Timing/Frequency Definitions tables.

NOTES:

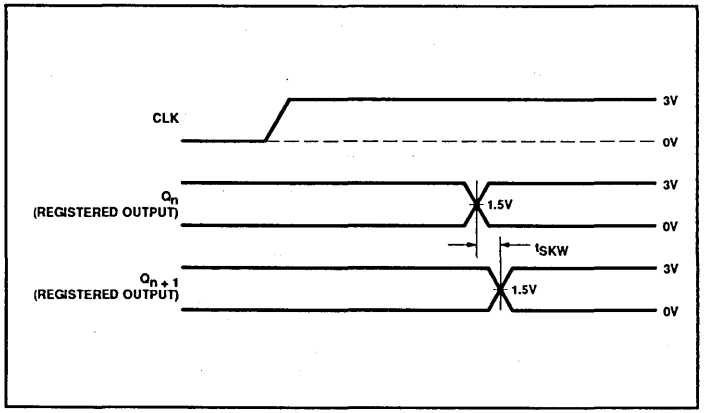
1. CL = 0pF while measuring minimum output delays.
2. t_{PD} test conditions: CL = 50pF (with jig and scope capacitance), $V_{IH} = 3V$, $V_{IL} = 0V$, $V_{OH} = V_{OL} = 1.5V$.
3. t_{CKF} was calculated from measured Internal f_{MAX}.
4. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
5. Same function as t_{OE1} and t_{OD1}, with the difference of using product term control.
6. Not 100% tested, but calculated at initial characterization and at any time a modification in design takes place which may affect the frequency.

TEST LOAD CIRCUIT

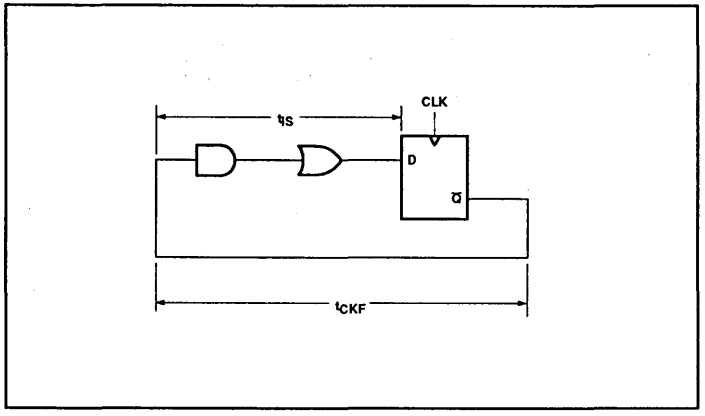


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

OUTPUT REGISTER SKEW



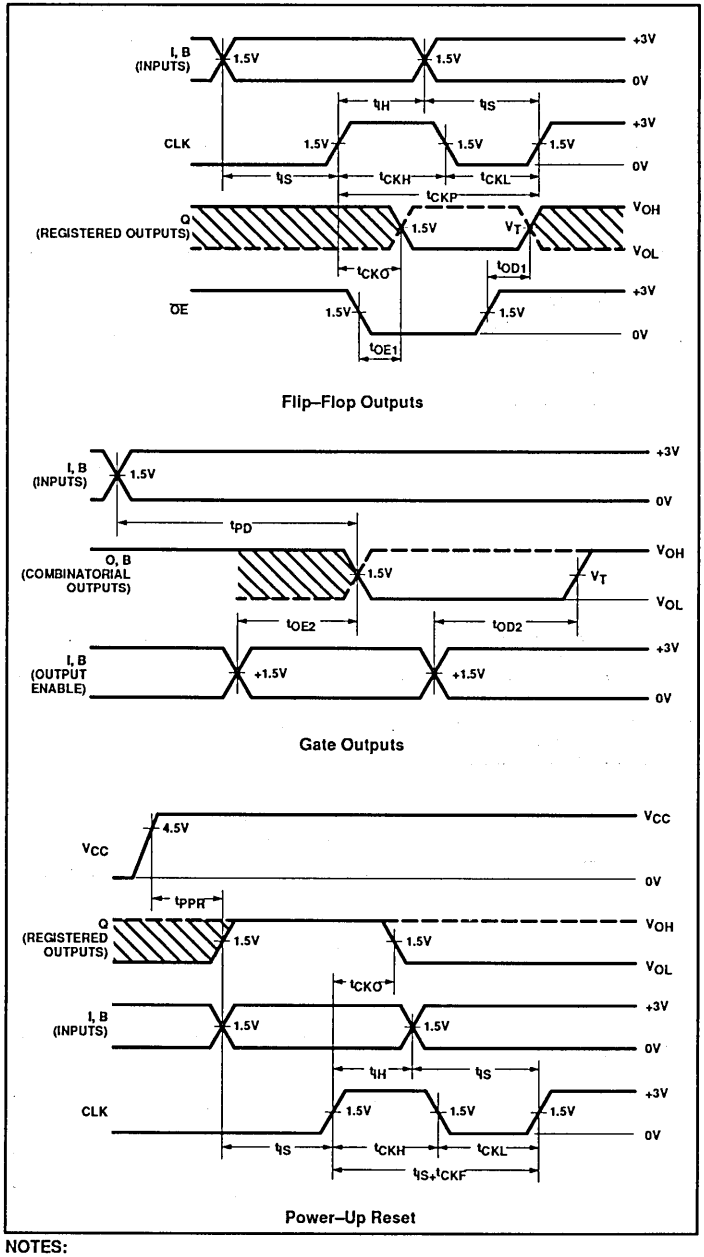
CLOCK TO FEEDBACK PATH



PAL-type devices 16L8, 16R8, 16R6, 16R4

PLUS16R8D/–7 Series

TIMING DIAGRAMS[1, 2]



TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{IS}	Required delay between beginning of valid input and positive transition of clock.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKF}	Delay between positive transition of clock and when internal $\overline{Q}$ output of flip-flop becomes valid.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the Off-State.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the Off-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.

FREQUENCY DEFINITIONS

f_{MAX}	**No feedback:** Determined by the minimum clock period, $1/(t_{CKL} + t_{CKH})$. **Internal feedback:** Determined by the internal delay from flip-flop outputs through the internal feedback and array to the flip-flop inputs, $1/(t_{IS} + t_{CKF})$. **External feedback:** Determined by clock-to-output delay and input setup time, $1/(t_{IS} + t_{CKO})$.

NOTES:
1. Input pulse amplitude is 0V to 3V.
2. Input rise and fall times are 2.5ns.

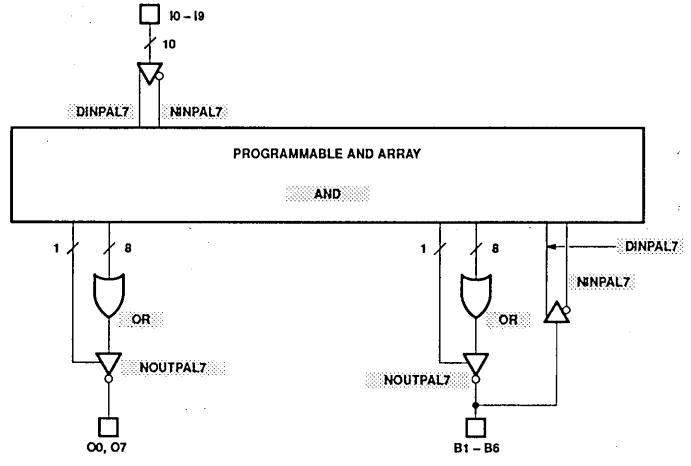
PROGRAMMING

The PLUS16XX Series are programmable on conventional programmers for 20-pin PAL® devices. Refer to the following charts for qualified manufacturers of programmers and software tools:

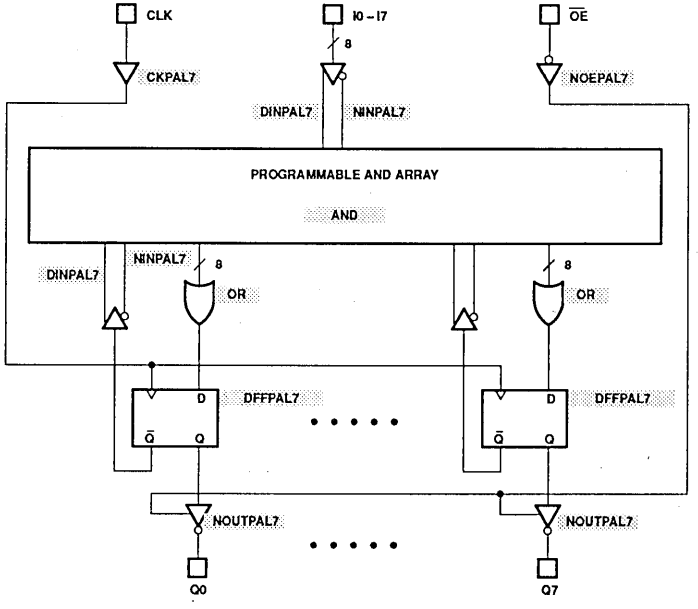
PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	SYSTEM 29B, LogicPak™ 303A–V04 ADAPTER 303A–011A–V08 303A–011B–V04 UNISITE 40/48, V2.3 (DIP) V2.5 (PLCC) MODEL 60, 60A/H, V.15.0	16L8–7/16L8D : 1B/17 16R8–7/16R8D : 1B/24 16R6–7/16R6D : 1B/24 16R4–7/16R4D : 1B/24
STAG MICROSYSTEMS, INC. 1600 WYATT DRIVE SUITE 3 SANTA CLARA, CALIFORNIA 95054 (408)988–1118	ZL30/30A PROGRAMMER REV. 30A31 (DIP) REV. 30A001 (PLCC) PPZ PROGRAMMER TBA	16L8–7/16L8D : 11/29 16R8–7/16R8D : 11/30 16R6–7/16R6D : 11/31 16R4–7/16R4D : 11/32

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
SIGNETICS COMPANY 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SNAP REV 1.6 AND LATER SLICE REV 1.0 AND LATER AMAZE SOFTWARE REV. 1.7 AND LATER
DATA I/O 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	ABEL™ SOFTWARE REV. 1.0 AND LATER
LOGICAL DEVICES, INC. 1201 NORTHWEST 65TH PLACE FORT LAUDERDALE, FLORIDA 33309 (800)331–7766	CUPL™ SOFTWARE REV. 1.01 AND LATER

SNAP RESOURCE SUMMARY DESIGNATIONS

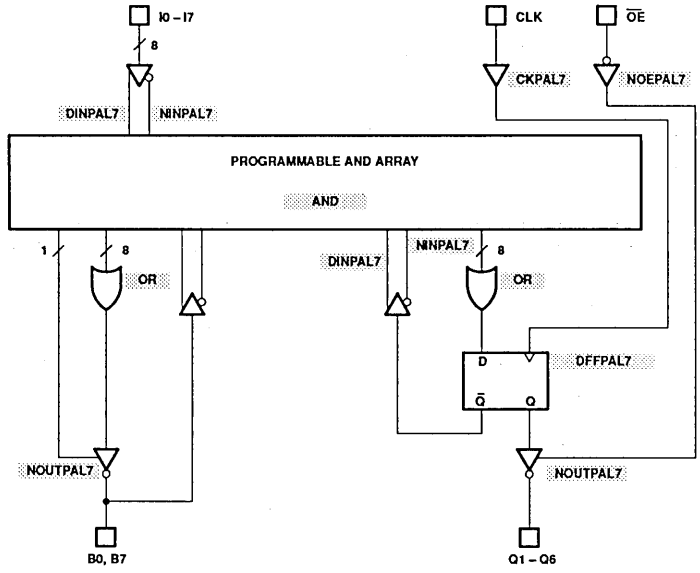


PLUS16L8

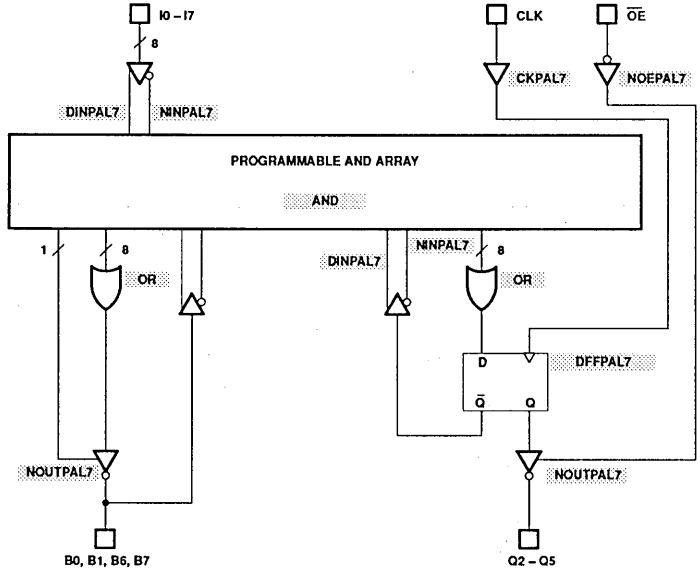


PLUS16R8

SNAP RESOURCE SUMMARY DESIGNATIONS (Continued)



PLUS16R6



PLUS16R4

Document No.	853–1417
ECN No.	99230
Date of Issue	March 27, 1990
Status	Product Specification
Programmable Logic Devices	

PHD16N8–5

Programmable high-speed decoder logic (16 × 16 × 8)

DESCRIPTION

The PHD16N8-5 is an ultra fast Programmable High-speed Decoder featuring a 5ns maximum propagation delay. The architecture has been optimized using Philips Components-Signetics state-of-the-art bipolar oxide isolation process coupled with titanium-tungsten fuses to achieve superior speed in any design.

The PHD16N8-5 is a single level logic element comprised of 10 fixed inputs, 8 AND gates, and 8 outputs of which 6 are bidirectional. This gives the device the ability to have as many as 16 inputs. Individual 3-State control of all outputs is also provided.

The device is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment. Proprietary designs can be protected by programming the security fuse.

The AMAZE software package from Philips Components-Signetics supports easy design entry for the PHD16N8-5 as well as other PLD devices.

Order codes are listed in the pages following.

FEATURES

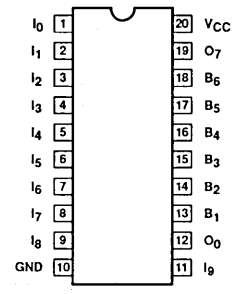
- Ideal for high speed system decoding
- Super high speed at 5ns t_{PD}
- 10 dedicated inputs
- 8 outputs
  - 6 bidirectional I/O
  - 2 dedicated outputs
- Security fuse to prevent duplication of proprietary designs.
- Individual 3-State control of all outputs
- Field-programmable on industry standard programmers
- Available in 20-pin Plastic DIP and 20-Pin PLCC

APPLICATIONS

- High speed memory decoders
- High speed code detectors
- Random logic
- Peripheral selectors
- Machine state decoders
- Footprint compatible to 16L8
- Fuse/Footprint compatible to TIBPAD

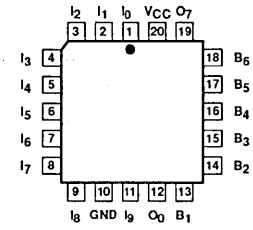
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package

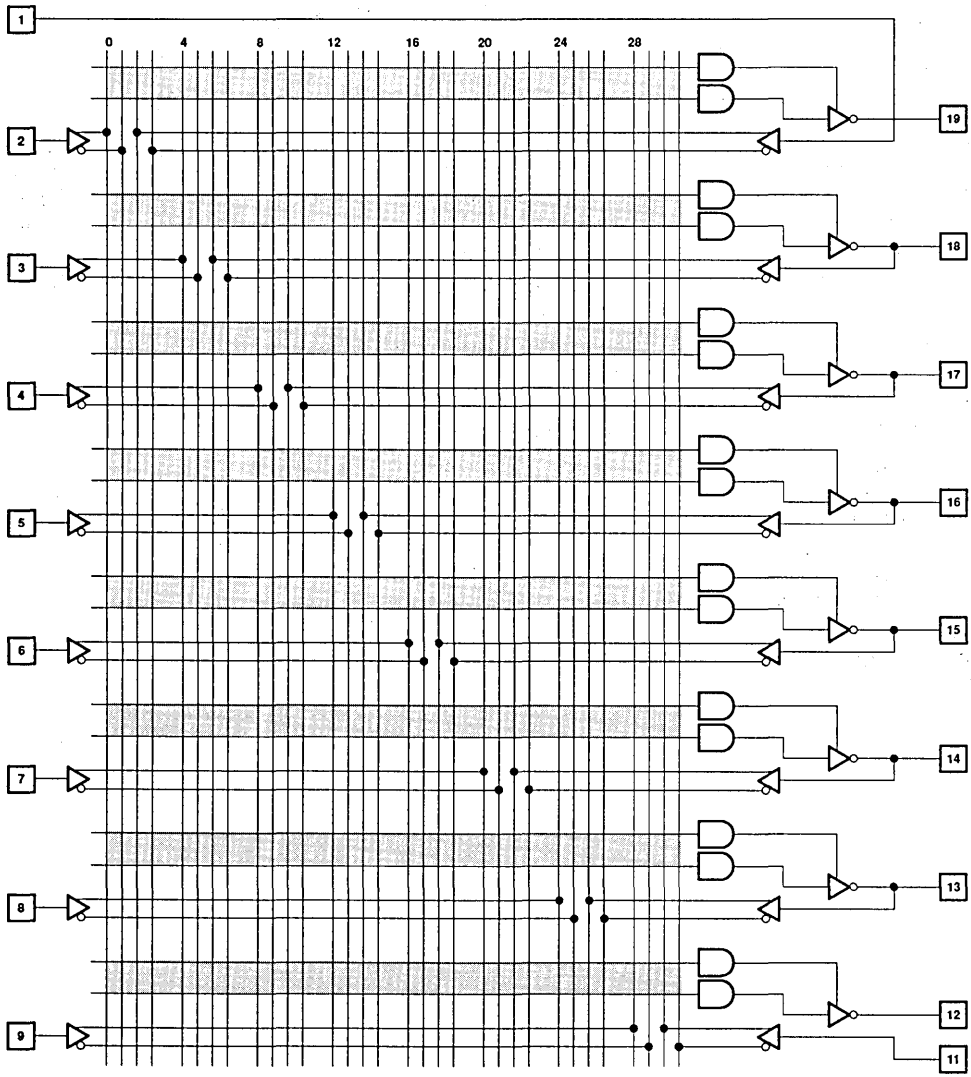


A = Plastic Leaded Chip Carrier

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8–5

LOGIC DIAGRAM

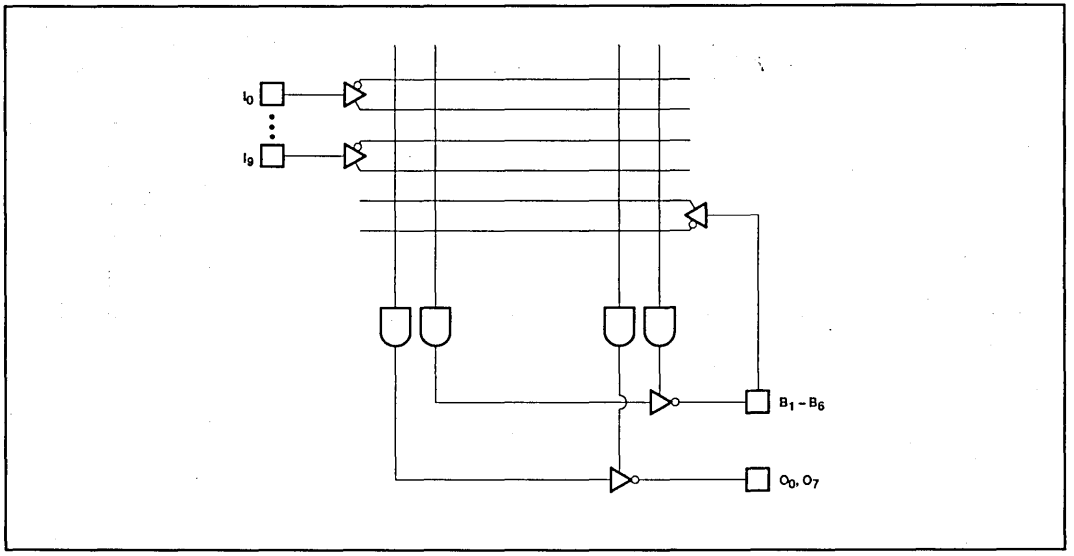


NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒ Programmable connections

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8–5

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic Dual In Line Package; (300mil-wide)	PHD16N8–5N
20-Pin Plastic Leaded Chip Carrier; (350mil square)	PHD16N8–5A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS Min	RATINGS Max	UNIT
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–0.5	+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS Min	RATINGS Max	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8–5

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = +24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = +0.40V		–20	–250	μA
I_{IH}	High	V_{IN} = +2.7V			25	μA
I_I	High	V_{IN} = V_{CC} = V_{CC} MAX			100	μA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage[3]	V_{OUT} = +2.7V			100	μA
I_{OZL}	Output leakage[3]	V_{OUT} = +0.40V			–100	μA
I_{OS}	Short circuit[4]	V_{OUT} = 0V	–30		–90	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		115	180	mA
Capacitance[5]						
		V_{CC} = +5V				
C_{IN}	Input	V_{IN} = 2.0V @ f = 1MHz		8		pF
C_{OUT}	I/O (B)	V_{OUT} = 2.0V @ f = 1MHz		8		pF

NOTES:
1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second.
5. These parameters are not 100% tested, but are periodically sampled.

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8–5

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 200\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS	LIMITS MIN	LIMITS MAX	UNIT
t_{PD}[1]	Propagation delay	(I, B) ±	Output ±	$C_L = 50pF$		5	ns
t_{OE}[2]	Output Enable	(I, B) ±	Output enable	$C_L = 50pF$		10	ns
t_{OD}[2]	Output Disable	(I, B) ±	Input disable	$C_L = 5pF$		10	ns

NOTES:
1. t_{PD} is tested with switch S_1 closed and $C_L = 50pF$.
2. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

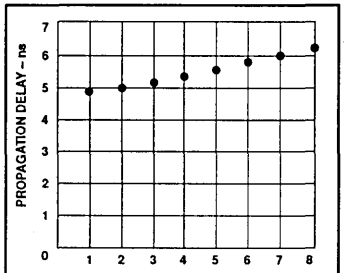
VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All outputs are disabled.
2. All p-terms are disabled in the AND array.

TIMING DEFINITIONS

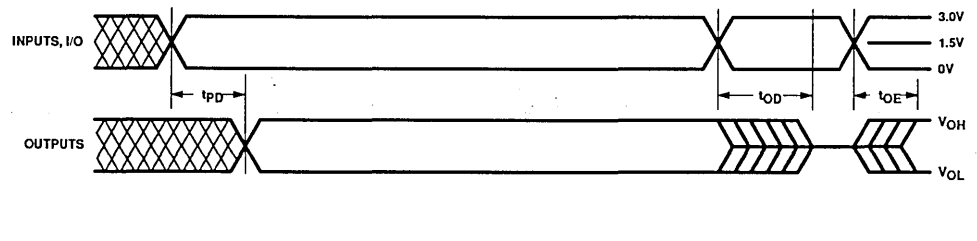
SYMBOL	PARAMETER
t_{PD}	Input to output propagation delay.
t_{OD}	Input to Output Disable (3-State) delay (Output Disable).
t_{OE}	Input to Output Enable delay (Output Enable).



TEST CONDITIONS: $T_{amb} = 75°C$; $V_{CC} = 4.75V$; $C_L = 50pF$; $R_1 = 200\Omega$; $R_2 = 390\Omega$

Worst-Case Propagation Delay vs. Number of Outputs Switching

TIMING DIAGRAM



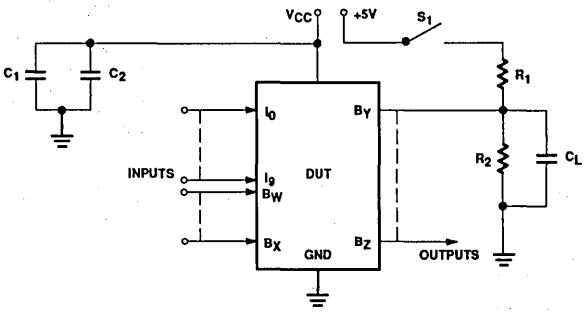
WAVEFORM	INPUTS	OUTPUTS
———	MUST BE STEADY	WILL BE STEADY
———		

WAVEFORM	INPUTS	OUTPUTS
▒▒▒	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
▒▒▒	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

Programmable high-speed decoder logic (16 × 16 × 8)

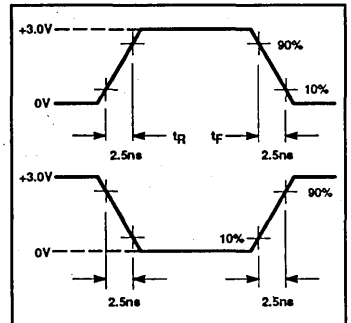
PHD16N8–5

AC TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

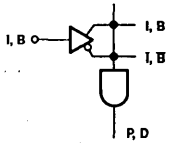
The PHD16N8–5 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PHD16N8–5 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

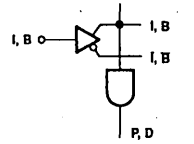
PHD16N8–5 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, each logic variable (I, B, P and D) from the logic equations is assigned a symbol. TRUE (High), COMPLEMENT (Low), DON'T CARE and INACTIVE symbols are defined below.

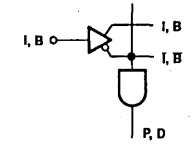
"AND" ARRAY – (I, B)



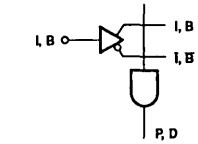
STATE	CODE
INACTIVE[1]	O



STATE	CODE
TRUE	H



STATE	CODE
COMPLEMENT	L



STATE	CODE
DON'T CARE	–

NOTE:
1. This is the initial state.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8–5

PROGRAM TABLE

OR (FIXED)

D	A	
DIRECTION	ACTIVE OUTPUT	NOT USED

AND

I, B(i)

0	H	L	—
INACTIVE	I, B	$\overline{I}$, $\overline{B}$	DON'T CARE

TERM	AND: INPUT (I) 9	8	7	6	5	4	3	2	1	0	INPUTS (B) 6	5	4	3	2	1	OR (FIXED): OUTPUTS (B, O) 7	6	5	4	3	2	1	0
0																	D							
1																	A							
2																		D						
3																		A						
4																			D					
5																			A					
6																				D				
7																				A				
8																					D			
9																					A			
10																						D		
11																						A		
12																							D	
13																							A	
14																								D
15																								A
PIN	11	9	8	7	6	5	4	3	2	1	18	17	16	15	14	13	19	18	17	16	15	14	13	12
VARIABLE NAME																								

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # CF(XXXX)

CUSTOMER SYMBOLIZED PART # _____

TOTAL NUMBER OF PARTS _____

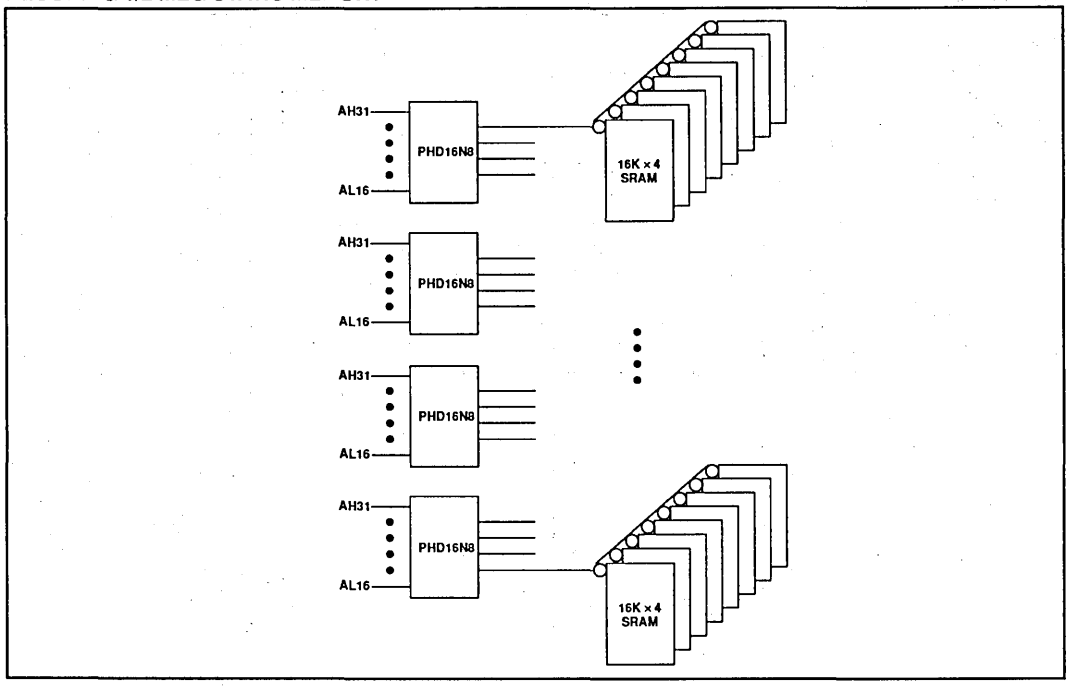
PROGRAM TABLE # _____ REV ___ DATE ___

NOTES:
1. The PHD16N8-5 is shipped with all links intact.
2. Unused I and B bits in the AND array exist as INACTIVE in the virgin state.
3. All p-terms are inactive until programmed otherwise.
4. Data cannot be entered into the OR array field due to the fixed nature of the device architecture.

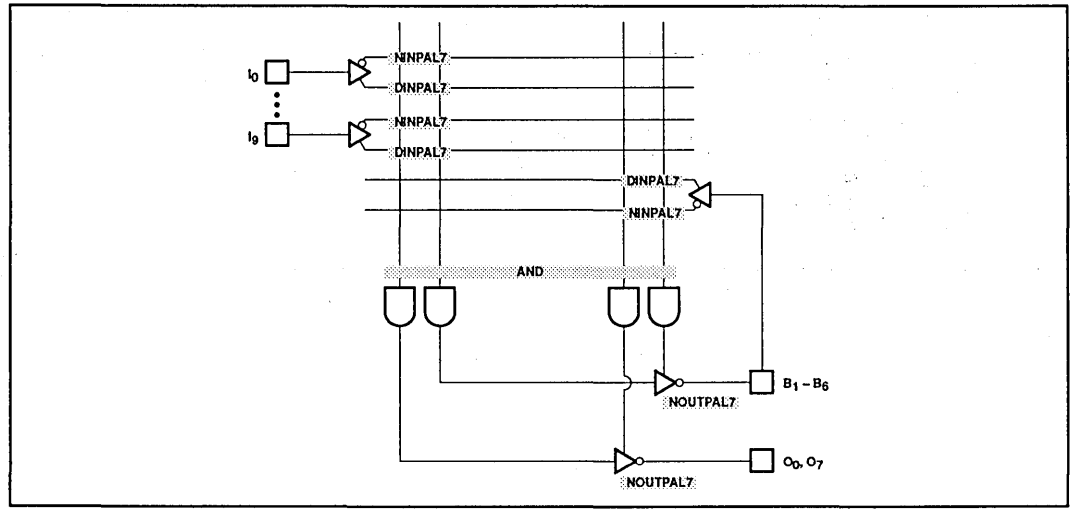
Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8–5

DECODING 1/2 MEG STATIC MEMORY



SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–0863
ECN No.	97886
Date of Issue	October 16, 1989
Status	Product Specification
Programmable Logic Devices	

PLHS18P8A/B

PAL®-type devices

DESCRIPTION

The PLHS18P8A and the PLHS18P8B are two-level logic elements consisting of 72 AND gates and 8 OR gates with fusible connections for programming I/O polarity and direction.

All AND gates are linked to 10 inputs (I) and 8 bidirectional I/O lines (B). These yield variable I/O gate configurations via 8 direction control gates, ranging from 18 inputs to 8 outputs.

On-chip T/C buffers couple either True (I, B) or Complement ($\overline{I}$, $\overline{B}$) input polarities to all AND gates. The 72 AND gates are separated into 8 groups of 9 each. Each group of 9 is associated with one bidirectional pin. In each group, eight of the AND terms are ORed together, while the ninth is used to establish I/O direction. All outputs are individually programmable via an Ex–OR gate to allow implementation of AND/OR or NAND/NOR logic functions.

In the virgin state, the AND array fuses are back-to-back CB-EB diode pairs which will act as open connections. Current is avalanched across individual diode pairs during fusing, which essentially short circuits the EB diode and provides the connection for the associated product term.

The PLHS18P8A/B is field-programmable, allowing the user to quickly generate custom patterns using standard programming equipment.

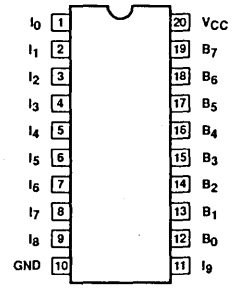
Order codes are listed in the Ordering Information Table.

FEATURES

- "A" version 100% functionally compatible with AmPAL18P8A and all 16L8, 16P8, 16H8, 16L2, 16H2, 14L4, 14H4, 12L6, 12H6, 10L8, 10H8, 16LD8 and 16HD8 "A" speed PAL-type products
- "B" version 100% functionally compatible with AmPAL18P8B and all 16L8, 16P8, 16H8, 16L2, 16H2, 14L4, 14H4, 12L6, 12H6, 10L8, 10H8, 16LD8 and 16HD8 "B" speed PAL-type products
- Field-programmable
- 10 inputs
- 8 bidirectional I/O lines
- 72 AND gates/product terms
  - configured into eight groups of nine
- Programmable output polarity (3-State output)
- I/O propagation delay:
  - PLHS18P8A: 20ns (max)
  - PLHS18P8B: 15ns (max)
- Power dissipation: 500mW (typ)
- TTL compatible
- Security fuse

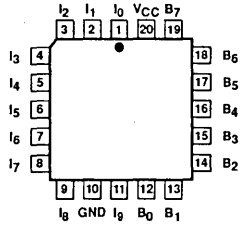
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



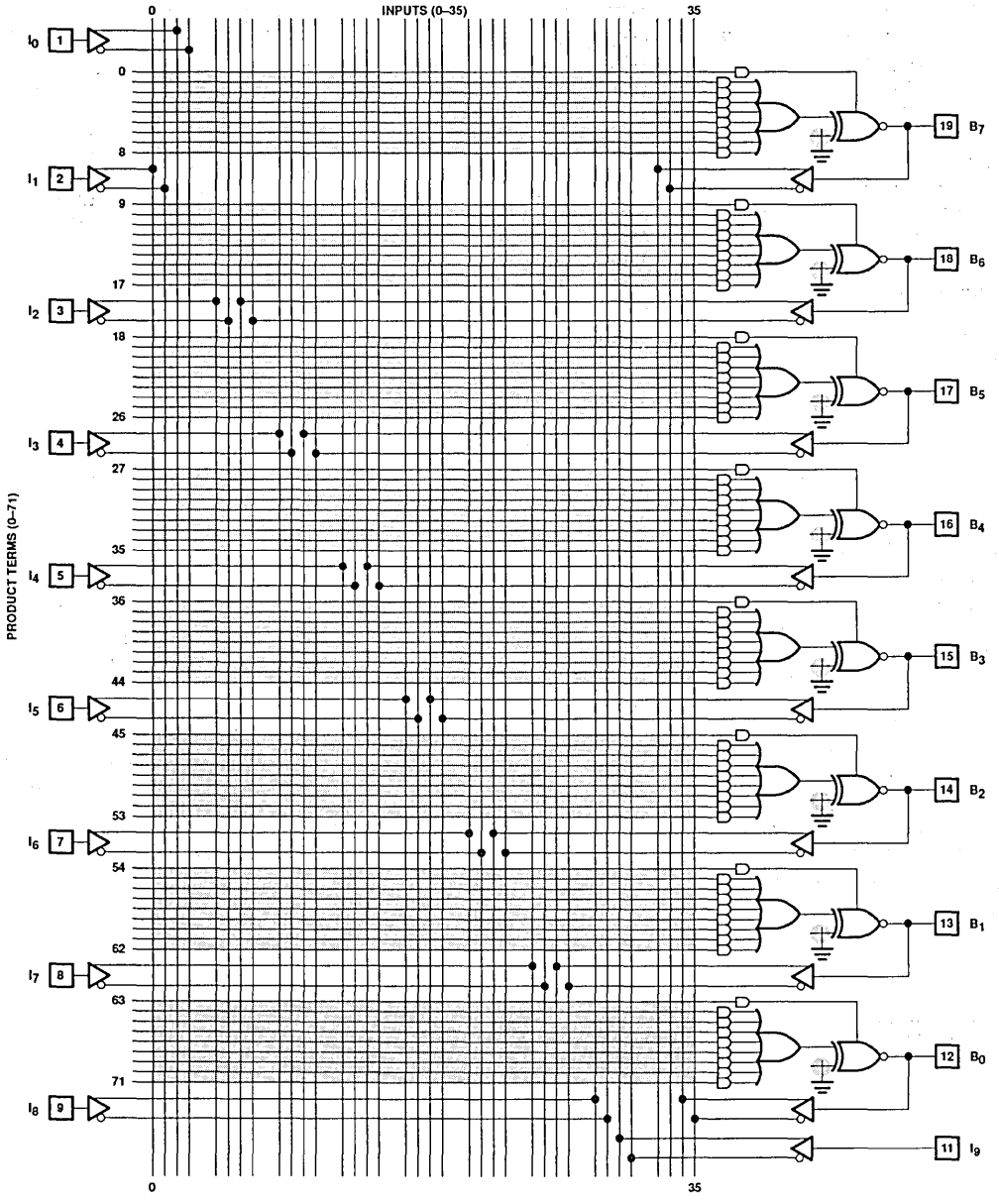
A = Plastic Leaded Chip Carrier

APPLICATIONS

- 100% functional replacement for all 20-pin combinatorial PAL devices
- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

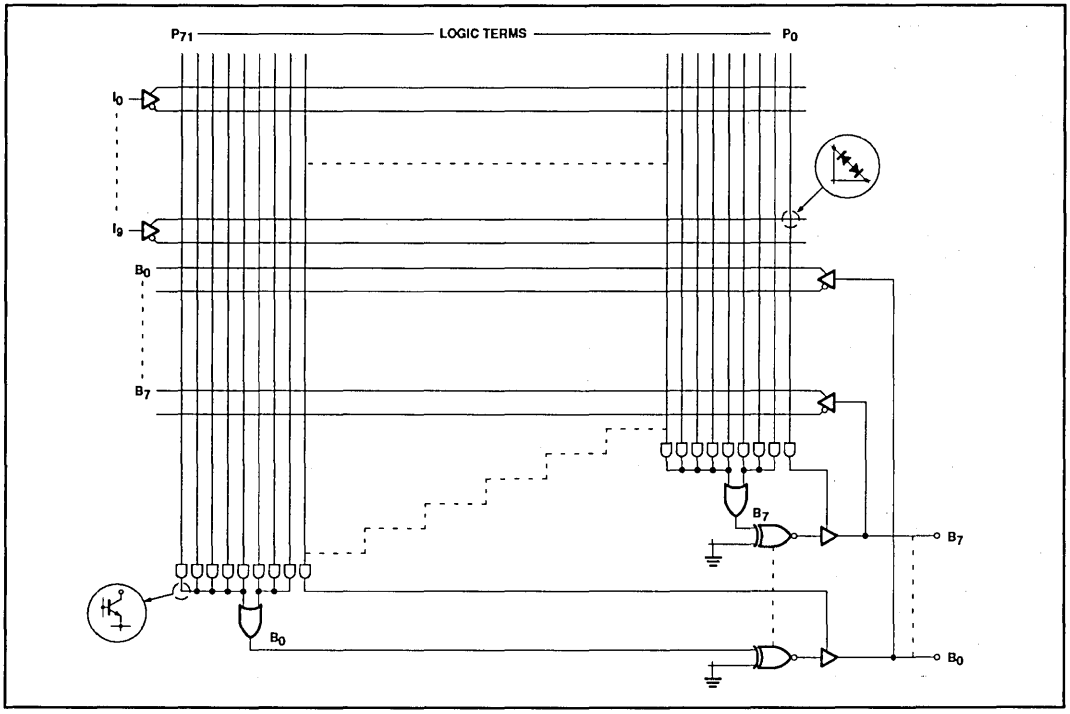
®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

LOGIC DIAGRAM



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "1".
2. ⊗ Programmable connections.

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic Dual In–Line (300mil-wide)	PLHS18P8AN, PLHS18P8BN
20-Pin Plastic Leaded Chip Carrier	PLHS18P8AA, PLHS18P8BA

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	–0.5 to +7	V_{DC}
V_{IN}	Input voltage	–0.5 to +5.5	V_{DC}
V_{OUT}	Output voltage	–0.5 to V_{CC} Max	V_{DC}
V_{OUTPRG}	Output voltage (programming)	+21	V_{DC}
I_{IN}	Input current	–30 to +5	mA
I_{OUT}	Output current	+100	mA
I_{OUTPRG}	Output current (programming)	+170	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS						UNIT
			PLHS18P8A			PLHS18P8B			
			MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Input voltage[2]									
V_{IL}	Low	V_{CC} = MIN			+0.8			+0.8	V
V_{IH}	High	V_{CC} = MAX	+2.0			+2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −18mA		−0.9	−1.2		−0.9	−1.2	V
Output voltage									
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}							
V_{OL}	Low	I_{OL} = +24mA			+0.50			+0.50	V
V_{OH}	High	I_{OH} = −3.2mA	+2.4	+3.5		+2.4	+3.5		V
Input current									
		V_{CC} = MAX							
I_{IL}	Low	V_{IN} = +0.40V		−20	−100		−20	−100	µA
I_{IH}	High	V_{IN} = +2.7V			+25			+25	µA
I_I	High	V_{IN} = +5.5V			+1.0			+1.0	mA
Output current									
		V_{CC} = MAX, V_{IL} = 0.8V, V_{IH} = 2.0V							
I_{OZH}	Output leakage	V_{OUT} = +2.7V			+100			+100	µA
I_{OZL}	Output leakage	V_{OUT} = +0.40V			−250			−250	µA
I_{OS}	Short circuit[3]	V_{OUT} = +0.5V	−25	−60	−90	−30	−60	−90	mA
I_{CC}	V_{CC} current	V_{CC} = MAX, All inputs = GND		100	155		100	155	mA
Capacitance[4]									
		V_{CC} = +5V							
C_{IN}	Input	V_{IN} = 2.0V @ f = 1MHz		6			6		pF
C_{OUT}	I/O	V_{OUT} = 2.0V @ f = 1MHz		9			9		pF

NOTES:

1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
3. Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second. V_{OUT} = 0.5V has been chosen to avoid test problems caused by tester ground degradation.
4. These parameters are not 100% tested, but are periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 200\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS	LIMITS						UNIT
					PLHS18P8A			PLHS18P8B			
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PD}	Propagation delay	Output ±	Input ±	$C_L = 50pF$		14	20		12	15	ns
t_{EA}	Output enable[3]	Output –	Input ±	$C_L = 50pF$		14	20		12	15	ns
t_{ER}	Output disable[3]	Output +	Input ±	$C_L = 5pF$		14	20		12	15	ns

NOTES:
1. Typical limits are at $V_{CC} = 5.0V$ and $T_{amb} = +25°C$.
2. t_{PD} is tested with switch S_1 closed and $C_L = 50pF$.
3. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

VIRGIN STATE

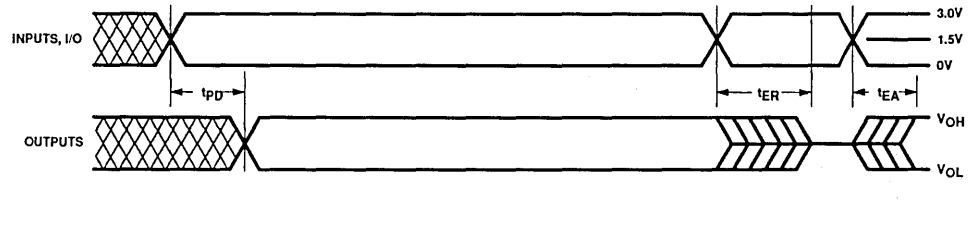
A factory shipped virgin device contains all fusible links open, such that:

1. All outputs are at "H" polarity.
2. All outputs are enabled.
3. All p-terms are enabled.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Input to output propagation delay.
t_{ER}	Input to output disable (3-State) delay (Output Disable).
t_{EA}	Input to Output Enable delay (Output Enable).

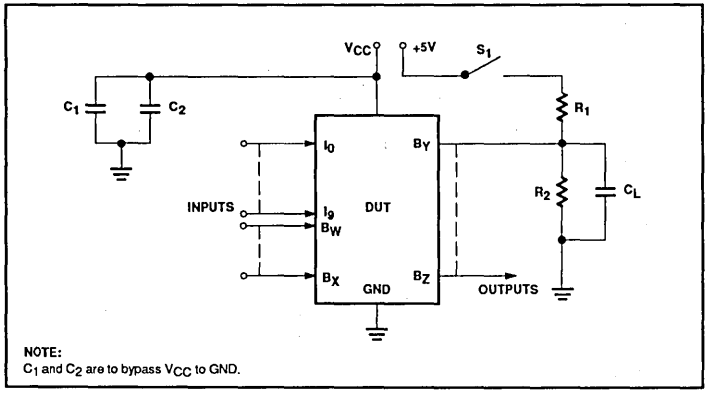
TIMING DIAGRAM



WAVEFORM	INPUTS	OUTPUTS
———	MUST BE STEADY	WILL BE STEADY
———		

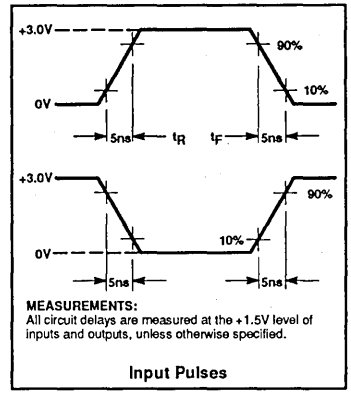
WAVEFORM	INPUTS	OUTPUTS
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

AC TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

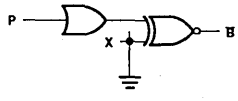
The PLHS18P8A/B is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE design software package. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLHS18P8A/B architecture.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

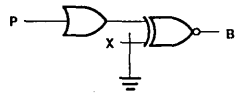
PLHS18P8A/B logic designs can also be generated using the program table format detailed on the following pages. This program table entry (PTE) format is supported by the Signetics' AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

OUTPUT POLARITY – (B)

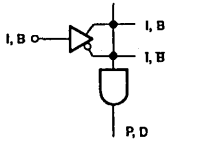


ACTIVE LEVEL	CODE
LOW	L

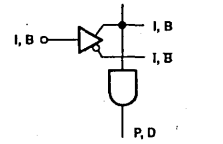


ACTIVE LEVEL	CODE
HIGH[1]	H

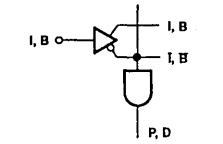
"AND" ARRAY – (I, B)



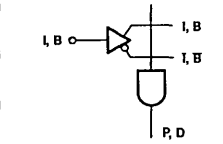
STATE	CODE
INACTIVE[2]	O



STATE	CODE
I, B	H



STATE	CODE
I, B	L



STATE	CODE
DON'T CARE[1]	–

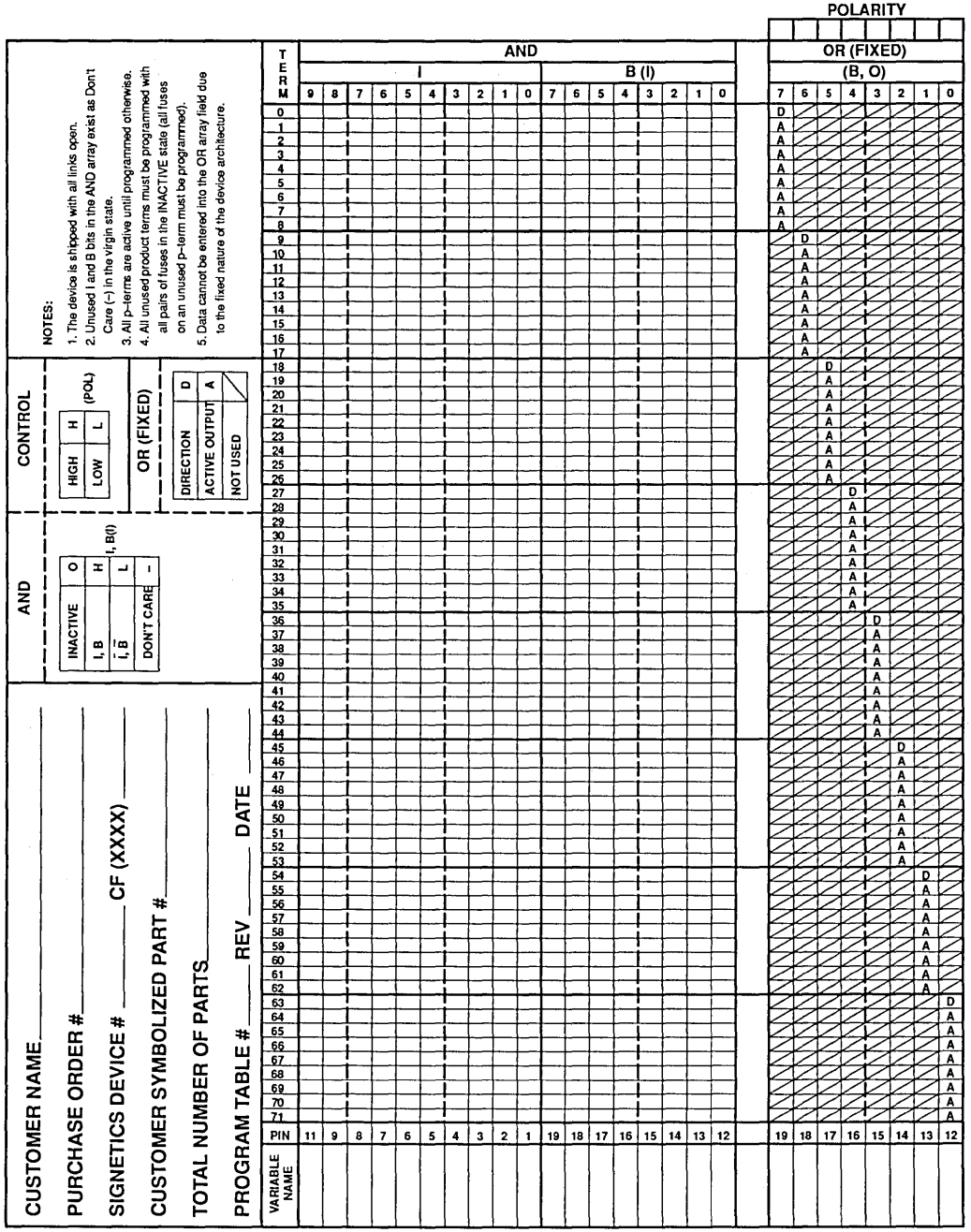
NOTE:
1. This is the initial state of all link pairs.
2. All unused product terms must be programmed with all pairs of fuses in the INACTIVE state (all fuses on an unused p-term must be programmed).

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

PAL-type devices

PLHS18P8A/B

PROGRAM TABLE



Document No.	853–1396
ECN No.	97550
Date of Issue	September 1, 1989
Status	Product Specification
Programmable Logic Devices	

PLC18V8Z35 / PLC18V8ZI

Zero standby power universal PAL®-type devices

DESCRIPTION

The PLC18V8Z35 and PLC18V8ZI are universal PAL-type devices featuring high performance and virtually zero-standby power for power sensitive applications. They are reliable, user-configurable substitutes for discrete TTL/CMOS logic. While compatible with TTL and HCT logic, the PLC18V8ZI can also replace HC logic over the V_{CC} range of 4.5 to 5.5V.

The PLC18V8Z is a two-level logic element comprised of 10 inputs, 74 AND gates (product terms) and 8 output Macro cells.

Each output features an "Output Macro Cell" which can be individually configured as a dedicated input, a combinatorial output, or a registered output with internal feedback. As a result, the PLC18V8Z is capable of emulating all common 20-pin PAL devices to reduce documentation, inventory, and manufacturing costs.

A power-up reset function and a Register Preload function have been incorporated in the PLC18V8Z architecture to facilitate state machine design and testing.

With a standby current of less than 100μA and active power consumption of 1.5mA/MHz, the PLC18V8Z is ideally suited for power sensitive applications in battery operated/backed portable instruments and computers.

The PLC18V8Z is also processed to industrial requirements for operation over an extended temperature range of -40°C to +85°C and supply voltage of 4.5V to 5.5V.

Ordering information can be found in the Ordering Information table.

FEATURES

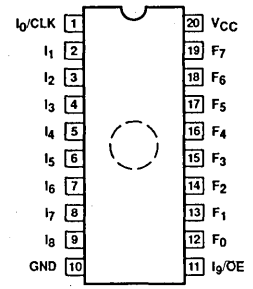
- 20-pin Universal Programmable Array Logic
- Virtually Zero-Standby-power
- Functional replacement for Series 20 PAL devices
  - I_{OL} = 24mA
- High-performance CMOS EPROM cell technology
  - Erasable
  - Reconfigurable
  - 100% testable
- 35ns Max propagation delay (comm)
- 40ns Max propagation delay (Industrial)
- Up to 18 inputs and 8 input/output macro cells
- Programmable output polarity
- Power-up reset on all registers
- Register Preload capability
- Synchronous Preset/Asynchronous Reset
- Security fuse to prevent duplication of proprietary designs
- Design support provided using AMAZE software development package and other CAD tools for PLDs
- Available in 300mil-wide DIP with quartz window, plastic DIP (OTP) or PLCC (OTP)

APPLICATIONS

- Battery powered instruments
- Laptop and pocket computers
- Industrial control
- Medical Instruments
- Portable communications equipment

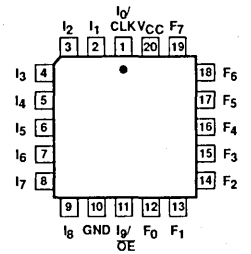
PIN CONFIGURATIONS

N and FA Packages



N = Plastic DIP (300mil-wide)
FA = Ceramic DIP with Quartz Window (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

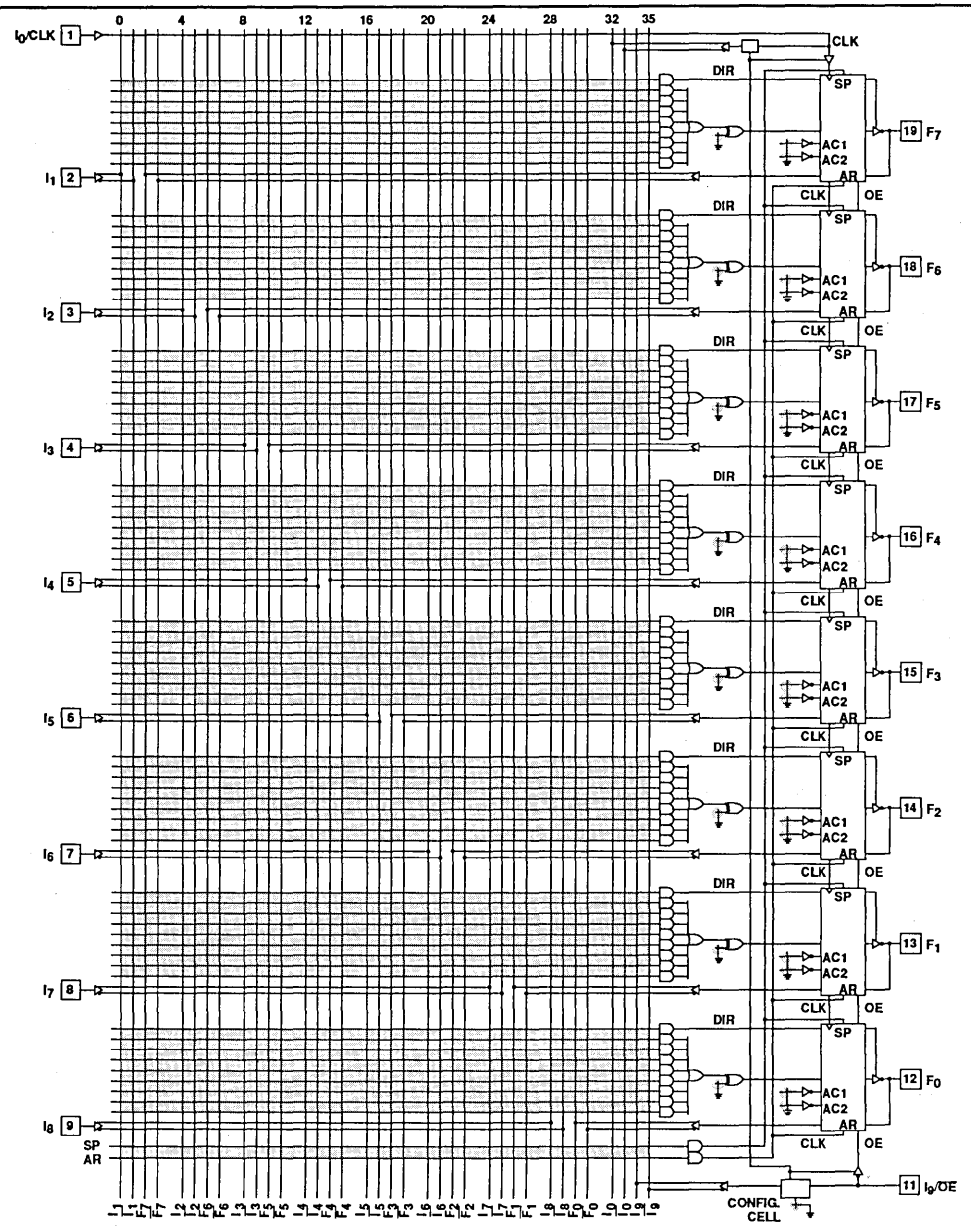
PIN LABEL DESCRIPTIONS

I	Dedicated input
B	Bidirectional input/output
O	Dedicated output
D	Registered output (D-type flip-flop)
F	Macrocell Input/Output
CLK	Clock Input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

Zero standby power universal PAL-type devices

PLC18V8Z Series

LOGIC DIAGRAM



NOTES:

In the unprogrammed or virgin state:

All cells are in a conductive state.
All AND gate locations are pulled to a logic "0" (Low).
Output polarity is inverting.

Pins 1 and 11 are configured as Inputs 0 and 9, respectively, via the configuration cell. The clock and $\overline{OE}$ functions are disabled.
All output macro cells (OMC) are configured as bidirectional I/O, with the outputs disabled via the direction term.
Denotes a programmable cell location.

Zero standby power universal PAL-type devices

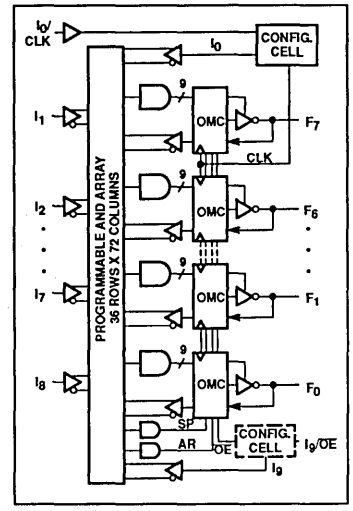
PLC18V8Z Series

PAL DEVICE TO PLC18V8Z OUTPUT PIN CONFIGURATION CROSS REFERENCE

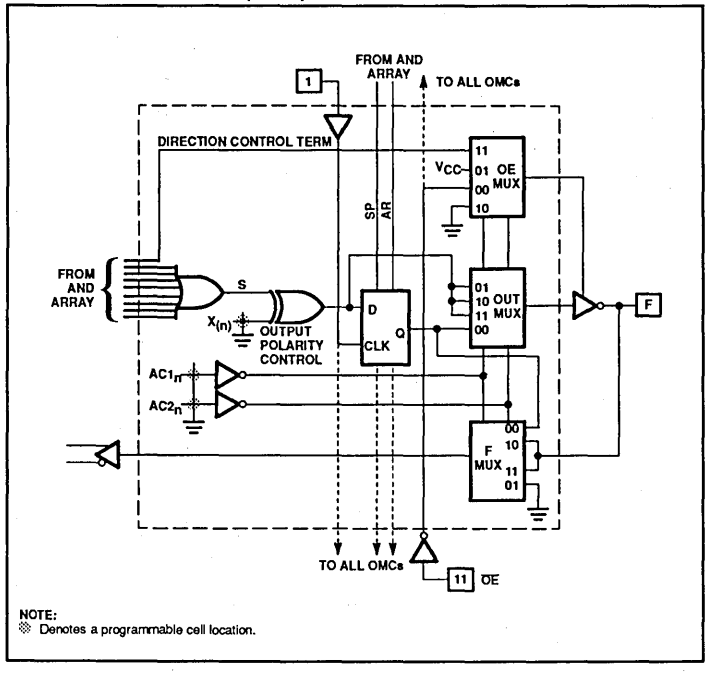
PIN NO.	PLC 18V8Z	16L8 16H8 16P8 16P8	16R4 16RP4	16R6 16RP6	16R8 16RP8	16L2 16H2 16P2	14L4 14H4 14P4	12L6 12H6 12P6	10L8 10H8 10P8
1	I_0/CLK	I	CLK	CLK	CLK	I	I	I	I
19	F7	B	B	B	D	I	I	I	O
18	F6	B	B	D	D	I	I	O	O
17	F5	B	D	D	D	I	O	O	O
16	F4	B	D	D	D	O	O	O	O
15	F3	B	D	D	D	O	O	O	O
14	F2	B	D	D	D	I	O	O	O
13	F1	B	B	D	D	I	I	O	O
12	F0	B	B	B	D	I	I	I	O
11	I_9/$\overline{OE}$	I	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	I	I	I	I

The Signetics state-of-the-art Floating-Gate CMOS EPROM process yields bipolar equivalent performance at less than one-quarter the power consumption. The erasable nature of the EPROM process enables Signetics to functionally test the devices prior to shipment to the customer. Additionally, this allows Signetics to extensively stress test, as well as ensure the threshold voltage of each individual EPROM cell. 100% programming yield is subsequently guaranteed.

FUNCTIONAL DIAGRAM



OUTPUT MACRO CELL (OMC)



NOTE:
▒ Denotes a programmable cell location.

THE OUTPUT MACRO CELL (OMC)

The PLC18V8Z series devices have 8 individually programmable Output Macro Cells. The 72 AND inputs (or product terms) from the programmable AND array are connected to the 8 OMCs in groups of 9. Eight of the AND terms are dedicated to logic functions; the ninth is for asynchronous direction control, which enables/disables the respective bidirectional I/O pin. Two product terms are dedicated for the Synchronous Preset and Asynchronous Reset functions.

Each OMC can be independently programmed via 16 architecture control bits, $AC1_n$ and $AC2_n$ (one pair per macro cell). Similarly, each OMC has a programmable output polarity control bit (Xn). By configuring the pair of architecture control bits according to the configuration cell table, 4 different configurations may be implemented. Note that the configuration cell is automatically programmed based on the OMC configuration.

DESIGN SECURITY

The PLC18V8Z series devices have a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

Zero standby power universal PAL-type devices

PLC18V8Z Series

CONFIGURATION CELL

A single configuration cell controls the functions of Pins 1 and 11. Refer to Functional Diagram. When the configuration cell is programmed, Pin 1 is a dedicated clock and Pin 11 is dedicated for output enable. When the configuration cell is unprogrammed, Pins 1 and 11 are both dedicated inputs. Note that the output enable for all registered OMCs is common—from Pin 11 only. Output enable control of the bidirectional I/O OMCs is provided from the AND array via the direction product term.

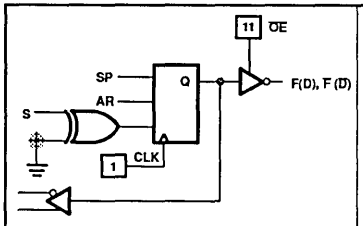
If any one OMC is configured as registered, the configuration cell will be automatically configured (via the design software) to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively. If none of the OMCs are registered, the configuration cell will be programmed such that Pins 1 and 11 are dedicated inputs. The programming codes are as follows:

Pin 1 = CLK, Pin 11 = $\overline{OE}$	L
Pin 1 and Pin 11 = Input	H

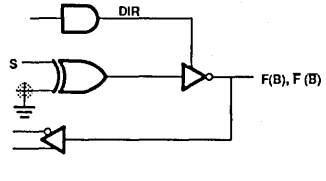
FUNCTION	CONTROL CELL CONFIGURATIONS			COMMENTS
	$AC1_1$	$AC2_N$	CONFIG. CELL	
Registered mode	Programmed	Programmed	Programmed	Dedicated clock from Pin 1. $\overline{OE}$ Control for all registerd OMCs from Pin 11 only.
Bidirectional I/O mode[1]	Unprogrammed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. 3-State control from AND array only.
Fixed input mode	Unprogrammed	Programmed	Unprogrammed	Pins 1 and 11 are dedicated inputs.
Fixed output mode	Programmed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. The feedback path (via F_{MUX}) is disabled.

NOTE:
1. This is the virgin state as shipped from the factory.

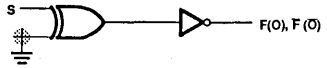
ARCHITECTURE CONTROL—AC1 and AC2



OMC CONFIGURATION	CODE
REGISTERED (D-TYPE)	D

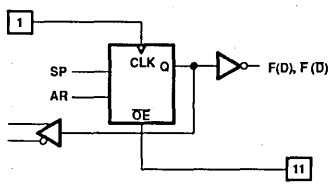


OMC CONFIGURATION	CODE
BIDIRECTIONAL I/O[1] (COMBINATORIAL)	B

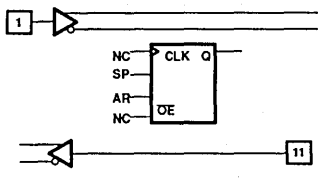


OMC CONFIGURATION	CODE
FIXED OUTPUT	O

OMC CONFIGURATION	CODE
FIXED INPUT	I



CONFIGURATION CELL	CODE
PIN 1 = CLK PIN 11 = $\overline{OE}$	L



CONFIGURATION CELL	CODE
PIN 1 = INPUT PIN 11 = INPUT	H[6]

NOTE:
A factory shipped unprogrammed device is configured such that:
1. This is the initial unprogrammed state. All cells are in a conductive state.
2. All AND gates are pulled to a logic "0" (Low).
3. Output polarity is inverting.
4. Pins 1 and 11 are configured as inputs 0 and 9. The clock and $\overline{OE}$ functions are disabled.
5. All Output Macro Cells (OMCs) are configured as bidirectional I/O, with the outputs disabled via the direction term.
6. This configuration cannot be used if any OMCs are configured as registered (Code = D). The configuration cell will be automatically configured to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively, if any one OMC is programmed as registered.

Zero standby power universal PAL-type devices

PLC18V8Z Series

ORDERING INFORMATION

DESCRIPTION	OPERATING CONDITIONS	ORDER CODE
20-Pin Plastic Dual In-Line Package 300mil-wide (t_{PD} = 35ns)	Commercial Temperature Range ± 5% Power Supplies	PLC18V8Z35N
20-Pin Ceramic Dual In-Line Package 300mil-wide with quartz window (t_{PD} = 35ns)		PLC18V8Z35FA
20-Pin Plastic Leaded Chip Carrier 350mil square (t_{PD} = 35ns)		PLC18V8Z35A
20-Pin Plastic Dual In-Line Package 300mil-wide (t_{PD} = 40ns)	Industrial Temperature Range ± 10% Power Supplies	PLC18V8ZIN
20-Pin Ceramic Dual In-Line Package 300mil-wide with quartz window (t_{PD} = 40ns)		PLC18V8ZIFA
20-Pin Plastic Leaded Chip Carrier 350mil square (t_{PD} = 40ns)		PLC18V8ZIA

ABSOLUTE MAXIMUM RATINGS[1]

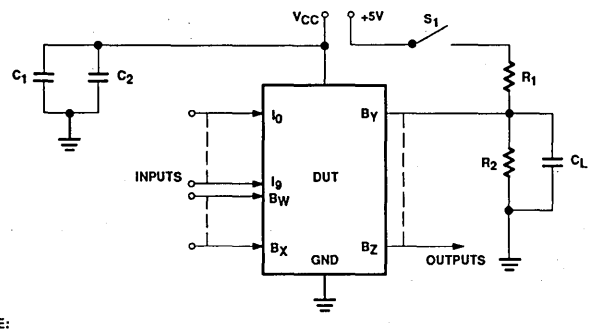
SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	–0.5 to +7	V_{DC}
V_{CC}	Operating supply voltage	4.5 to 5.5 (Industrial) 4.75 to 5.25 (Commercial)	V_{DC}
V_{IN}	Input voltage	–0.5 to V_{CC} + 0.5	V_{DC}
V_{OUT}	Output voltage	–0.5 to V_{CC} + 0.5	V_{DC}
I_{IN}	Input currents	–10 to +10	mA
I_{OUT}	Output currents	+24	mA
T_{amb}	Operating temperature range	–40 to +85 (Industrial) 0 to +75 (Commercial)	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:
1. Stresses above those listed may cause malfuncion or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

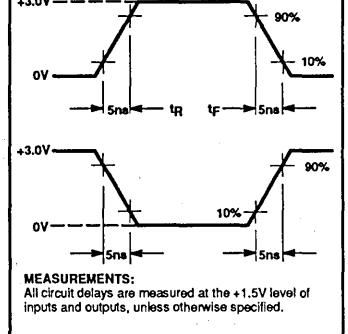
TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

AC TEST CONDITIONS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

Zero standby power universal PAL-type devices

PLC18V8Z Series

DC ELECTRICAL CHARACTERISTICS

Commercial = 0°C ≤ T_{AMB} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V;
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 20µA V_{CC} = MIN, I_{OL} = 24mA			0.100 0.500	V V
V_{OH}	High	V_{CC} = MIN, I_{OH} = –3.2mA V_{CC} = MIN, I_{OH} = –20µA	2.4 V_{CC} – 0.1V			V V
Input current						
I_{IL}	Low[7]	V_{IN} = GND			–10	µA
I_{IH}	High	V_{IN} = V_{CC}			10	µA
Output current						
$I_{O(OFF)}$	Hi–Z state	V_{OUT} = V_{CC} V_{OUT} = GND			10 –10	µA µA
I_{OS}	Short-circuit[3]	V_{OUT} = GND			–130	mA
I_{CC}	V_{CC} supply current (Standby)	V_{CC} = MAX, V_{IN} = 0 or V_{CC}[8]			100	µA
I_{CC}/f	V_{CC} supply current (Active)[4]	V_{CC} = MAX (CMOS inputs)[5,6]			1.5	mA/MHz
Capacitance						
C_I	Input	V_{CC} = 5V V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short-circuit should not exceed one second. Test one at a time.
4. Tested with TTL input levels: V_{IL} = 0.45V, V_{IH} = 2.4V. Measured with all outputs switching.
5. ΔI_{CC}/TTL input = 2mA.
6. ΔI_{CC} vs frequency (registered configuration) = 2mA/MHz.
7. I_{IL} for Pin 1 (I_0/CLK) is ± 10µA with V_{IN} = 0.4V.
8. V_{IN} includes CLK and $\overline{OE}$ if applicable.

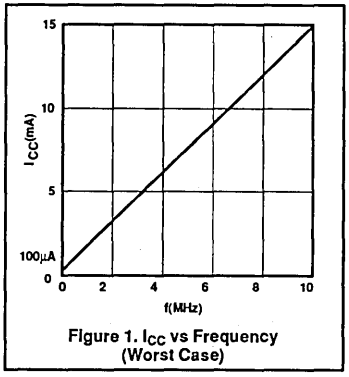


Figure 1. I_{CC} vs Frequency (Worst Case)

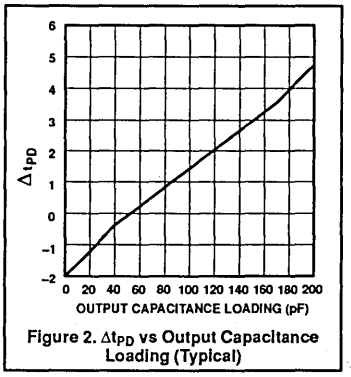


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

AC ELECTRICAL CHARACTERISTICS

Commercial = 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V;
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V; R_2 = 390Ω

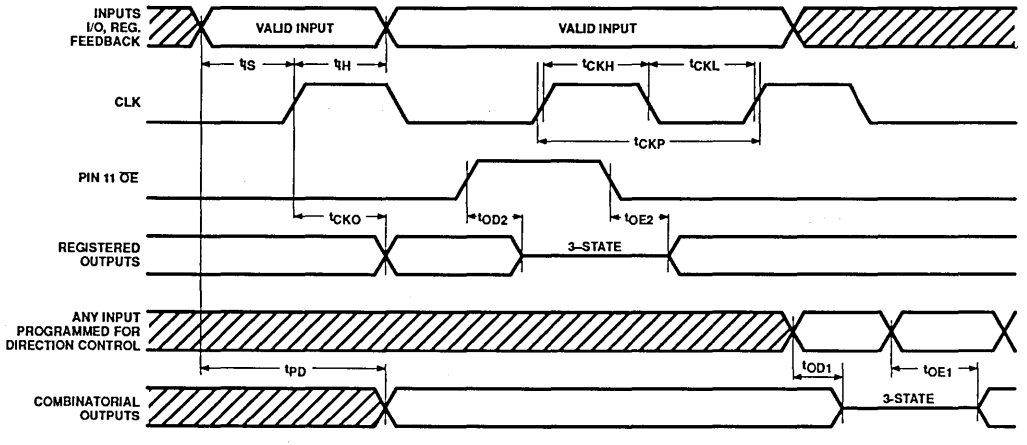
SYMBOL	PARAMETER	FROM	TO	TEST CONDITION[1]		PLC18V8Z35 (Commercial)		PLC18V8ZI (Industrial)		UNIT
				R_1 (Ω)	C_L (pF)	MIN	MAX	MIN	MAX	
Pulse width										
t_{CKP}	Clock period (Minimum $t_{IS} + t_{CKO}$)	CLK +	CLK +	200	50	47		57		ns
t_{CKH}	Clock width High	CLK +	CLK –	200	50	20		25		ns
t_{CKL}	Clock width Low	CLK –	CLK +	200	50	20		25		ns
t_{ARW}	Async reset pulse width	I ±, F±	I $\overline{+}$, F $\overline{+}$			35		40		ns
Hold time										
t_{IH}	Input or feedback data hold time	CLK +	Input ±	200	50	0		0		ns
Setup time										
t_{IS}	Input or feedback data setup time	I ±, F±	CLK +	200	50	25		30		ns
Propagation delay										
t_{PD}	Delay from input to active output	I ±, F±	F±	200	50		35		40	ns
t_{CKO}	Clock High to output valid access Time	CLK +	F±	200	50		22		27	ns
t_{OE1}[3]	Product term enable to outputs off	I ±, F±	F±	Active-High R = 1.5k Active-Low R = 550	50		35		40	ns
t_{OD1}[2]	Product term disable to outputs off	I ±, F±	F±	From V_{OH} R = ∞ From V_{OL} R = 200	5		35		40	ns
t_{OD2}[2]	Pin 11 output disable High to outputs off	OE –	F±	From V_{OH} R = ∞ From V_{OL} R = 200	5		25		30	ns
t_{OE2}[3]	Pin 11 output enable to active output	OE +	F±	Active-High R = 1.5k Active-Low R = 550	50		25		30	ns
t_{ARU}	Async reset delay	I ±, F±	F +				35		40	ns
t_{ARR}	Async reset recovery time	I ±, F±	CLK +			25		30		ns
t_{SPR}	Sync preset recovery time	I ±, F±	CLK +			25		30		ns
t_{PPR}	Power-up reset	V_{CC} +	F +				35		40	ns
Frequency of operation										
f_{MAX}	Maximum frequency	I/($t_{IS} + t_{CKO}$)		200	50		21		18	MHz

NOTES:
1. Refer also to AC Test Conditions. (Test Load Circuit)
2. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} – 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. Resistor values of 1.5k and 550Ω provide 3-State levels of 1.0V and 2.0V, respectively. Output timing measurements are to 1.5V level.

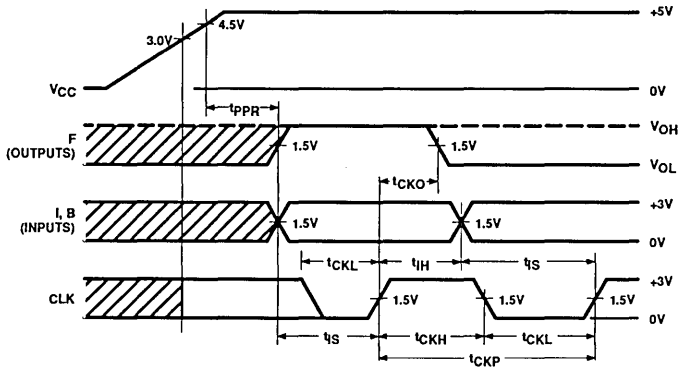
POWER-UP RESET

In order to facilitate state machine design and testing, a power-up reset function has been incorporated in the PLC18V8Z. All internal registers will reset to Active-Low (logical "0") after a specified period of time (t_{PPR}). Therefore, any OMC that has been configured as a registered output will always produce an Active-High on the associated output pin because of the inverted output buffer. The internal feedback (Q) of a registered OMC will also be set Low. The programmed polarity of OMC will not affect the Active-High output condition during a system power-up condition.

TIMING DIAGRAMS



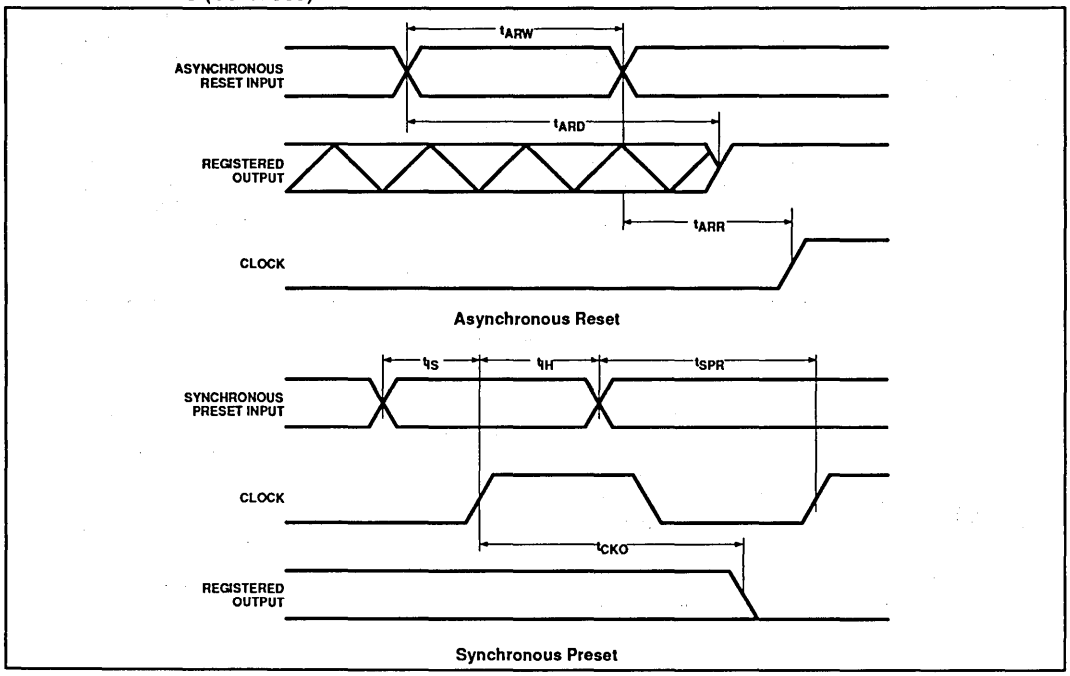
Switching Waveforms



NOTE:
Diagram presupposes that the outputs (F) are enabled. The reset occurs regardless of the output condition (enabled or disabled).

Power-Up Reset

TIMING DIAGRAMS (Continued)



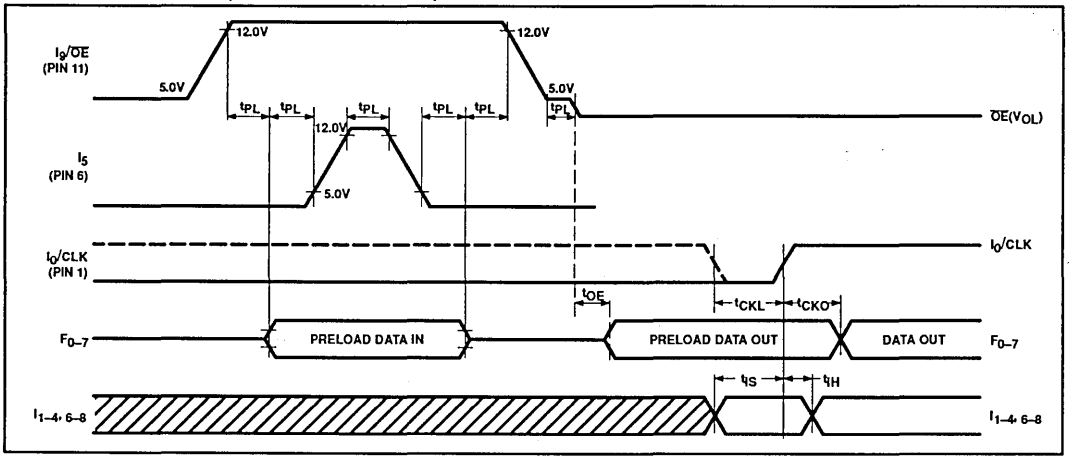
REGISTER PRELOAD FUNCTION (DIAGNOSTIC MODE ONLY)

In order to facilitate the testing of state machine/controller designs, a diagnostic mode register preload feature has been incorporated into the PLC18V8Z series device. This feature enables the user to load the registers with predetermined states while a super voltage is applied to Pins 11 and 6 ($I_9/\overline{OE}$ and I_5). (See diagram for timing and sequence.)

To read the data out, Pins 11 and 6 must be returned to normal TTL levels. The outputs, F_{0-7}, must be enabled in order to read data out. The Q outputs of the registers will reflect data in as input via F_{0-7} during preload. Subsequently, the register Q output via the feedback path will reflect the data in as input via F_{0-7}.

Refer to the voltage waveform for timing and voltage references. t_{PL} = 10µsec.

REGISTER PRELOAD (DIAGNOSTIC MODE)



Zero standby power universal PAL-type devices

PLC18V8Z Series

LOGIC PROGRAMMING

The PLC18V8Z series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLC18V8Z architecture.

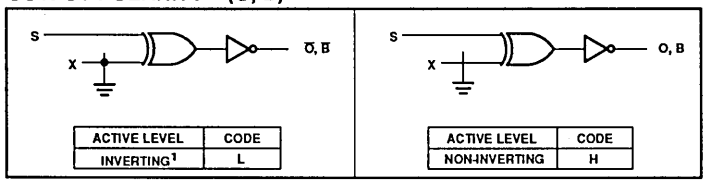
All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC18V8Z logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

With Logic programming, the AND/OR/Ex-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table. Similarly, various OMC configurations are implemented by programming the Architecture Control bits AC1 and AC2. Note that the configuration cell is automatically programmed based on the OMC configuration.

In this table, the logic state of variables I, P and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

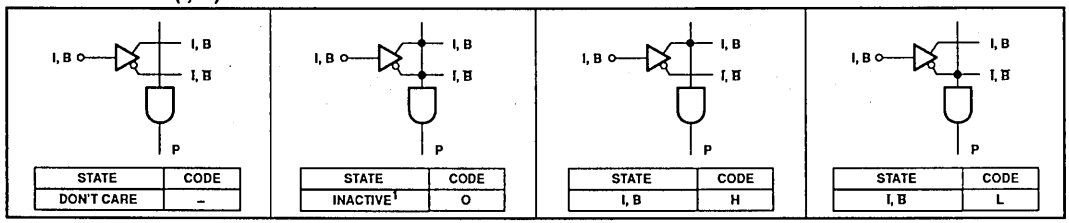
OUTPUT POLARITY – (O, B)



ACTIVE LEVEL	CODE
INVERTING[1]	L

ACTIVE LEVEL	CODE
NON-INVERTING	H

"AND" ARRAY – (I, B)



STATE	CODE
DON'T CARE	–

STATE	CODE
INACTIVE[1]	O

STATE	CODE
I, B	H

STATE	CODE
Ī, B̄	L

NOTE:
1. A factory shipped unprogrammed device is configured such that all cells are in a conductive state.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC18V8Z Series devices are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lighting could erase a typical PLC18V8Z in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC18V8Z is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the PLC18V8Z is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000µW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

PROGRAMMING

The PLC18V8Z35/I is programmable on conventional programmers for 20-pin PAL devices. Refer to the following charts for qualified manufacturers of programmers and software tools:

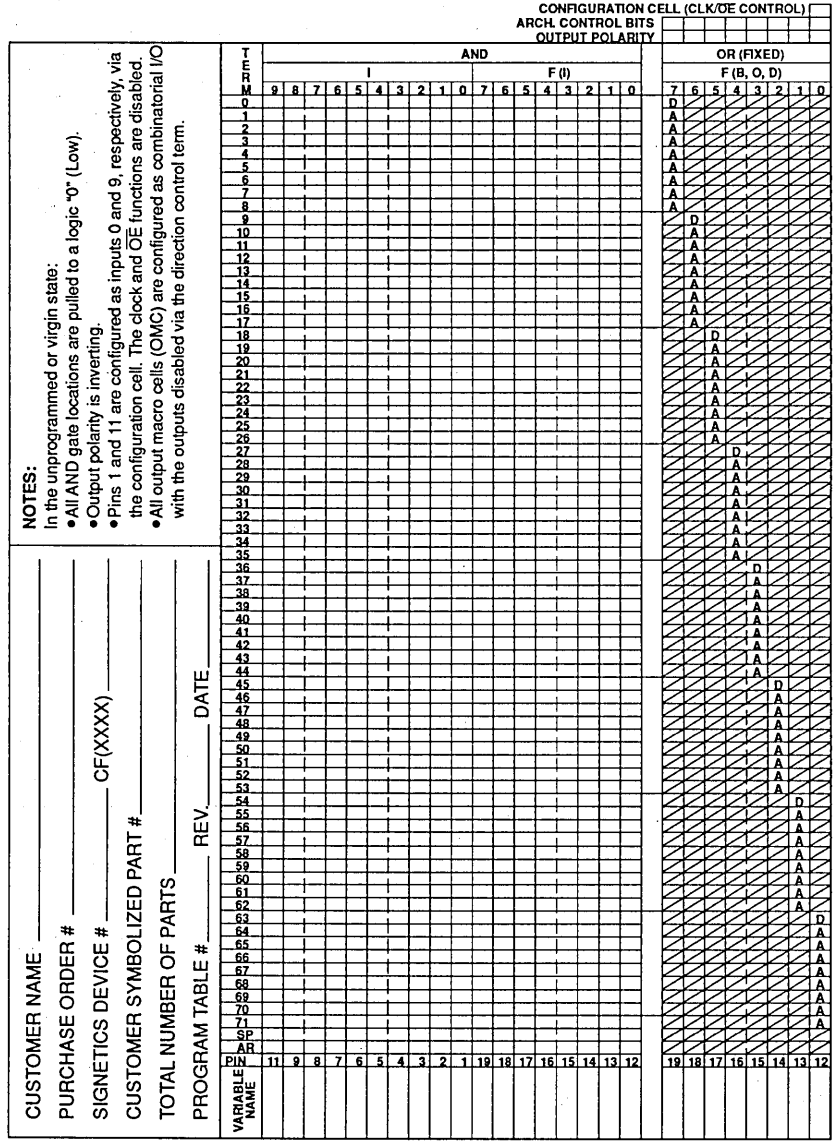
PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	System 29B, LogicPak™ 303A–011A; V09 (DIL) 303A–011B; V04 (PLCC) UNISITE 40/48 V2.6 (DIL) Chipsite (PLCC) – V2.8 MODEL 60 360A001 (DIL) 360A006 (PLCC)	86/4F
STAG MICROSYSTEMS, INC. 1600 WYATT DRIVE SUITE 3 SANTA CLARA, CALIFORNIA 95054 (408)988–1118	ZL30/30A PROGRAMMER REV. 30A34 (DIL) 30A001 Adaptor (PLCC) PPZ PROGRAMMER TBA	12/205

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
SIGNETICS COMPANY 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SNAP REV. 1.6 AND LATER SLICE REV. 1.0 AND LATER AMAZE SOFTWARE REV. 1.8 AND LATER
DATA I/O 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	ABEL™ SOFTWARE
LOGICAL DEVICES, INC. 1201 NORTHWEST 65TH PLACE FORT LAUDERDALE, FLORIDA 33309 (800)331–7766	CUPL™ SOFTWARE

Zero standby power universal PAL-type devices

PLC18V8Z Series

PROGRAM TABLE



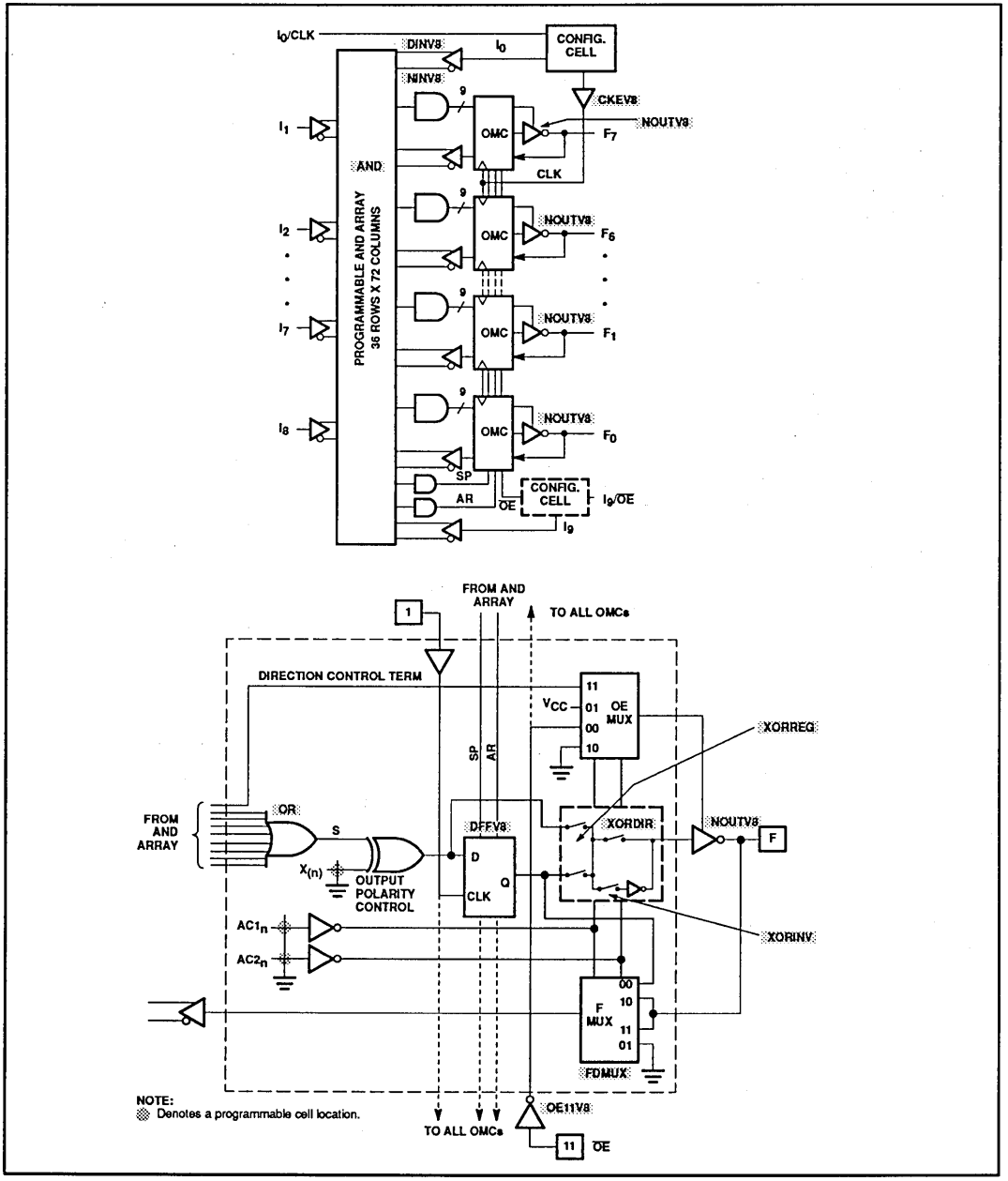
AND ARRAY	
INACTIVE	O
I, F (I, B)	H
I, F (I, B)	L
**DON'T CARE	–

CONTROL			
OMC ARCH.		OUTPUT POLARITY	
REGISTERED (D-TYPE)	D	NON-INVERTING	H
FIXED INPUT	I	INVERTING	L
FIXED OUTPUT	O	CONFIG. CELL*	
BIDIRECTIONAL I/O	B	PIN 1 = CLK; PIN 11 = OE	L
		PIN 1, PIN 11 = INPUT	H

OR ARRAY (FIXED)	
DATA CANNOT BE ENTERED INTO THE OR ARRAY FIELD DUE TO THE FIXED NATURE OF THE DEVICE ARCHITECTURE.	
DIRECTION CONTROL	D
ACTIVE OUTPUT	A
NOT USED	

* THE CONFIGURATION CELL IS AUTOMATICALLY PROGRAMMED BASED ON THE OMC ARCHITECTURE.
** FOR SP, AR: "–" IS NOT ALLOWED.

SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–1423
ECN No.	99373
Date of Issue	April 17, 1990
Status	Product Specification
Programmable Logic Devices	

10H20EV8 / 10020EV8

ECL programmable array logic

DESCRIPTION

The 10H20EV8/10020EV8 is an ultra high-speed universal ECL PAL®-type device. Combining versatile output macrocells with a standard AND/OR single programmable array, this device is ideal in implementing a user's custom logic. The use of Signetics state-of-the-art bipolar oxide isolation process enables the 10H20EV8/10020EV8 to achieve optimum speed in any design. The AMAZE design software package from Signetics simplifies design entry based upon Boolean or state equations.

The 10H20EV8/10020EV8 is a two-level logic element comprised of 11 fixed inputs, an input pin that can either be used as a clock or 12th input, 90 AND gates, and 8 Output Logic Macrocells. Each Output Macrocell can be individually configured as a dedicated input, dedicated output with polarity control, a bidirectional I/O, or as a registered output that has both output polarity control and feedback to the AND array. This gives the part the capability of having up to 20 inputs and eight outputs.

The 10H20EV8/10020EV8 has a variable number of product terms that can be OR'd per output. Four of the outputs have 12 AND terms available and the other four have 8 terms per output. This allows the designer the extra flexibility to implement those functions that he couldn't in a standard PAL device. Asynchronous Preset and Reset product terms are also included for system design ease. Each output has a separate output enable product term. Another feature added for the system designer is a power-up Reset on all registered outputs.

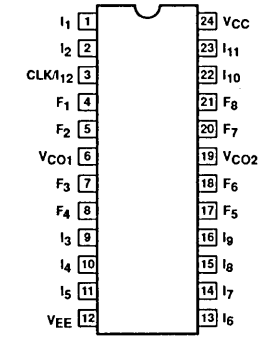
The 10H20EV8/10020EV8 also features the ability to Preload the registers to any desired state during testing. The Preload is not affected by the pattern within the device, so can be performed at any step in the testing sequence. This permits full logical verification even after the device has been patterned.

FEATURES

- Ultra high speed ECL device
  - t_{PD} = 4.5ns (max)
  - t_{IS} = 2.7ns (max)
  - t_{CKO} = 2.2ns (max)
  - f_{MAX} = 208MHz
- Universal ECL Programmable Array Logic
  - 8 user programmable output macrocells
  - Up to 20 inputs and 8 outputs
  - Individual user programmable output polarity
- Variable product term distribution allows increased design capability
- Asynchronous Preset and Reset capability
- 10KH and 100K options
- Power-up Reset and Preload function to enhance state machine design and testing
- Design support provided via AMAZE and other CAD tools
- Security fuse for preventing design duplication
- Available in 24-Pin 300mil-wide DIP and 28-Pin PLCC.

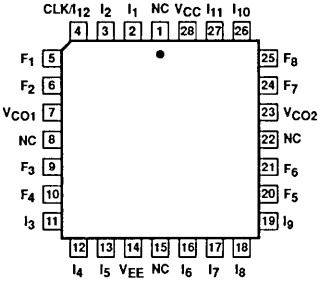
PIN CONFIGURATIONS

F Package



F = Ceramic DIP (300mil-wide)

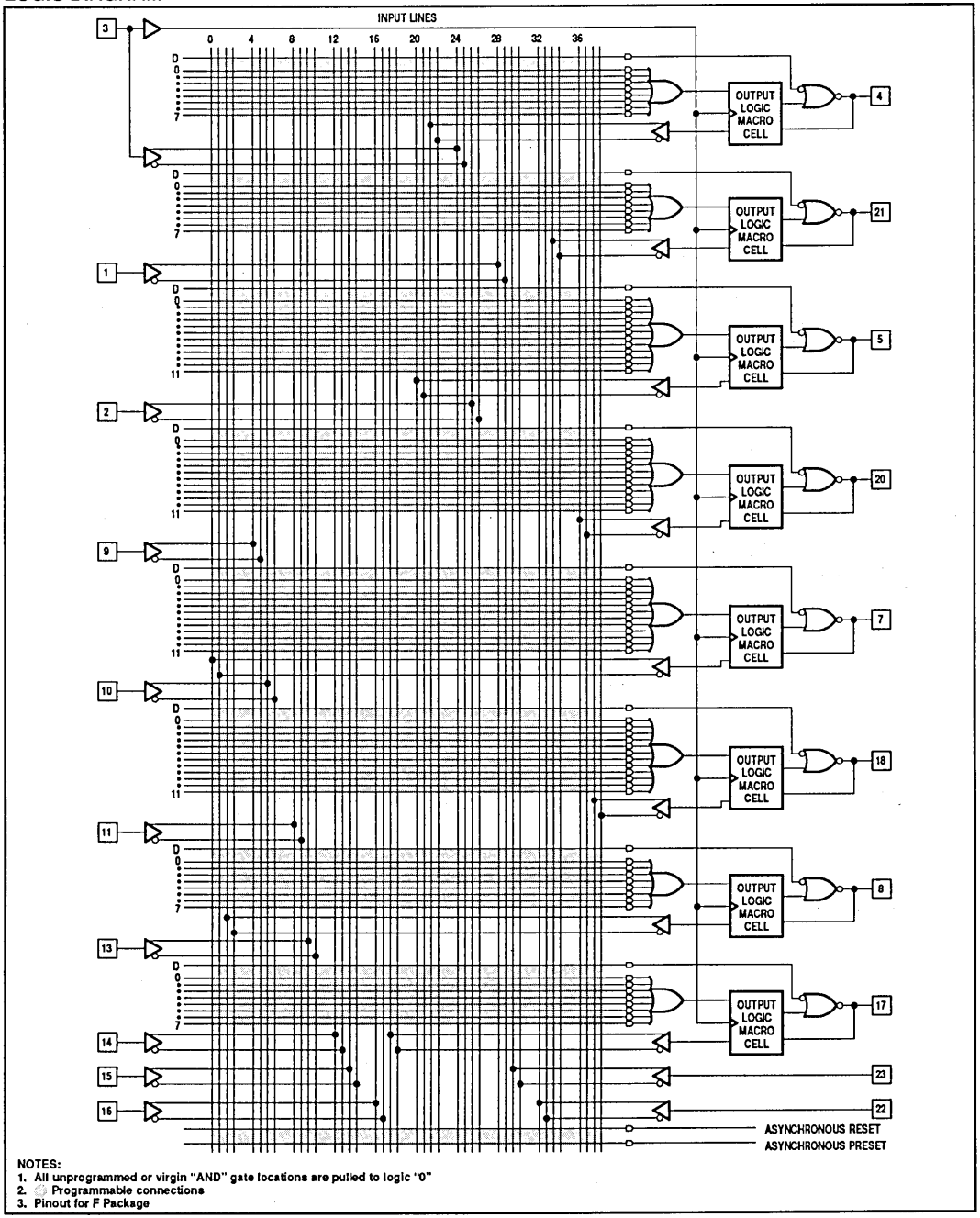
A Package



A = Plastic Leaded Chip Carrier

®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

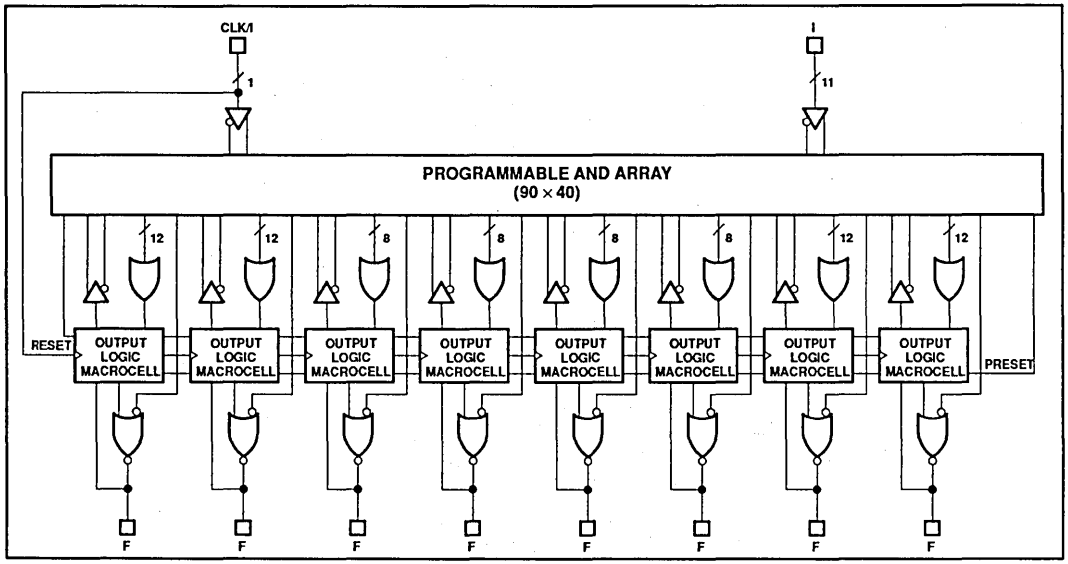
LOGIC DIAGRAM



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. Programmable connections
3. Pinout for F Package

ECL programmable array logic 10H20EV8 / 10020EV8

FUNCTIONAL DIAGRAM



FUNCTIONAL DESCRIPTION

The 10H20EV8/10020EV8 is an ultra high-speed universal ECL PAL-type device. Combining versatile Output Macrocells with a standard AND/OR single programmable array, this device is ideal in implementing a user's custom logic.

As can be seen in the Logic Diagram, the device is a two-level logic element with a programmable AND array. The 20EV8 can have up to 20 inputs and 8 outputs. Each output has a versatile Macrocell whereby the output can either be configured as a dedicated input, a dedicated combinatorial output with polarity control, a bidirectional I/O, or as a registered output that has both output polarity control and feedback into the AND array.

The device also features 90 product terms. Two of the product terms can be used for a global asynchronous preset and/or reset. Eight of the product terms can be used for individual output enable control of each Macrocell. The other 80 product terms are distributed among the outputs. Four of the outputs have eight product terms, while the other four have 12. This arrangement allows the utmost in flexibility when implementing user patterns.

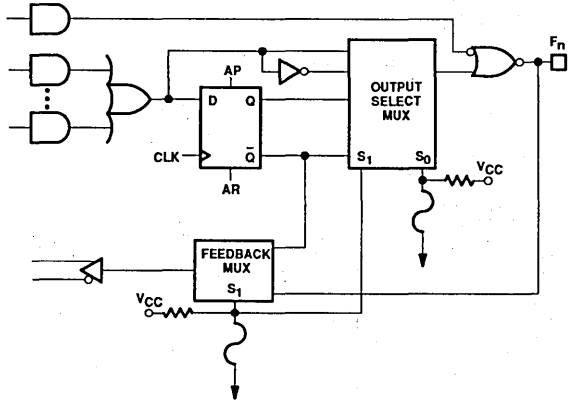


Figure 1. Output Logic Macrocell

Output Logic Macrocell

The 10H20EV8/10020EV8 incorporates an extremely versatile Output Logic Macrocell that allows the user complete flexibility when configuring outputs.

As seen in Figure 1, the 10H20EV8/ 10020EV8 Output Logic Macrocell consists of an edge-triggered D-type flip-flop, an output select MUX, and a feedback select MUX. Fuses S_0 and S_1 allow the user to select between the various cells. S_1 controls whether the output will be either registered with internal feedback or combinatorial I/O. S_0 controls the polarity of the output (Active-HIGH or Active-LOW). This allows the user to achieve the following configurations: Registered Active-HIGH output, Registered Active-LOW output, Combinatorial Active-HIGH output, and Combinatorial Active-LOW output. With the output enable product term, this list can be extended by adding the configurations of a Combinatorial I/O with Polarity or another input.

ECL programmable array logic 10H20EV8 / 10020EV8

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Ceramic Dual In-Line (300mil-wide)	10H20EV8–6F 10H20EV8–4F 10020EV8–6F 10020EV8–4F
28-Pin Plastic Leaded Chip Carrier	10H20EV8–6A 10H20EV8–4A 10020EV8–6A 10020EV8–4A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
V_{EE}	Supply voltage (V_{CC} = 0)	–8	V_{DC}
V_{IN}	Input voltage (V_{CC} = 0)	0 to V_{EE}	V_{DC}
I_O	Output source current	40	mA_{DC}
T_{amb}	Operating Temperature range	0 to +75 (10KH) 0 to +85 (100K)	°C
T_{stg}	Storage Temperature range	–55 to +150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

DEVICE	SYMBOL	PARAMETER	RATINGS		UNIT
			MIN	MAX	
10H20EV8	V_{EE}	Supply voltage	–5.46	–4.94	V_{DC}
	T_{amb}	Operating free-air temperature	0	+75	°C
10020EV8	V_{EE}	Supply voltage	–4.8	–4.2	V_{DC}
	T_{amb}	Operating free-air temperature	0	+85	°C

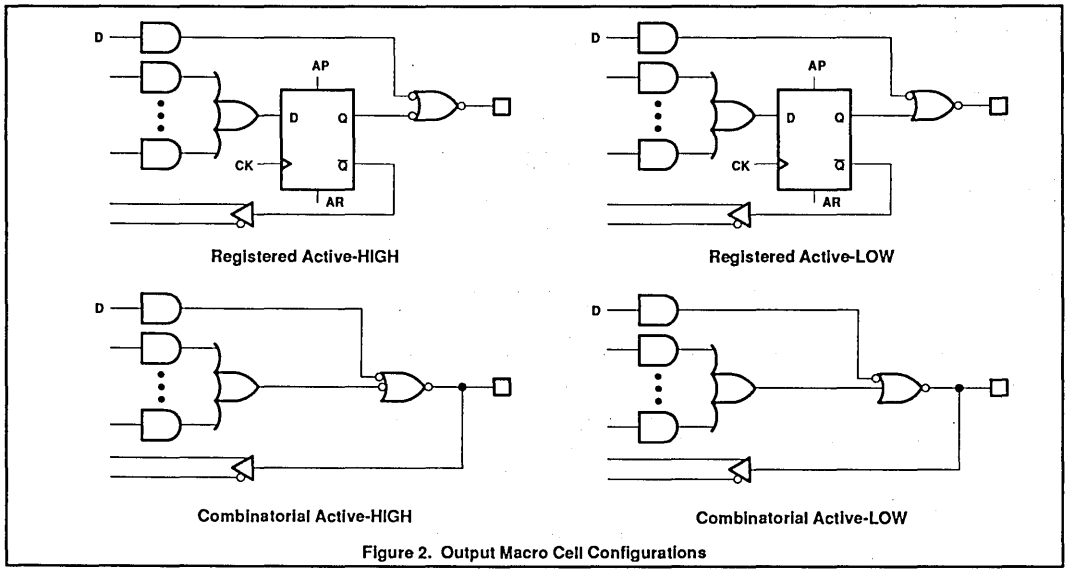


Figure 2. Output Macro Cell Configurations

OUTPUT MACRO CELL CONFIGURATION

Shown in Figure 2 are the four possible configurations of the output macrocell using fuses S_0 and S_1. As seen, the output can either be registered Active-HIGH/LOW with feedback or combinatorial Active-HIGH/LOW with feedback. If the registered mode is chosen, the feedback is from the $\overline{Q}$ output to the AND array enables one to make state machines or shift registers without having to tie the output to one of the inputs. If a combinatorial output is chosen, the feedback gate is enabled from the pin and allows one to create permanent outputs, permanent inputs, or I/O pins through the use of the output enable (D) product term.

OUTPUT ENABLE

Each output on the 10H20EV8/10020EV8 has its own individual product term for output enable. The use of the D product term (direction control) allows the user three possible configurations of the outputs. They are always enabled, always disabled, and controlled by a programmed pattern. A HIGH on the D term enables the output, while a LOW performs the disable function. Output enable control can be achieved by programming a pattern on the D term.

The output enable control can also be used to expand a designer's possibilities once a combinatorial output has been chosen. If the D term is always HIGH, the pin becomes a permanent Active-HIGH/LOW output. If the D term is always LOW (all fuses left intact), the pin now becomes an extra input.

PRESET AND RESET

The 10H20EV8/10020EV8 also includes a separate product term for asynchronous Preset and asynchronous Reset. These lines are common for all registers and are asserted when the specific product term goes HIGH. Being asynchronous, they are independent of the clock. It should be noted that the actual state of the output is dependent on how the polarity of the particular output has been chosen. If the outputs are a mix of Active-HIGH and Active-LOW, a Preset signal will force the Active-HIGH outputs HIGH while the Active-LOW outputs would go LOW, even though the Q output of all flip-flops would go HIGH. A Reset signal would force the opposite conditions.

PRELOAD

To simplify testing, the 10H20EV8/10020EV8 has also included PRELOAD circuitry. This allows a user to load any particular data desired into the registers regardless of the programmed pattern. This means that the PRELOAD can be done on a blank part and after that same part has been programmed to facilitate any post-fuse testing desired.

It can also be used by a designer to help debug his/her circuit. This could be important if a state machine was implemented in the 10H20EV8/ 10020EV8. The PRELOAD would allow a designer to enter any state in the sequence desired and start clocking from that particular point. Any or all transitions could be verified.

AMAZE

The AMAZE PLD Design Software development system also supports the 10H20EV8/10020EV8. AMAZE provides the following capabilities for the 10H20EV8/ 10020EV8:

- State equation entry
- Boolean equation entry
- Logic and timing simulation
- Automatic test vector generation

AMAZE operates on an IBM PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.0 or higher. The minimum system configuration for AMAZE is 640K bytes of RAM and a hard disk.

AMAZE compiles the design after completion for syntax and completeness. Programming data is generated in JEDEC format.

DESIGN SECURITY

The 10H20EV8/10020EV8 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

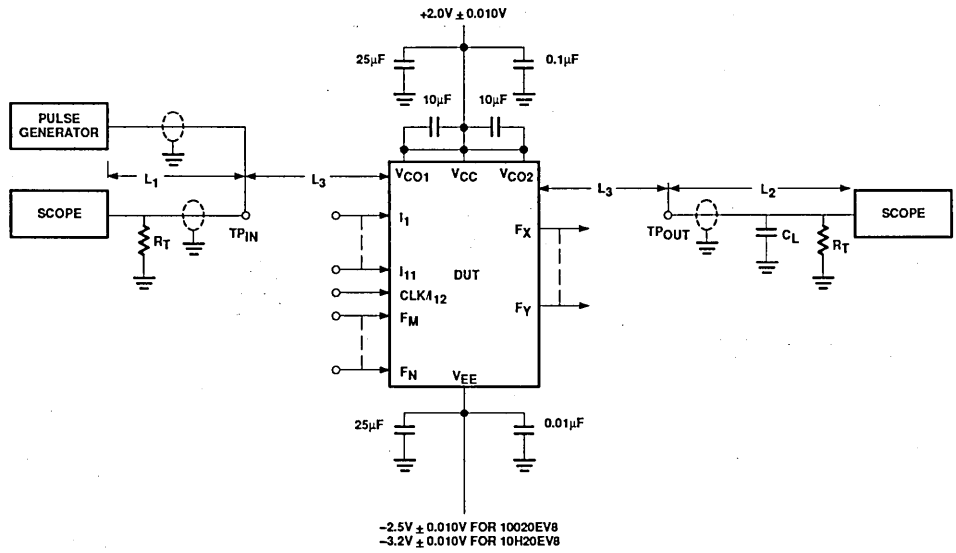
DC ELECTRICAL CHARACTERISTICS

10H20EV8: $0°C \leq T_{amb} \leq +75°C$, $V_{EE} = -5.2V \pm 5\%$, $V_{CC} = V_{CO1} = V_{CO2} = GND$
10020EV8: $0°C \leq T_{amb} \leq +85°C$, $-4.8V \leq V_{EE} \leq -4.2V$, $V_{CC} = V_{CO1} = V_{CO2} = GND$

SYMBOL	PARAMETER[1]	TEST CONDITIONS[2]		T_{amb}	LIMITS MIN	LIMITS MAX	UNITS
V_{OH}	High level output voltage	$V_{IN} = V_{IH}$ MAX or V_{IL} MIN	10KH	0°C +25°C +75°C	–1020 –980 –920	–840 –810 –735	mV
			100K	0°C to 85°C	–1025	–880	mV
V_{OHT}	High level output threshold voltage	$V_{IN} = V_{IH}$ MAX or V_{IL} MIN	100K	0°C to 85°C	–1035		mV
V_{OL}	Low level output voltage	$V_{IN} = V_{IH}$ MAX or V_{IL} MIN	10KH	0°C +25°C +75°C	–1950 –1950 –1950	–1630 –1630 –1600	mV
			100K	0°C to 85°C	–1810	–1620	mV
V_{OLT}	Low level output threshold voltage	$V_{IN} = V_{IH}$ MAX or V_{IL} MIN	100K	0°C to 85°C		–1610	mV
V_{IH}	High level input voltage	Guaranteed input voltage high for all inputs	10KH	0°C +25°C +75°C	–1170 –1130 –1070	–840 –810 –735	mV
			100K	0°C to 85°C	–1165	–880	mV
V_{IL}	Low level input voltage	Guaranteed input voltage low for all inputs	10KH	0°C +25°C +75°C	–1950 –1950 –1980	–1480 –1480 –1450	mV
			100K	0°C to 85°C	–1810	–1475	mV
I_{IH}	High level input current	$V_{IN} = V_{IH}$ MAX	10KH	0°C +75°C		220	µA
			100K	0°C to 85°C			
I_{IL}	Low level input current	$V_{IN} = V_{IL}$ MIN Except I/O Pins	10KH	0°C +75°C	0.3		µA
			100K	0°C to 85°C	0.5		
I_{EE}	Supply current	V_{EE} = MAX All inputs and outputs open	10KH	0°C to 75°C	–230		mA
			100K	0°C to 85°C			

NOTES:
1. All voltage measurements are referenced to the ground terminal.
2. Each ECL 10KH/100K series device has been designed to meet the DC specification after thermal equilibrium has been established. The circuit is in a test socket or mounted on a printed circuit board and transverse air flow greater than 500 (150 meters) linear fpm is maintained. Voltage levels will shift approximately 4mV with an air flow of 200 linear fpm. Outputs are terminated through a 50Ω resistor to –2V.
3. Terminals not specifically referenced can be left electrically open. Open inputs assume a logic LOW state. Any unused pins can be terminated to –2V. If tied to V_{EE}, it must be through a resistor > 10K.

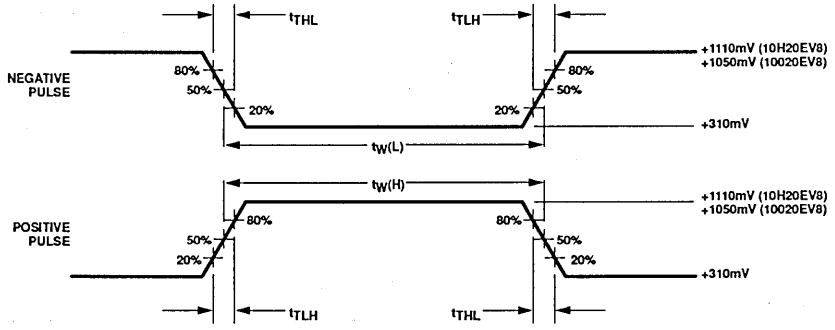
AC TEST CIRCUIT



NOTES:

1. Use decoupling capacitors of 0.1μF and 25μF from GND to V_{CC}, and 0.01μF and 25μF from GND to V_{EE} (0.01 and 0.1μF capacitors should be NPO Ceramic or MLC type). Decoupling capacitors should be placed as close as physically possible to the DUT and lead length should be kept to less than $^1/_4$ inch (6mm).
2. All unused inputs should be connected to either HIGH or LOW state consistent with the LOGIC function required.
3. All unused outputs are loaded with 50Ω to GND.
4. L_1 and L_2 are equal length 50Ω impedance lines. L_3, the distance from the DUT pin to the junction of the cable from the Pulse Generator and the cable to the Scope, should not exceed $^1/_4$ inch (6mm).
5. R_T = 50Ω terminator internal to Scope.
6. The unmatched wire stub between coaxial cable and pins under test must be less than $^1/_4$ inch (6mm) long for proper test.
7. C_L = Fixture and stray capacitance ≤ 3pF.
8. Any unterminated stubs connected anywhere along the transmission line between the Pulse Generator and the DUT or between the DUT and the Scope should not exceed $^1/_4$ inch (6mm) in length (refer to section on AC setup procedure).
9. All 50Ω resistors should have tolerance of ± 1% or better.
10. Test procedures are shown for only one input or set of input conditions. Other inputs are tested in the same manner.
11. Two 10μF capacitors between V_{CC} and V_{CO1} and V_{CO2} respectively located as close to the device as possible is recommended to reduce ringing.
12. Normal practice in test fixtures layout should be followed. Lead lengths, particular to the power supply, should be as short as possible.

VOLTAGE WAVEFORMS



INPUT PULSE REQUIREMENTS $V_{CC} = V_{CO1} = V_{CO2} = +2.0V \pm 0.010V$, $V_{EE} = -3.2V \pm 0.010V$, V_T = GND (0V)					
FAMILY	AMPLITUDE	REP RATE	PULSE WIDTH	t_{TLH}	t_{THL}
10KH ECL	800mV$_{P-P}$	1MHz	500ns	1.3 ± 0.2ns	1.3 ± 0.2ns
INPUT PULSE REQUIREMENTS $V_{CC} = V_{CO1} = V_{CO2} = +2.0V \pm 0.010V$, $V_{EE} = -2.5V \pm 0.010V$, V_T = GND (0V)					
FAMILY	AMPLITUDE	REP RATE	PULSE WIDTH	t_{TLH}	t_{THL}
100K ECL	740mV$_{P-P}$	1MHz	500ns	0.7 ± 0.1ns	0.7 ± 0.1ns

Input Pulse Definition

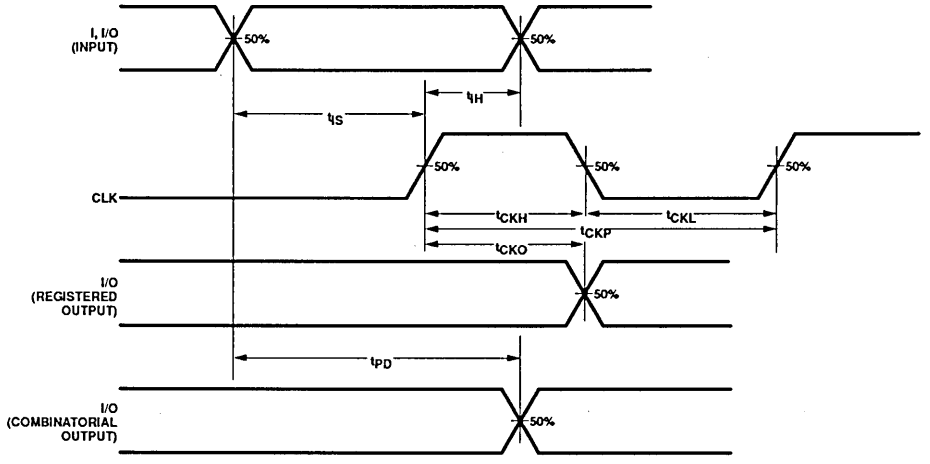
AC ELECTRICAL CHARACTERISTICS

10H20EV8: $0°C \leq T_{amb} \leq +75°C$, $V_{EE} = -5.2V \pm 5\%$, $V_{CC} = V_{CO1} = V_{CO2} = GND$
10020EV8: $0°C \leq T_{amb} \leq +85°C$, $-4.8V \leq V_{EE} \leq -4.2V$, $V_{CC} = V_{CO1} = V_{CO2} = GND$

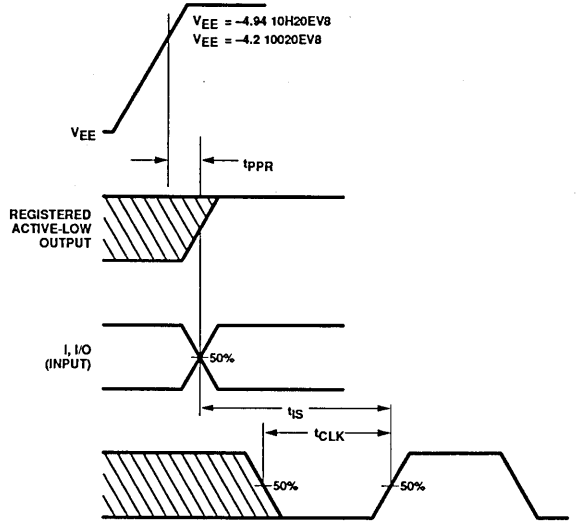
SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS[1]	LIMITS				UNIT
					–4		–6		
					MIN	MAX	MIN	MAX	
Pulse Width									
t_{CKH}	Clock High	CLK +	CLK –		2		3		ns
t_{CKL}	Clock Low	CLK –	CLK +		2		3		ns
t_{CKP}	Clock Period	CLK +	CLK +		4		6		ns
t_{PRH}	Preset/Reset Pulse	(I, I/O) ±	(I, I/O) ±		4.5		6		ns
Setup and Hold Time									
t_{IS}	Input	(I, I/O) ±	CLK +		2.7		4		ns
t_{IH}	Input	CLK +	(I, I/O) ±		0		0		ns
t_{PRS}	Clock Resume after Preset/Reset	(I, I/O) ±	CLK +		4.5		6		ns
Propagation Delay									
t_{PD}	Input	(I, I/O) ±	I/O ±			4.5		6	ns
t_{CKO}	Clock	CLK +	I/O ±			2.2		3	ns
t_{OE}	Output Enable	(I, I/O) ±	I/O			4		6	ns
t_{OD}	Output Disable	(I, I/O) ±	I/O			4		6	ns
t_{PRO}	Preset/Reset	(I, I/O) ±	I/O ±			4.5		6	ns
t_{PPR}	Power–on Reset	V_{EE}	I/O			10			ns

NOTES:
1. Refer to AC Test Circuit and Voltage Wafeforms diagrams.

TIMING DIAGRAMS

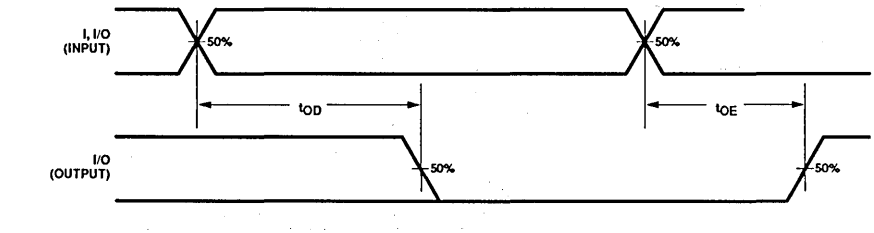


Flip-Flop and Gate Outputs

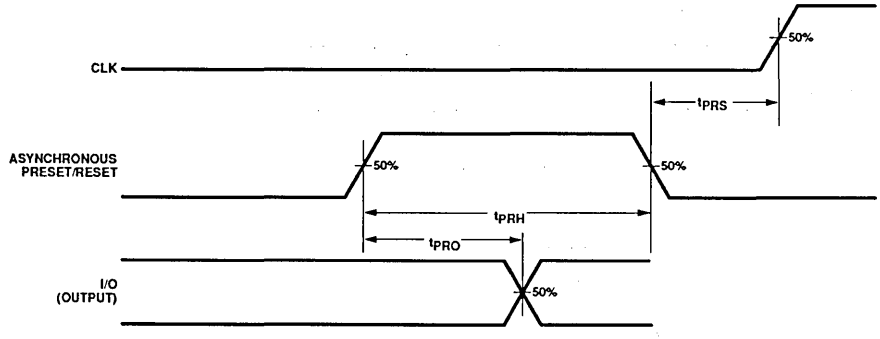


Power-On Reset

TIMING DIAGRAMS (Continued)



Output Enable/Disable

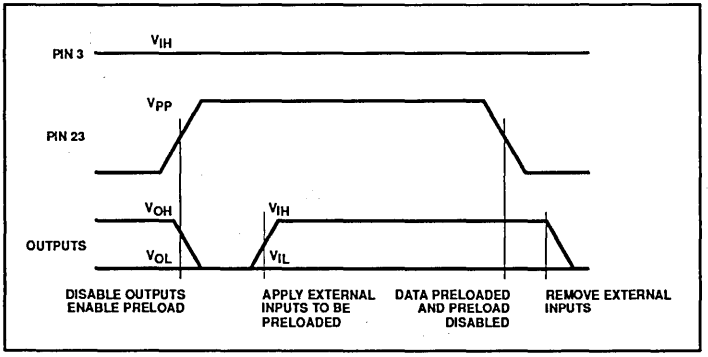


Asynchronous Preset/Reset

REGISTER PRELOAD

The 10H20EV8/10020EV8 has included circuitry that allows a user to load data into the output registers. Register PRELOAD can be done at any time and is not dependent on any particular pattern programmed into the device. This simplifies the ability to fully verify logic states and sequences even after the device has been patterned.

The pin levels and sequence necessary to perform the register PRELOAD are shown below.



SYMBOL	PARAMETER	LIMITS			UNIT
		MIN	TYP	MAX	
V_{IH}	Input HIGH level during PRELOAD and Verify	–1.1	–0.9	–0.7	V
V_{IL}	Input LOW level during PRELOAD and Verify	–1.85	–1.65	–1.45	V
V_{PP}	PRELOAD enable voltage applied to I_{11}	1.45	1.6	1.75	V

NOTE:

1. Unused inputs should be handled as follows:
   - Set at V_{IH} or V_{IL}
   - Terminated to –2V
   - Tied to V_{EE} through a resistor > 10K
   - Open

ECL programmable array logic | 10H20EV8 / 10020EV8

LOGIC PROGRAMMING

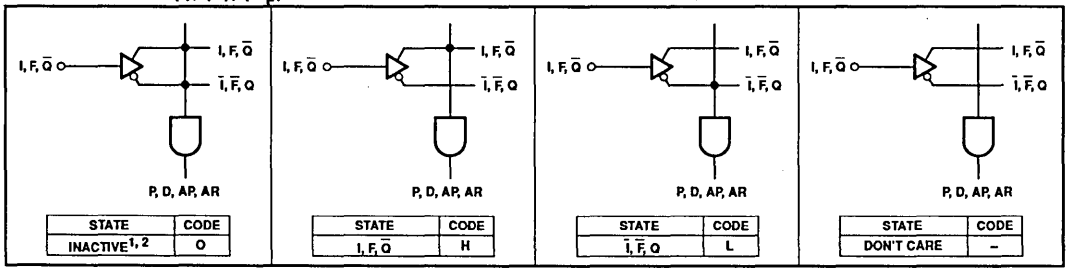
The 10H20EV8/10020EV8 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the 10H20EV8/10020EV8.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

10H20EV8/10020EV8 logic designs can also be generated using the program table entry format detailed on the following page. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE are available, free of charge, to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, F, Q, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

"AND" ARRAY – (I), (F), ($\overline{Q}_p$)



STATE	CODE
INACTIVE[1,2]	0

STATE	CODE
I, F, Q̄	H

STATE	CODE
Ī, F̄, Q	L

STATE	CODE
DON'T CARE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate (P, D, AP, AR) will be unconditionally inhibited if any one of the I, F or Q link pairs is left intact.

OUTPUT MACROCELL CONFIGURATIONS

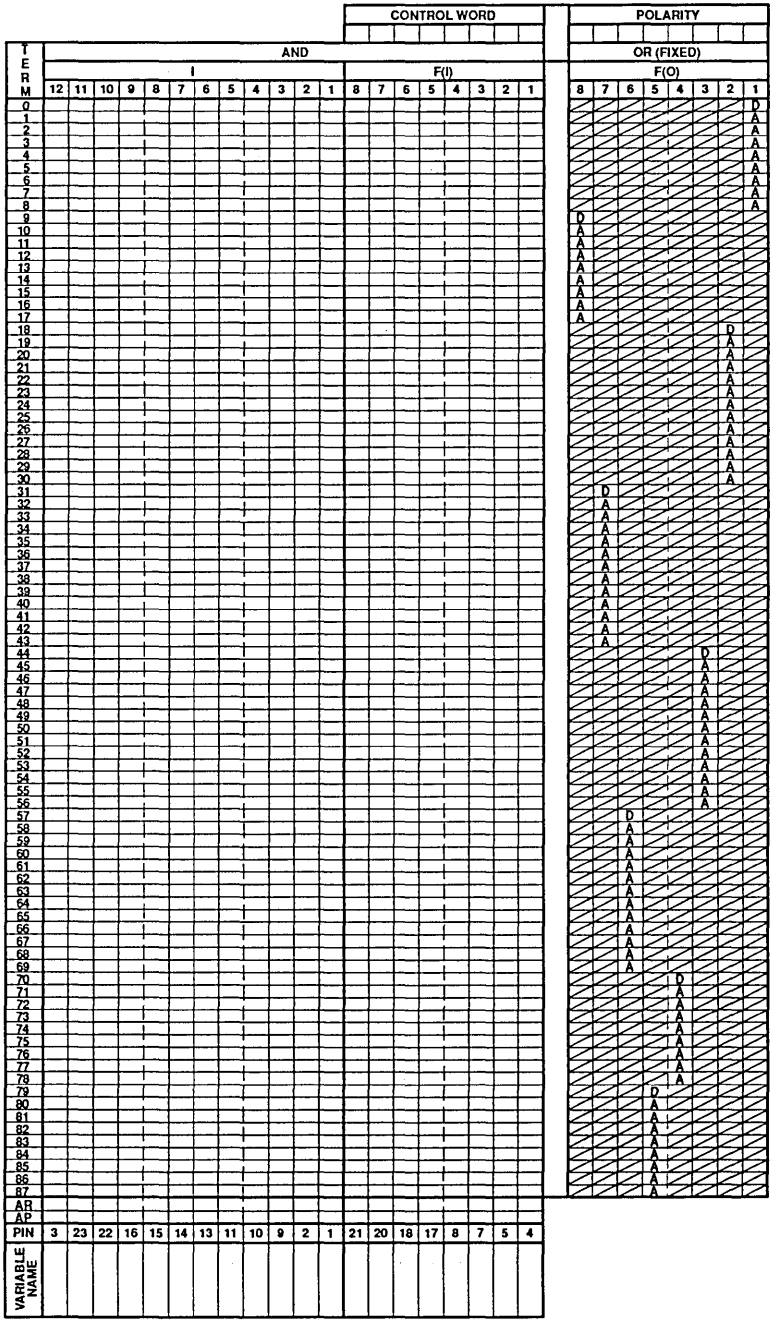
OUTPUT MACROCELL CONFIGURATION	CONTROL WORD FUSE	POLARITY FUSE
Registered Output, Active–HIGH	D	H
Registered Output, Active–LOW	D[1]	L[1]
Combinatorial I/O, Active–HIGH	B	H
Combinatorial I/O, Acitve–LOW	B	L

NOTES:
1. This is the initial (unprogrammed) state of the device.

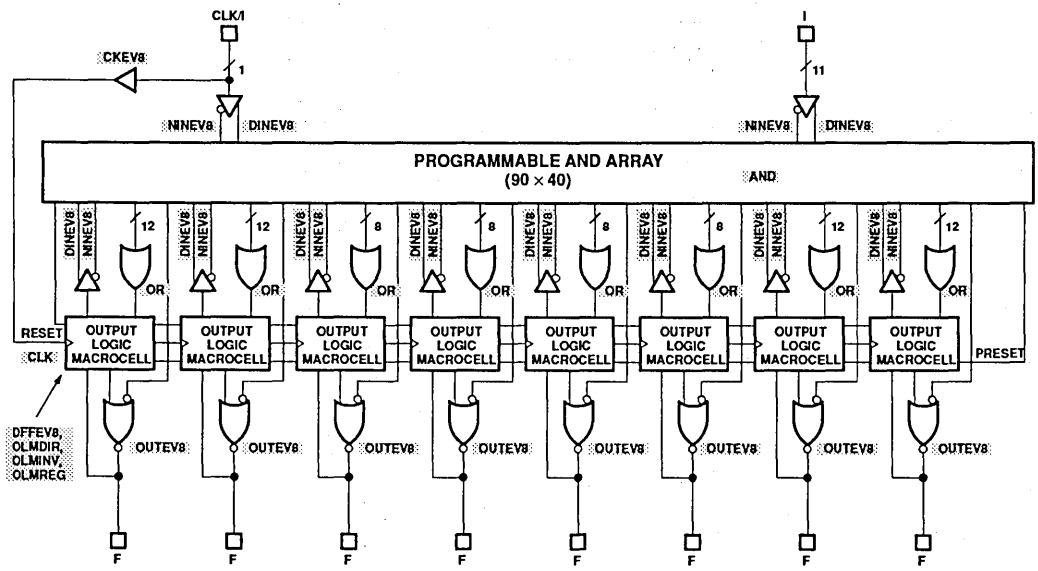
ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

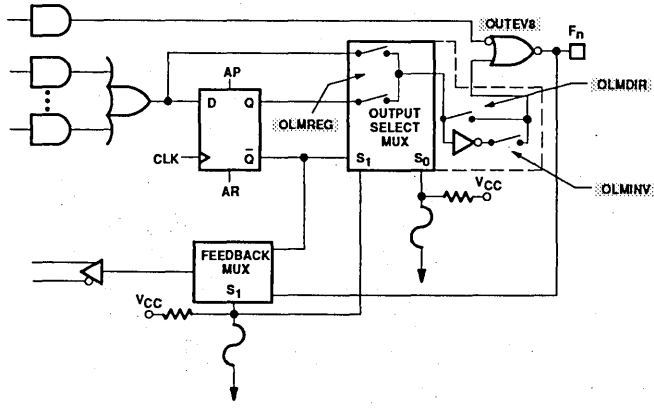
ECL programmable array logic 10H20EV8 / 10020EV8

PROGRAM TABLE



SNAP RESOURCE SUMMARY DESIGNATIONS





Output Logic Macrocell

Document No.	
ECN No.	
Date of Issue	June 1990
Status	Preliminary Specification
Programmable Logic Devices	

PLQ20R8–5 Series

PAL®-type devices

20L8, 20R8, 20R6, 20R4

FEATURES

- Ultra high-speed
  - t_{PD} = 5ns and f_{MAX} = 118MHz
- 100% functionally and pin-for-pin compatible with industry standard 24-pin PAL ICs
- Power-up reset function to enhance state machine design and testability
- Design support provided via AMAZE and other CAD tools for Series 24 PAL devices
- Field-programmable on industry standard programmers
- Security fuse
- Individual 3-State control of all outputs
- Register preload for testability
- Power-up 3-State
- 24-Pin DIP and 28-Pin PLCC

DESCRIPTION

The Signetics PLQ20XX family consists of ultra high-speed 5ns versions of Series 24 PAL devices.

The PLQ20XX family is 100% functional and pin-compatible with the 20L8, 20R8, 20R6, and 20R4 Series devices.

The sum of products (AND-OR) architecture is comprised of 64 AND gates and 8 fixed OR gates. Multiple bidirectional pins provide variable input/output pin ratios. Individual 3-State control of all outputs and registers with feedback (R8, R6, R4) is also provided. Proprietary designs can be protected by programming the security fuse.

The PLQ20R8, R6, and R4 have D-type flip-flops which are loaded on the Low-to-High transition of the clock input.

In order to facilitate state machine design and testing, a power-up reset function has been incorporated into these devices to reset all internal registers to active-Low after a specific period of time.

The Signetics State-of-the-Art BiCMOS process, known as QUBiC, has been employed to achieve higher levels of operating performance for the PLQ20XX family of PLDs. The QUBiC transistors have been optimized to provide two-thirds more speed at less than half the power consumed from products using our last generation of bipolar technology. QUBiC reduces on-chip delays and provides high output drive currents while consuming power at very low levels.

The PLQ20XX family of devices are field programmable, enabling the user to quickly generate custom patterns using standard programming equipment. See the programmer chart for qualified programmers.

The AMAZE software package from Signetics supports easy design entry for the PLQ20XX series as well as other PLD devices from Signetics. The PLQ20XX series are also supported by other standard CAD tools for PAL-type devices.

Order codes are listed in the Ordering Information table.

DEVICE NUMBER	DEDICATED INPUTS	COMBINATORIAL OUTPUTS	REGISTERED OUTPUTS
PLQ20L8	14	8 (6 I/O)	0
PLQ20R8	12	0	8
PLQ20R6	12	2 I/O	6
PLQ20R4	12	4 I/O	4

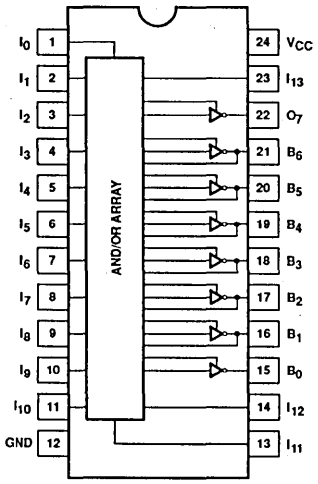
®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

PAL-type devices
20L8, 20R8, 20R6, 20R4

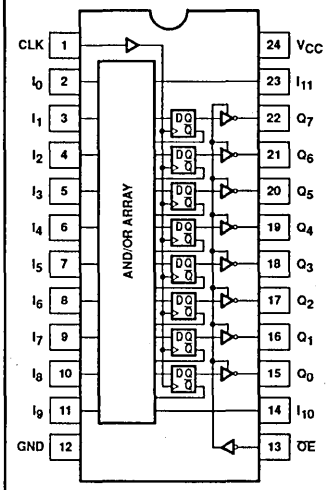
PLQ20R8–5 Series

PIN CONFIGURATIONS

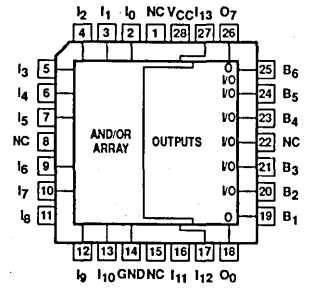
PLQ20L8



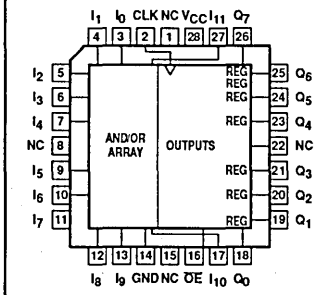
PLQ20R8



PLQ20L8



PLQ20R8



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock Input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

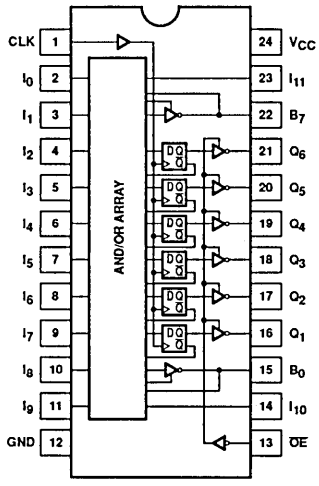
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock Input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

PAL-type devices
20L8, 20R8, 20R6, 20R4

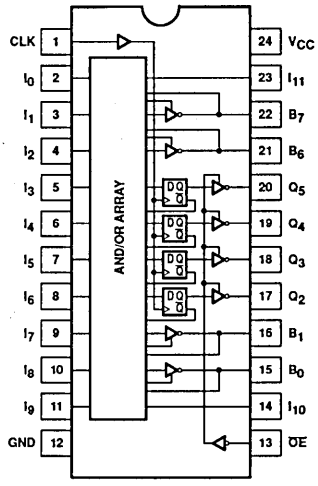
PLQ20R8–5 Series

PIN CONFIGURATIONS

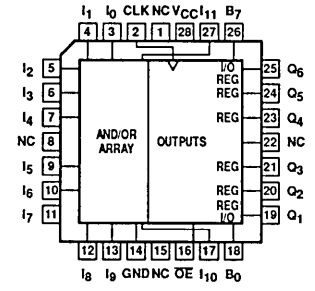
PLQ20R6



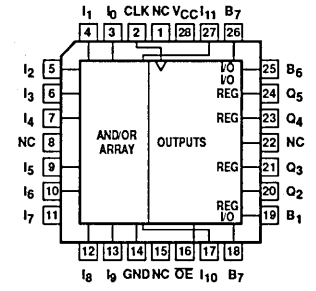
PLQ20R4



PLQ20R6



PLQ20R4



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

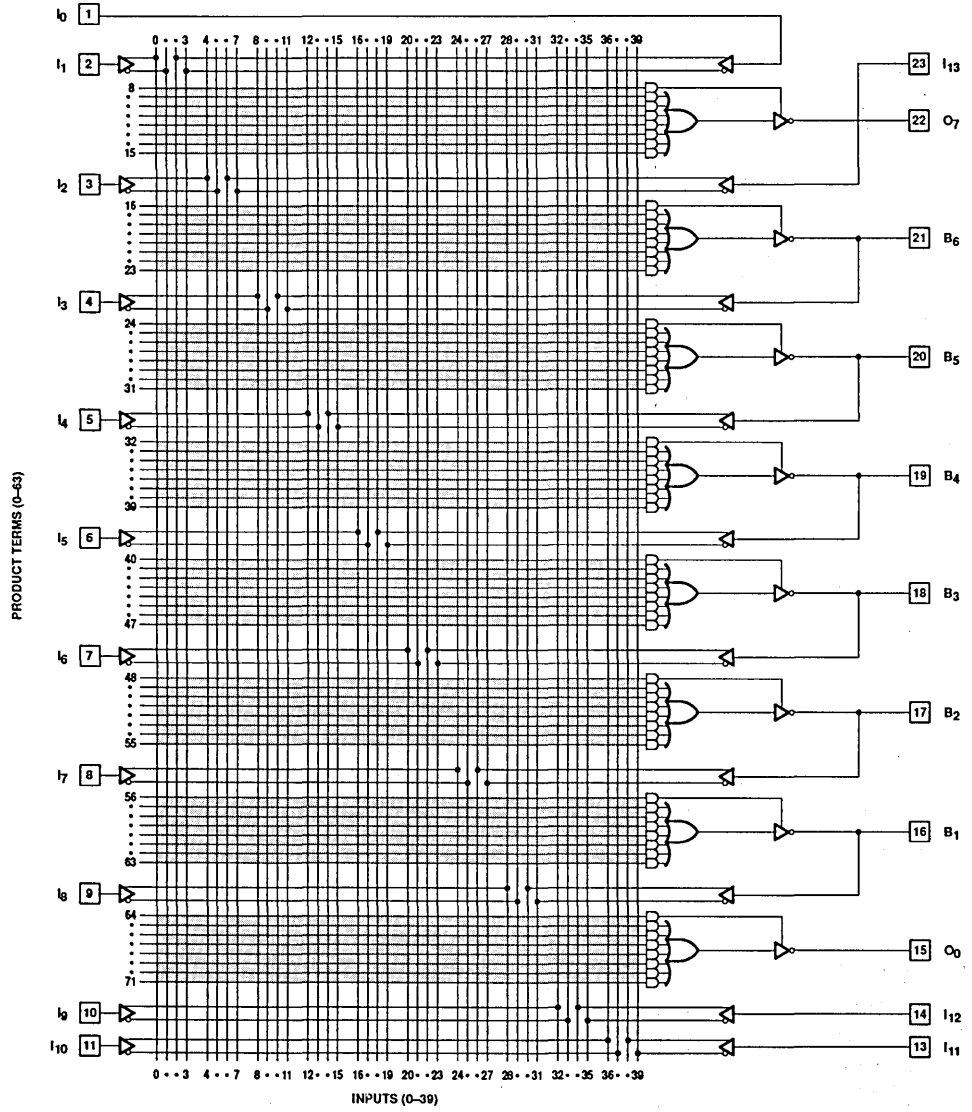
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

LOGIC DIAGRAM

PLQ20L8



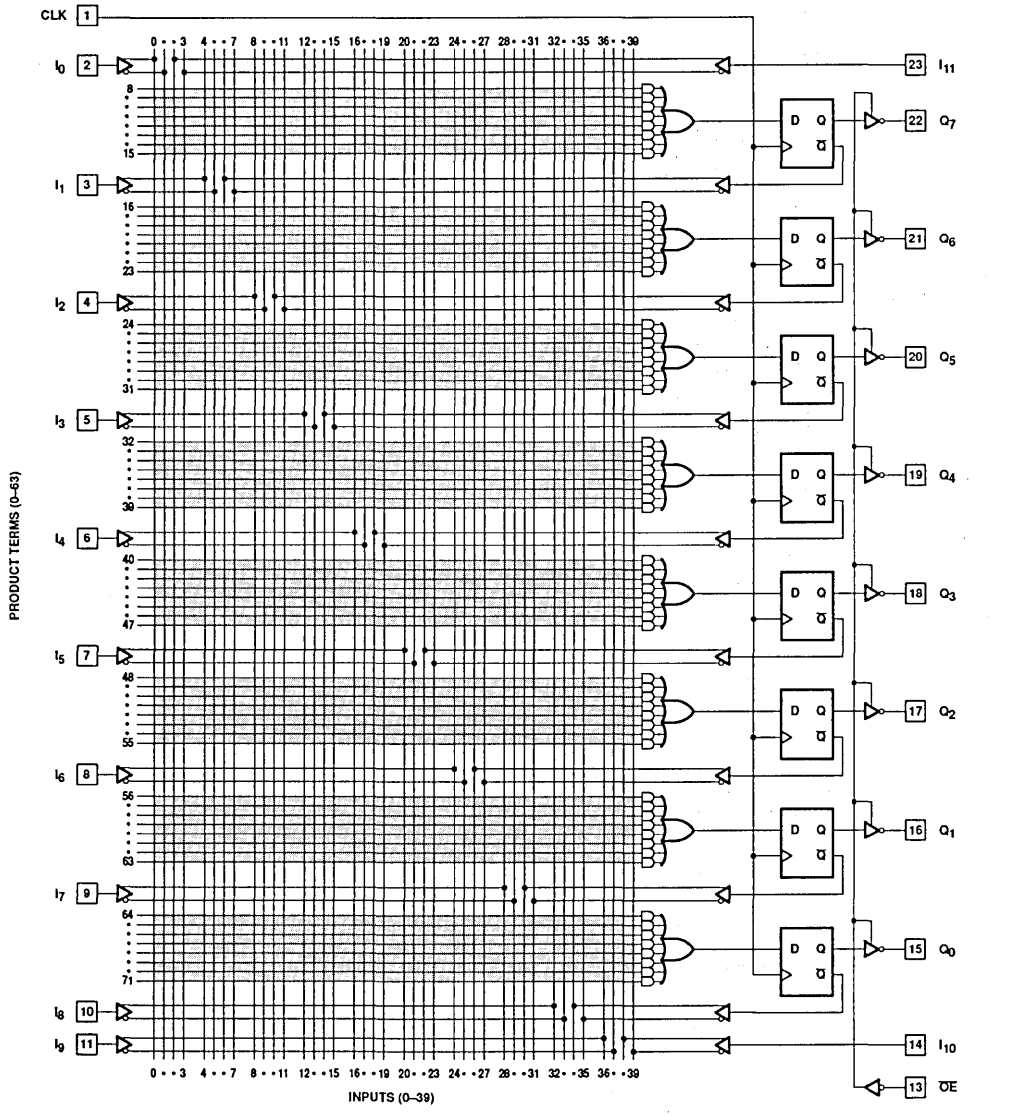
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ⊗ Programmable connections.

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

LOGIC DIAGRAM

PLQ20R8



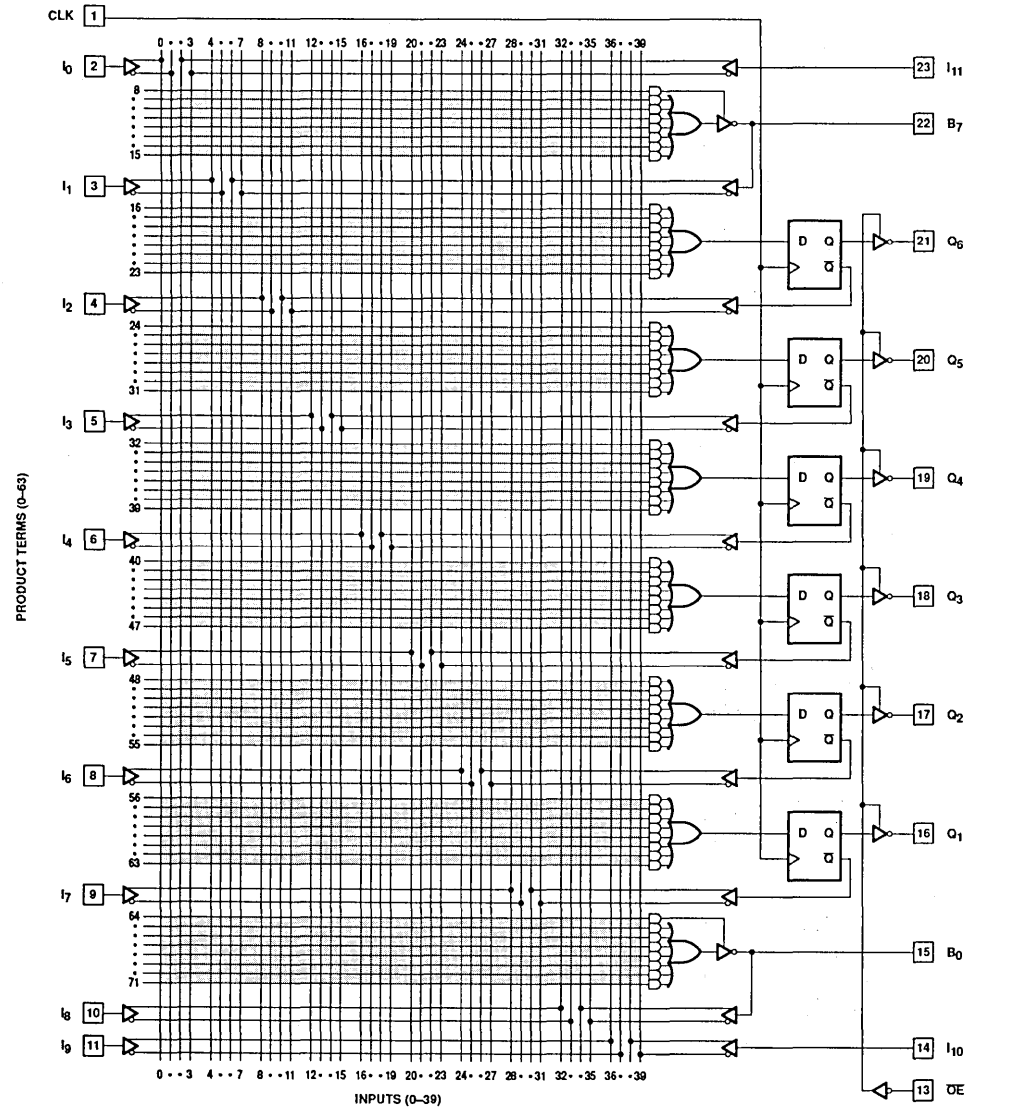
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

LOGIC DIAGRAM

PLQ20R6



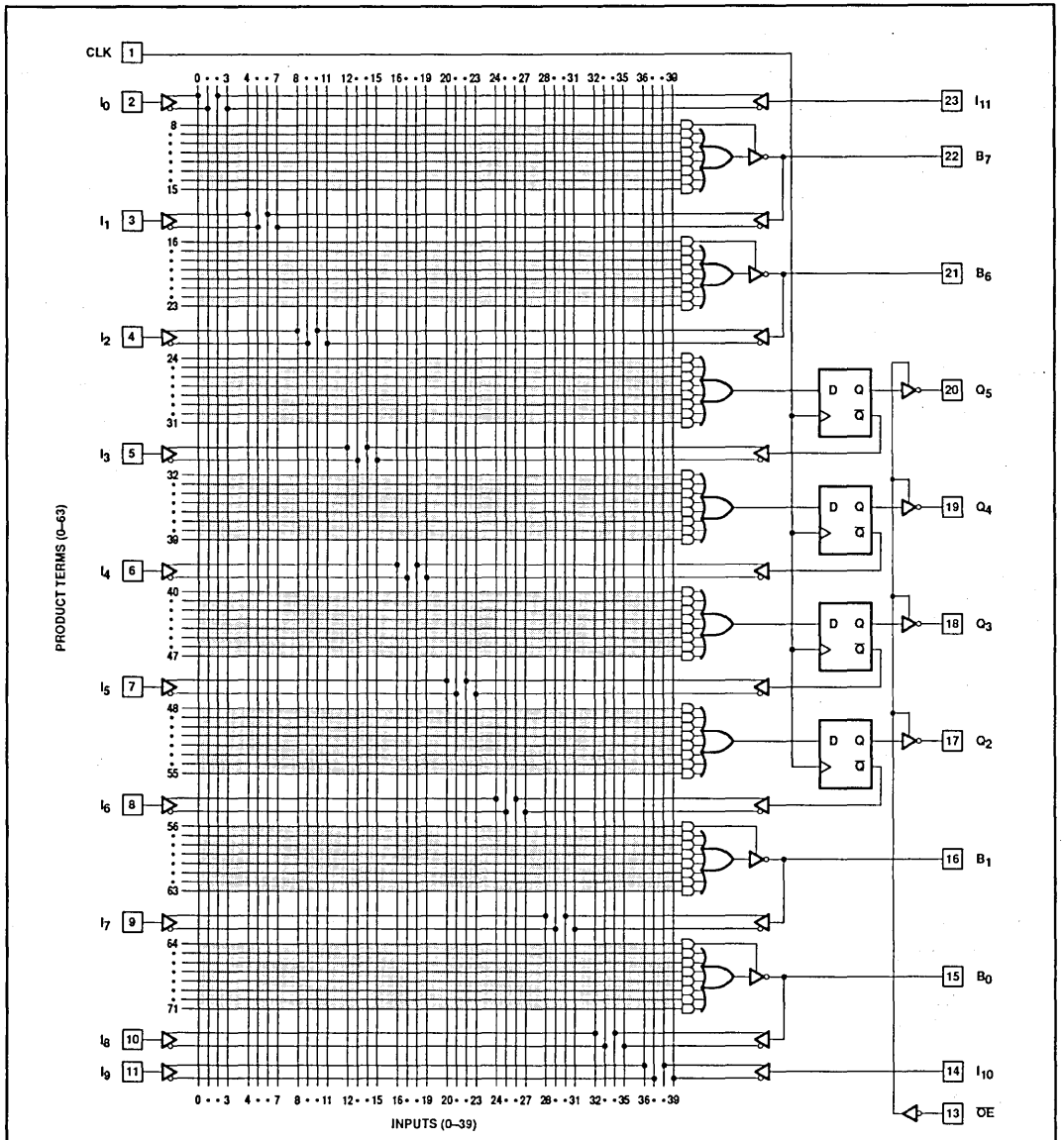
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ◌ Programmable connections.

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

LOGIC DIAGRAM

PLQ20R4



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ⊗ Programmable connections.

PAL-type devices 20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

FUNCTIONAL DESCRIPTIONS

The PLQ20XX series utilizes the familiar sum-of-products implementation consisting of a programmable AND array and a fixed OR array. These devices are capable of replacing an equivalent of four or more SSI/MSI integrated circuits to reduce package count and board area occupancy, consequently improving reliability and design cycle over Standard Cell or gate array options. By programming the security fuse, proprietary designs can be protected from duplication.

The PLQ20XX series consists of four PAL-type devices. Depending on the particular device type, there are a variable number of combinatorial and registered outputs available to the designer. The PLQ20L8 is a combinatorial part with 8 user configurable outputs (6 bidirectional), while the other three devices, PLQ20R8, PLQ20R6, PLQ20R4, have respectively 8, 6, and 4 output registers.

3-State Outputs

The PLQ20XX series devices also feature 3-State output buffers on each output pin which can be programmed for individual control of all outputs. The registered outputs (Qn) are controlled by an external input (/OE), and the combinatorial outputs (On, Bn) use a product term to control the enable function.

Programmable Bidirectional Pins

The PLQ20XX products feature variable Input/Output ratios. In addition to 12 dedicated inputs, each combinatorial output pin of the registered devices can be individually programmed as an input or output. The PLQ20L8 provides 14 dedicated inputs and 6 Bidirectional I/O lines that can be individually configured as inputs or outputs.

Output Registers

The PLQ20R8 has 8 output registers, the 20R6 has 6, and the 20R4 has 4. Each output register is a D-type flip-flop which is loaded on the Low-to-High transition of the clock input. These output registers are capable of feeding the outputs of the registers back into the array to facilitate design of synchronous state machines.

Power-up Reset

By resetting all flip-flops to a logic Low, as the power is turned on, the PLQ20R8, R6, R4 enhance state machine design and initialization capability.

Register Preload

Preload function allows the register to be loaded from the output pins. This feature allows functional testing of sequential patterns by loading output states.

Power-up 3-State

All outputs will be disabled when V_{CC} is 3.0V ± 20% (25°C). This special feature keeps outputs 3-Stated during power-up. Only when V_{CC} reaches its normal operating range will device function normally.

Software Support

Like other Programmable Logic Devices from Signetics, the PLQ20XX series are supported by AMAZE, the PC-based software development tool from Signetics. The PLQ20XX family of devices are also supported by standard CAD tools for PAL devices, including ABEL and CUPL.

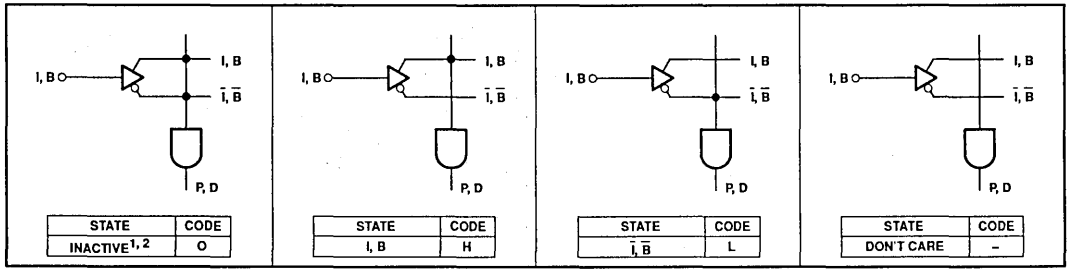
AMAZE is available free of charge to qualified users.

Logic Programming

The PLQ20XX series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLQ20XX architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

AND ARRAY – (I, B)



STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
$\overline{I, B}$	L

STATE	CODE
DON'T CARE	–

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All P_n terms are disabled.
2. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

PAL-type devices 20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic Dual-In-Line 300mil-wide	PLQ20R8–5N PLQ20R6–5N PLQ20R4–5N PLQ20L8–5N
28-Pin Plastic Leaded Chip Carrier (PLCC)	PLQ20R8–5A PLQ20R6–5A PLQ20R4–5A PLQ20L8–5A

NOTE:
The PLQ20XX series of devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Signetics Military Data Book.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7.0	V_{DC}
V_{IN}	Input voltage	–1.2	+7.0	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = 24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2 mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low[3]	V_{IN} = 0.40V	–250			µA
I_{IH}	High[3]	V_{IN} = 2.7V			25	µA
I_I	Maximum input current	V_{IN} = 5.5V, V_{CC} = MAX			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage	V_{OUT} = 2.7V			100	µA
I_{OZL}	Output leakage	V_{OUT} = 0.4V	–100			µA
I_{OS}	Short circuit[4, 5]	V_{OUT} = 0.5V	–30		–130	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		150	210	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5V				
		V_{OUT} = 2.0V		8		pF
C_B	I/O (B)	V_{OUT} = 2V, f = 1MHz		8		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Test one at a time.
5. Duration of short circuit should not exceed 1 second.
6. These parameters are not 100% tested but periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 200\Omega$, $R_2 = 390\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

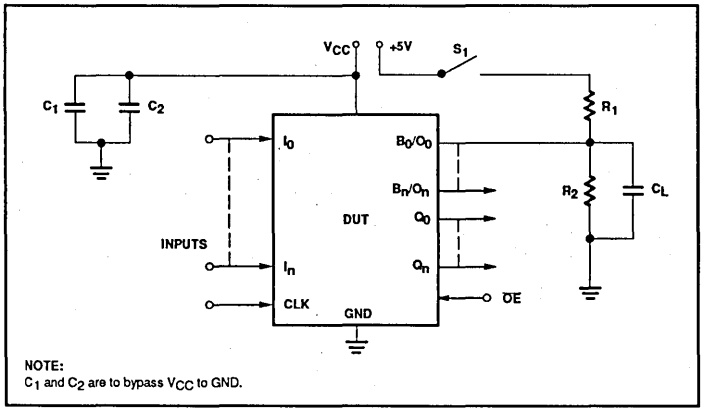
SYMBOL	PARAMETER	FROM	TO	LIMITS MIN[1]	LIMITS MAX	UNIT
Pulse Width						
t_{CKH}	Clock High	CLK+	CLK–	3.0		ns
t_{CKL}	Clock Low	CK–	CLK+	3.0		ns
t_{CKP}	Period	CLK+	CLK+	6.0		ns
Setup & Hold time						
t_{IS}	Input	Input or feedback	CLK+	4.0		ns
t_{IH}	Input	CLK+	Input or feedback	0		ns
Propagation delay						
t_{CKO}	Clock	CLK±	Q±		4.5	ns
t_{CKF}	Clock[3]	CLK±	$\overline{Q}$		2.5	ns
t_{PD}	Output (20L8, R6, R4)[2]	I, B	Output		5.0	ns
t_{OE1}	Output enable[4]	$\overline{OE}$	Output enable		6.0	ns
t_{OE2}	Output enable[4,5]	I	Output enable		8.0	ns
t_{OD1}	Output disable[4]	$\overline{OE}$	Output disable		6.0	ns
t_{OD2}	Output disable[4,5]	I	Output disable		8.0	ns
t_{SKW}	Output	Q	Q		1.0	ns
t_{PPR}	Power-Up Reset	V_{CC}+	Q+		8.0	ns
Frequency (20R8, R6, R4)						
f_{MAX}	No feedback $1/(t_{CKL} + t_{CKH})$[6]				167	MHz
	Internal feedback $1/(t_{IS} + t_{CKF})$[6]				154	MHz
	External feedback $1/(t_{IS} + t_{CKO})$[6]				118	MHz

* For definitions of the terms, please refer to the Timing/Frequency Definitions tables.

NOTES:

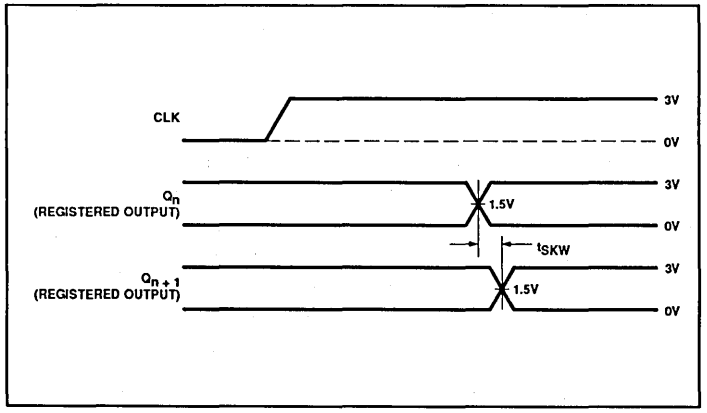
1. C_L = 0pF while measuring minimum output delays.
2. t_{PD} test conditions: C_L = 50pF (with jig and scope capacitance), V_{IH} = 3V, V_{IL} = 0V, $V_{OH} = V_{OL}$ = 1.5V.
3. t_{CKF} was calculated from measured Internal f_{MAX}.
4. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
5. Same function as t_{OE1} and t_{OD1}, with the difference of using product term control.
6. Not 100% tested, but calculated at initial characterization and at any time a modification in design takes place which may affect the frequency.

TEST LOAD CIRCUIT

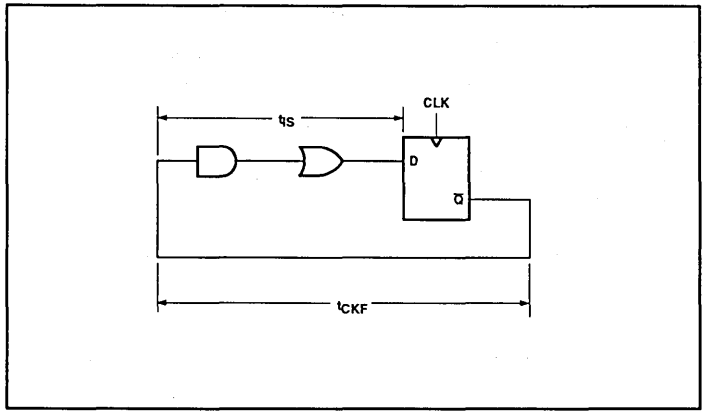


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

OUTPUT REGISTER SKEW



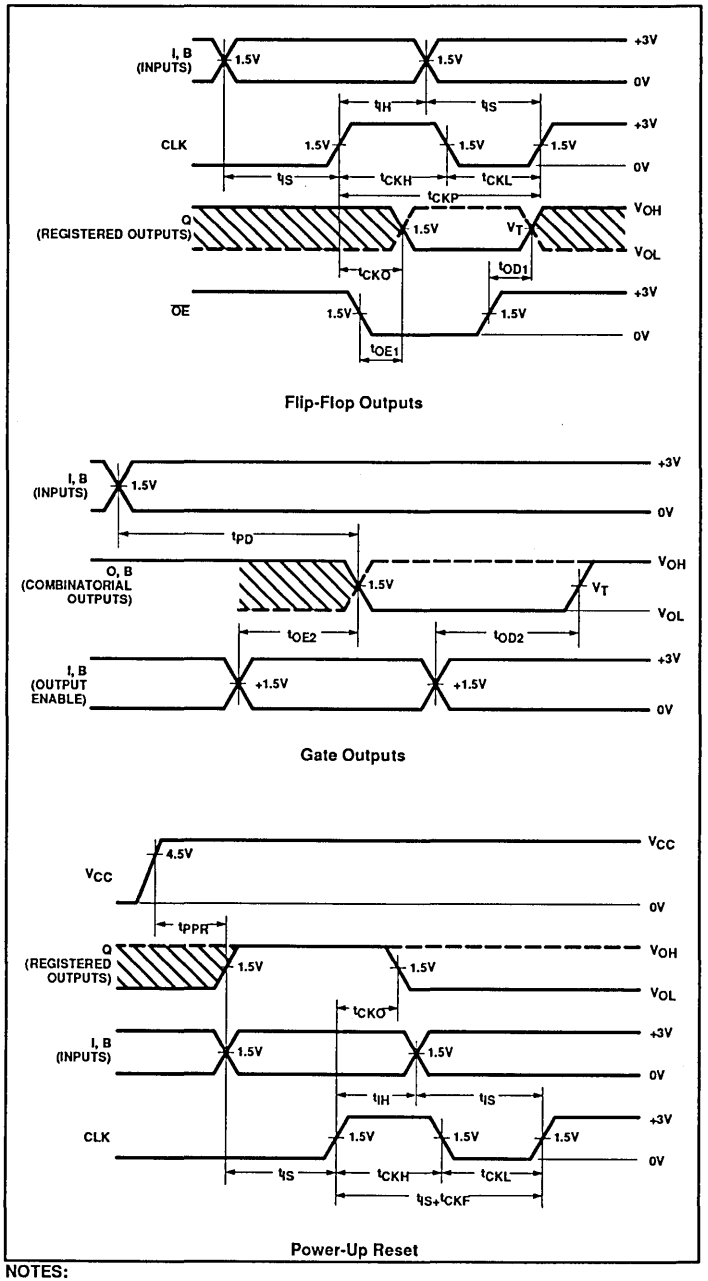
CLOCK TO FEEDBACK PATH



PAL-type devices 20L8, 20R8, 20R6, 20R4

PLQ20R8–5 Series

TIMING DIAGRAMS[1, 2]



NOTES:
1. Input pulse amplitude is 0V to 3V.
2. Input rise and fall times are 2.0ns typical.

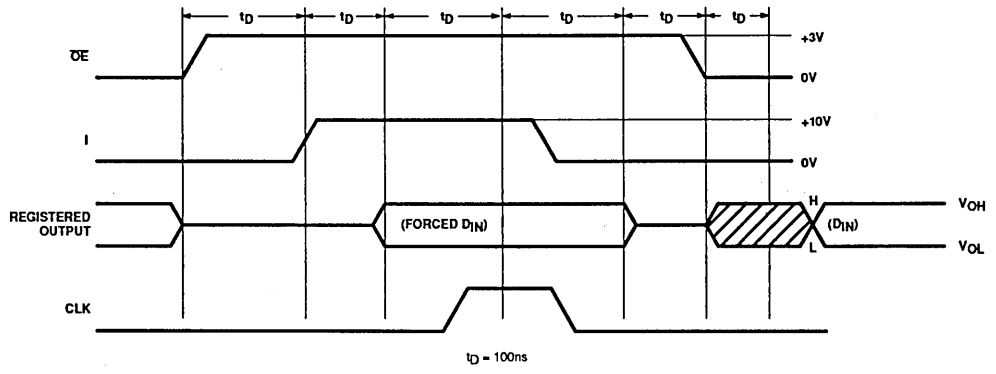
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{IS}	Required delay between beginning of valid input and positive transition of clock.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKF}	Delay between positive transition of clock and when internal $\overline{Q}$ output of flip-flop becomes valid.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the Off-State.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the Off-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.
t_D	Delay between each input change

FREQUENCY DEFINITIONS

f_{MAX}	**No feedback:** Determined by the minimum clock period, $1/(t_{CKL} + t_{CKH})$. **Internal feedback:** Determined by the internal delay from flip-flop outputs through the internal feedback and array to the flip-flop inputs, $1/(t_{IS} + t_{CKF})$. **External feedback:** Determined by clock-to-output delay and input setup time, $1/(t_{IS} + t_{CKO})$.

TIMING DIAGRAMS (Continued)



Output Register Preload

PROGRAMMING

The PLQ20XX Series are programmable on conventional programmers for 24-pin PAL devices. Refer to the following charts for qualified manufacturers of programmers and software tools:

PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	SYSTEM 29B, LogicPak™ 303A–V04 ADAPTER 303A–011A–V08 (DIP) 303A–011B–V04 (PLCC) UNISITE 40/48, V2.3 (DIP) V2.5 (PLCC) MODEL 60, 60A/H, V.13	20L8–7/20L8D : 1B/26 20R8–7/20R8D : 1B/27 20R6–7/20R6D : 1B/27 20R4–7/20R4D : 1B/27
STAG MICROSYSTEMS, INC. 1600 WYATT DRIVE SUITE 3 SANTA CLARA, CALIFORNIA 95054 (408)988–1118	ZL30/30A PROGRAMMER REV. 30A31 PPZ PROGRAMMER TBA	20L8–7/20L8D : 11/56 20R8–7/20R8D : 11/57 20R6–7/20R6D : 11/57 20R4–7/20R4D : 11/57

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
SIGNETICS COMPANY 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	AMAZE SOFTWARE REV. 1.8 AND LATER
DATA I/O 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	ABEL™ SOFTWARE REV. 1.0 AND LATER
LOGICAL DEVICES, INC. 1201 NORTHWEST 65TH PLACE FORT LAUDERDALE, FLORIDA 33309 (800)331–7766	CUPL™ SOFTWARE REV. 1.01 AND LATER

Philips Components–Signetics

Document No.	
ECN No.	
Date of Issue	November 1990
Status	Preliminary Specification
Programmable Logic Devices	

PLQ22V10–7

BiCMOS versatile PAL®-type device

DESCRIPTION

The PLQ22V10 is a versatile PAL-type device fabricated through the use of our BiCMOS process known as QUBiC. This is an excellent device where fast propagation delays are required.

The PLQ22V10 uses the familiar AND/OR logic array structure, which allows direct implementation of sum-of-product equations. This device has a programmable AND array driving a fixed OR array. This AND array is programmed to create custom product terms while the fixed OR array sums selected terms at the output.

The OR sum of the products feeds the "Output Macro Cell" (OMC) which can be individually configured as a dedicated input, a combinatorial output, or a registered output with internal feedback. In other words, the architecture provides maximum design flexibility by allowing the Output Macro Cell to be configured by the user.

This device is pin and JEDEC file compatible with industry standard 22V10 and can be used in all standard applications where speed is to be maximized.

Order codes can be found in the Ordering Information table.

FEATURES

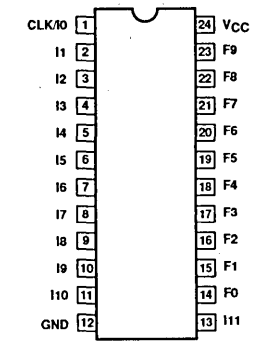
- Ultra fast 7.5ns t_{PD} and 6ns t_{CKO}
- Pin and JEDEC file compatible to industry standard 22V10
- 24-Pin Versatile Programmable Array Logic
- 10 input/output macro cells for architectural flexibility
- Varied product term distribution with up to 16 product terms per output for complex functions
- Programmable output polarity
- Power-up reset on all registers
- Synchronous Preset/ Asynchronous Reset
- Programmable on standard PAL-type device programmers
- Design support provided using SNAP or SLICE software development packages and other CAD tools for PLDs
- Available in 300mil-wide 24-Pin Plastic DIP and 28-Pin PLCC

APPLICATIONS

- DMA control
- State machine implementation
- High speed graphics processing
- Counters/shift registers
- SSI/MSI random logic replacement
- High speed memory decoder

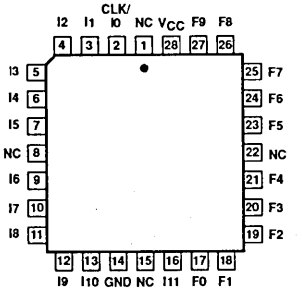
PIN CONFIGURATIONS

F and N Packages



N = Plastic DIP (300mil-wide)
F = Ceramic DIP (300mil-wide)

A Package



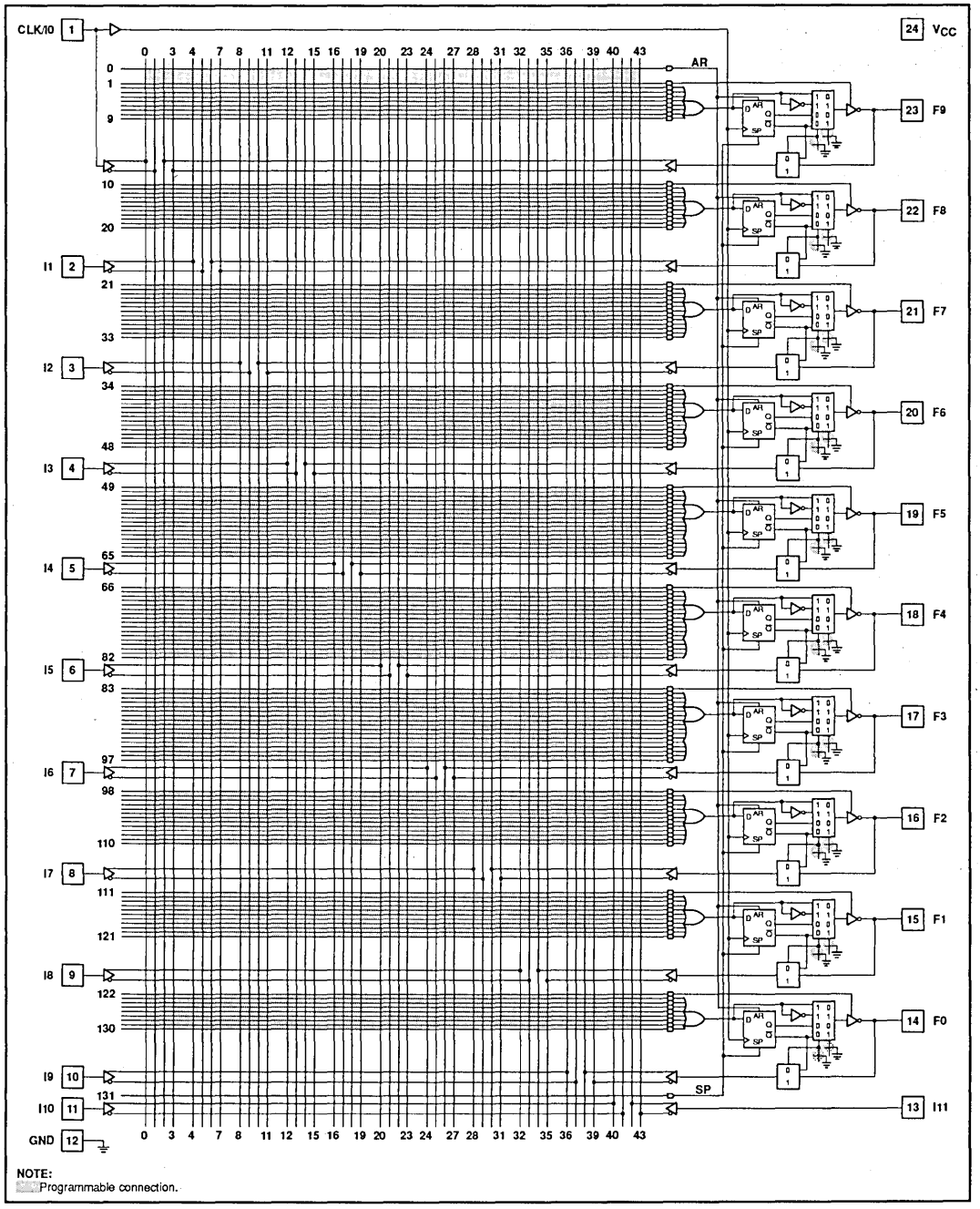
A = Plastic Leaded Chip Carrier

PIN LABEL DESCRIPTIONS

I1 – I11	Dedicated Input
NC	Not Connected
F0 – F9	Macro Cell Input/Output
CLK/I0	Clock Input/Dedicated Input
V_{CC}	Supply Voltage
GND	Ground

BiCMOS versatile PAL-type device PLQ22V10–7

LOGIC DIAGRAM



NOTE:
Programmable connection.

BiCMOS versatile PAL-type device PLQ22V10–7

FUNCTIONAL DESCRIPTION

The PLQ22V10 allows the systems engineer to implement the design on-chip, by opening fuse links to configure AND and OR gates within the device, according to the desired logic function.

Product terms with all fuses opened assume the logical HIGH state; product terms connected to both True and Complement of any single input assume the logical LOW state.

The PLQ22V10 has 12 inputs and 10 I/O Macro Cells (Figure 1.). The Macro Cell allows one of four potential output configurations, registered output or combinatorial I/O, Active-HIGH or Active-LOW (see Figure 2.). The configuration choice is made according to the user's design specification and corresponding programming of the configuration bits S_0 – S_1. Multiplexer controls are connected to ground (0) through a programmable fuse link, selecting the "0" path through the multiplexer. Programming the fuse disconnects the control line from GND and it floats to V_{CC} (1), selecting the "1" path.

The device is produced with a fuse link at each input to the AND gate array, and connections may be selectively removed by applying appropriate voltages to the circuit. Utilizing an easily-implemented programming algorithm, these products can be rapidly programmed to any customized pattern. Information on approved programmers can be found in the Programmer Reference Guide. Extra test fuses are pre-programmed during manufacturing to ensure extremely high field programming yields, and provide extra test paths to achieve excellent parametric correlation.

Variable Input/Output Pin Ratio

The PLQ22V10 has twelve dedicated input lines, and each Macro Cell output can be an I/O pin. Buffers for device inputs have complementary outputs to provide user-programmable input signal polarity. Unused input pins should be tied to V_{CC} or GND.

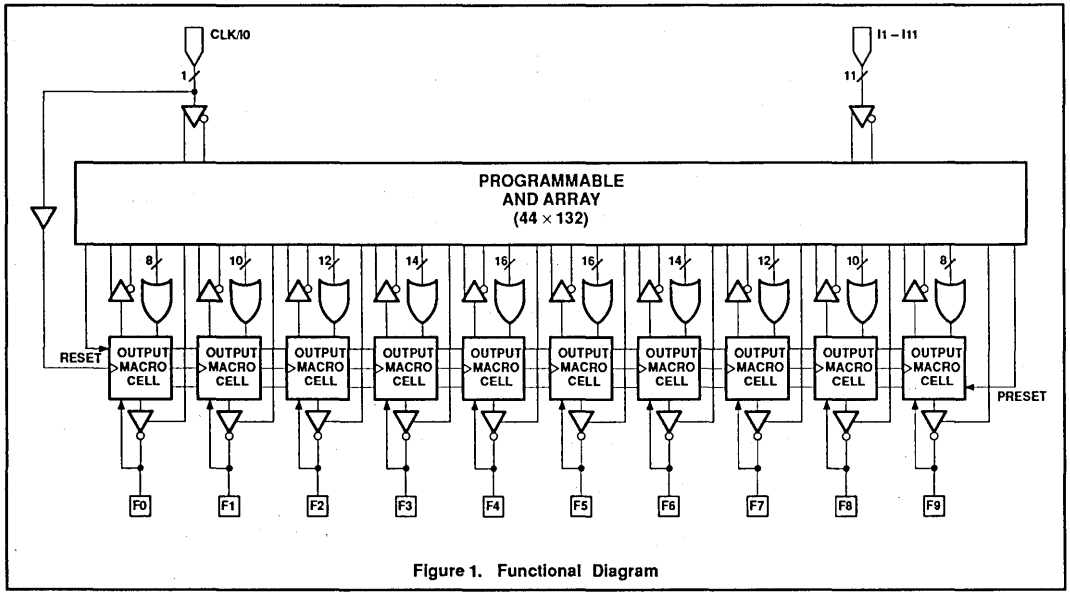
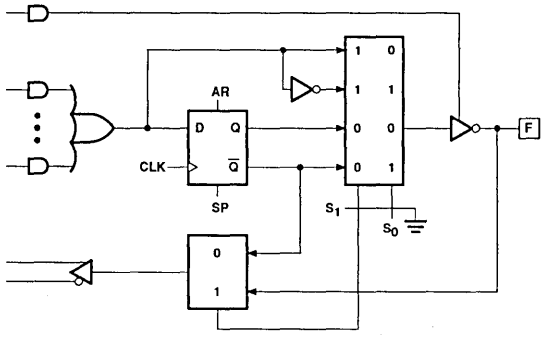


Figure 1. Functional Diagram

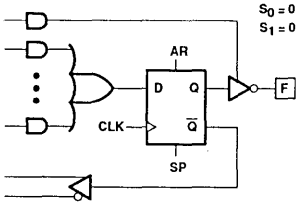
OUTPUT MACRO CELL



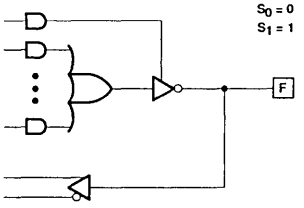
S_1	S_0	OUTPUT CONFIGURATION
0	0	Registered/Active-LOW
0	1	Registered/Active-HIGH
1	0	Combinatorial/Active-LOW
1	1	Combinatorial/Active-HIGH

0 = Unprogrammed fuse
1 = Programmed fuse

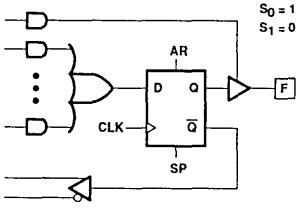
Figure 2. Output Macro Cell Logic Diagram



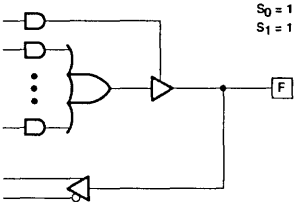
a. Registered/Active-LOW



c. Combinatorial/Active-LOW



b. Registered/Active-HIGH



d. Combinatorial/Active-HIGH

Figure 3.

Registered Output Configuration

Each Macro Cell of the PLQ22V10 includes a D-type flip-flop for data storage and synchronization. The flip-flop is loaded on the LOW-to-HIGH transition of the clock input. In the registered configuration (S_1 = 0), the array feedback is from $\overline{Q}$ of the flip-flop.

Combinatorial I/O Configuration

Any Macro Cell can be configured as combinatorial by selecting the multiplexer path that bypasses the flip-flop (S_1 = 1). In the combinatorial configuration, the feedback is from the pin.

BiCMOS versatile PAL-type device PLQ22V10–7

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic Dual-In-Line 300mil-wide	PLQ22V10–7N
24-Pin Ceramic Dual-In-Line 300mil-wide	PLQ22V10–7F
28-Pin Plastic Leaded Chip Carrier (PLCC)	PLQ22V10–7A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7.0	V_{DC}
V_{IN}	Input voltage	–1.2	V_{CC} + 0.5	V_{DC}
V_{OUT}	Output voltage	–0.5	V_{CC} + 0.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS Over commercial operating range unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS[1]	LIMITS		UNIT
			MIN	MAX	
Input voltage					
V_{IL}	Low	V_{CC} = MIN		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V
V_I	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–1.2	V
Output voltage					
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}			
V_{OL}	Low	I_{OL} = 16mA		0.5	V
V_{OH}	High	I_{OH} = –3.2 mA	2.4		V
Input current					
		V_{CC} = MAX			
I_{IL} (except Pin 1)	Low	V_{IN} = 0.40V		–100	µA
I_{IL} (Pin 1)	Low	V_{IN} = 0.40V		–150	µA
I_{IH}	High	V_{IN} = 2.7V		25	µA
I_I	Maximum input current	V_{IN} = 5.5V		1.0	mA
Output current					
		V_{CC} = MAX			
I_{OZH}	Output leakage[3]	V_{IN} = V_{IL} or V_{IH}, V_{OUT} = 2.7V		100	µA
I_{OZL}	Output leakage[3]	V_{IN} = V_{IL} or V_{IH}, V_{OUT} = 0.4V		–100	µA
I_{SC}	Short circuit[2]	V_{OUT} = 0.5V	–30	–130	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		210	mA

NOTES:
1. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
2. No more than one output should be tested at a time. Duration of the short-circuit test should not exceed one second. V_{OUT} = 0.5V has been chosen to avoid test problems caused by tester ground degradation.
3. I/O pin leakage is the worst case of I_{OZX} or I_{IX} (where X = H or L).

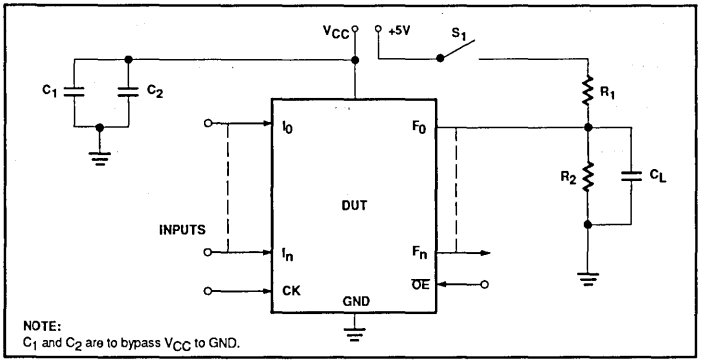
AC ELECTRICAL CHARACTERISTICS Over commercial operating range unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS		LIMITS[1] MIN	TYP	MAX	UNIT
t_{PD}	Input or feedback to non-registered output[2]	Active-LOW				7.5	
		Active-HIGH				7.5	ns
t_S	Setup time from input, feedback or SP to Clock			5.5			ns
t_H	Hold time			0			ns
t_{CO}	Clock to output					6.0	ns
t_{CF}	Clock to feedback[3]					2.5	ns
t_{AR}	Asynchronous Reset to registered output					10.0	ns
t_{ARW}	Asynchronous Reset width			7.5			ns
t_{ARR}	Asynchronous Reset recovery time			5.5			ns
t_{SPR}	Synchronous Preset recovery time			5.5			ns
t_{WL}	Width of Clock LOW			4.0			ns
t_{WH}	Width of Clock HIGH			4.0			ns
f_{MAX}	Maximum frequency; External feedback $1/(t_S + t_{CO})$[4]			87			MHz
	Maximum frequency; Internal feedback $1/(t_S + t_{CF})$[4]			125			MHz
t_{EA}	Input to Output Enable[5]					9.0	ns
t_{ER}	Input to Output Disable[5]					9.0	ns
Capacitance[6]							
C_{IN}	Input Capacitance (Pin 1)	V_{IN} = 2.0V	V_{CC} = 5.0V		6		pF
	Input Capacitance (Others)	V_{IN} = 2.0V	T_{amb} = 25°C		6		pF
C_{OUT}	Output Capacitance	V_{OUT} = 2.0V	f = 1MHz		8		pF

NOTES:

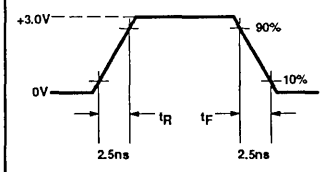
1. Commercial Test Conditions: $R_1 = 300\Omega$, $R_2 = 390\Omega$ (see Test Load Circuit).
2. t_{PD} is tested with switch S_1 closed and C_L = 50pF (including jig capacitance). V_{IH} = 3V, V_{IL} = 0V, V_T = 1.5V.
3. Calculated from measured f_{MAX} internal.
4. These parameters are not 100% tested, but are calculated at initial characterization and at any time the design is modified where frequency may be affected.
5. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
6. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modifed where capacitance may be affected.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

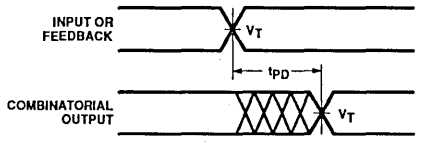
VOLTAGE WAVEFORM



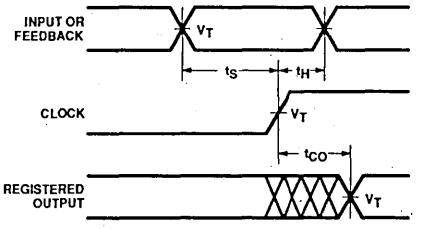
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

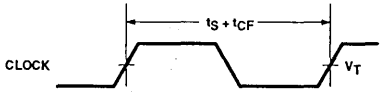
SWITCHING WAVEFORMS



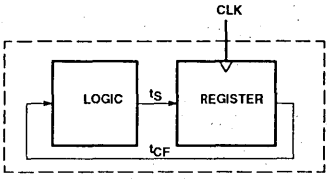
Combinatorial Output

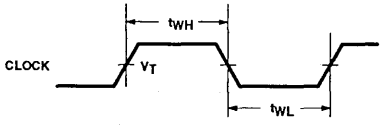


Registered Output

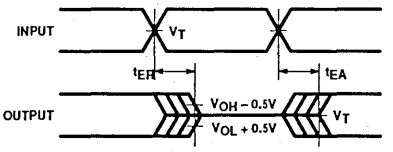


Clock to Feedback (f_{MAX} Internal)
(See Path at Right)

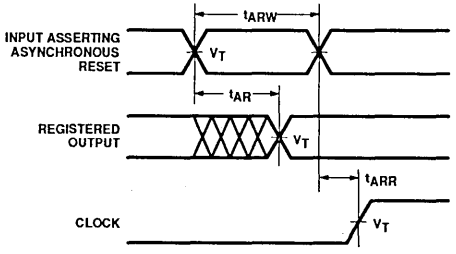




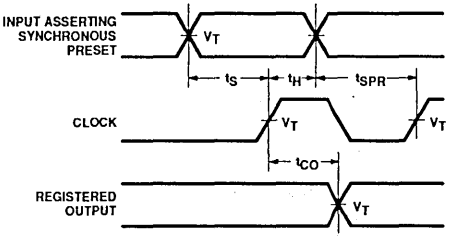
Clock Width



Input to Output Disable/Enable



Asynchronous Reset



Synchronous Preset

NOTES:
1. V_T = 1.5V.
2. Input pulse amplitude 0V to 3.0V.
3. Input rise and fall times 2.5ns max.

BiCMOS versatile PAL-type device PLQ22V10–7

Programmable 3-State Outputs

Each output has a 3-State output buffer with 3-State control. A product term controls the buffer, allowing enable and disable to be a function of any product of device inputs or output feedback. The combinatorial output provides a bidirectional I/O pin, and may be configured as a dedicated input if the buffer is always disabled.

Programmable Output Polarity

The polarity of each macro cell output can be Active-HIGH or Active-LOW, either to match output signal needs or to reduce product terms. Programmable polarity allows Boolean expressions to be written in their most compact form (true or inverted), and the output can still be of the desired polarity. It can also save "DeMorganizing" efforts.

Selection is controlled by programmable bit S_0 in the Output Macro Cell, and affects both registered and combinatorial outputs. Selection is automatic, based on the design specification and pin definitions. If the pin definition and output equation have the same polarity, the output is programmed to be Active-HIGH ($S_0 = 1$).

Preset/Reset

For initialization, the PLQ22V10 has additional Preset and Reset product terms. These terms are connected to all registered outputs. When the Synchronous Preset (SP) product term is asserted high, the output registers will be loaded with a HIGH on the next LOW-to-HIGH clock transition. When the Asynchronous Reset (AR) product term is asserted high, the output registers will be immediately loaded with a LOW independent of the clock.

Note that Preset and Reset control the flip-flop, not the output pin. The output level is determined by the output polarity selected.

Power-Up Reset

All flip-flops power-up to a logic LOW for predictable system initialization. Outputs of the PLQ22V10 will depend on the programmed output polarity. The V_{CC} rise must be monotonic and the reset delay time is 1–10μs maximum.

Register Preload

The register on the PLQ22V10 can be preloaded from the output pins to facilitate functional testing of complex state machine designs. This feature allows direct loading of arbitrary states, making it unnecessary to cycle through long test vector sequences to reach a desired state. In addition, transitions from illegal states can be verified by loading illegal states and observing proper recovery.

Security Fuse

After programming and verification, a PLQ22V10 design can be secured by programming the security fuse link. Once programmed, this fuse defeats readback of the internal programmed pattern by a device programmer, securing proprietary designs from competitors. When the security fuse is programmed, the array will read as if every fuse is programmed.

Quality and Testability

The PLQ22V10 offers a very high level of built-in quality. Extra programmable fuses provide a means of verifying performance of all AC and DC parameters. In addition, this verifies programmability and functionality of the device to provide the highest programming and post-programming functional yields.

Technology

The BiCMOS PLQ22V10 is fabricated with the Philips Components–Signetics process known as QUBiC. QUBiC combines an advanced, state-of-the-art 1.0μm (drawn feature size) CMOS process with an ultra fast bipolar process to achieve superior speed and drive capabilities. QUBiC incorporates three layers of Al/Cu interconnects for reduced chip size, and our proven Ti-W fuse technology ensures highest programming yields.

Programming

The PLQ22V10–7 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLQ22V10–7 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLQ22V10–7 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SLICE only. The SLICE design package is available, free of charge, to qualified users.

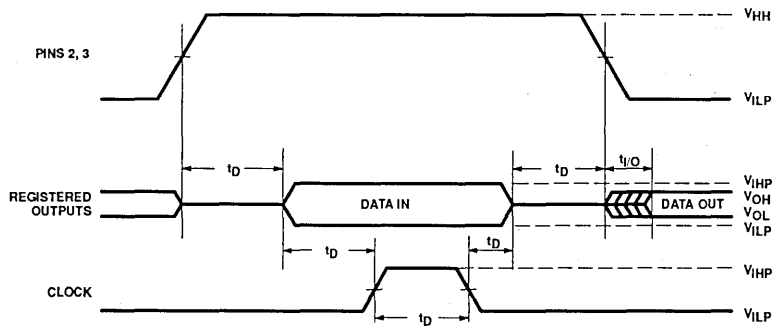
ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

OUTPUT REGISTER PRELOAD

The preload function allows the registers to be loaded from the output pins. This feature aids functional testing of sequential designs by allowing direct setting of output states. The procedure for preloading follows:

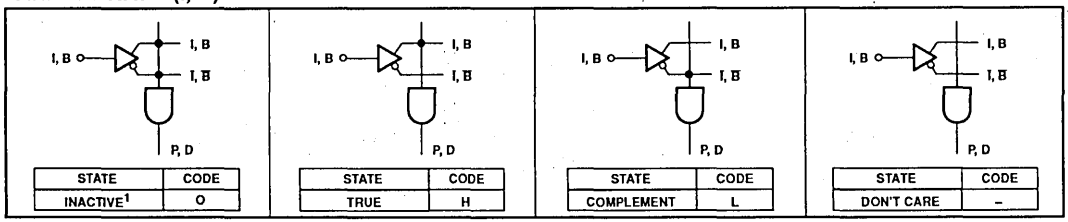
1. Raise V_{CC} to 5.0V ± 0.25V.
2. Set Pin 2 or 3 to V_{HH} to disable outputs and enable preload.
3. Apply the desired value (V_{ILP}/V_{IHP}) to all registered output pins. Leave combinatorial output pins floating.
4. Clock Pin 1 from V_{ILP} to V_{IHP}.
5. Remove V_{ILP}/V_{IHP} from all registered output pins.
6. Lower Pin 2 or 3 to V_{ILP}.
7. Enable the output registers according to the programmed pattern.
8. Verify V_{OL}/V_{OH} at all registered output pins. Note that the output pin signal will depend on the output polarity.

SYMBOL	PARAMETER	LIMITS			UNIT
		MIN	REC	MAX	
V_{HH}	Super-level input voltage	9.5	10	10.5	V
V_{ILP}	Low-level input voltage	0	0	0.5	V
V_{IHP}	High-level input voltage	2.4	5.0	5.5	V
t_D	Delay time	100	200	1000	ns
$t_{I/O}$	I/O valid after Pin 2 or 3 drops from V_{HH} to V_{ILP}	100			ns



Output Register Preload Waveform

"AND" ARRAY – (I, B)



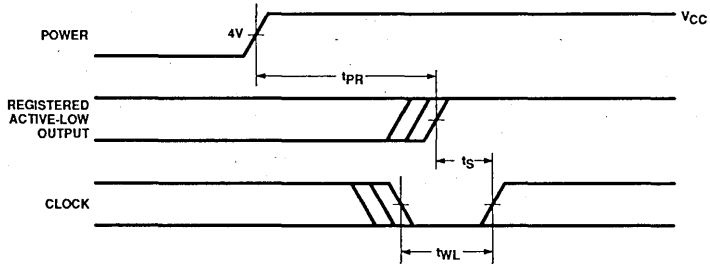
NOTE:
1. This is the initial state.

POWER-UP RESET

The power-up reset feature ensures that all flip-flops will be reset to LOW after the device has been powered up. The output state will depend on the programmed pattern. This feature is valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways V_{CC} can rise to its steady state, two conditions are required to ensure a valid power-up reset. These conditions are:

1. The V_{CC} rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

		LIMITS		
SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{PR}	Power-up Reset Time		1	µs
t_S	Input or Feedback Setup Time	See AC Electrical Characteristics		
t_{WL}	Clock Width LOW			



Power-Up Reset Waveform

PROGRAMMING

The PLQ22V10 Series are programmable on conventional programmers for 24-pin PAL devices. Refer to the following charts for qualified manufacturers of programmers and software tools:

PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	SYSTEM 29B, LogicPak™ 303A–V04 ADAPTER 303A–011A–V08 (DIP) 303A–011B–V04 (PLCC) UNISITE 40/48, V2.3 (DIP) V2.5 (PLCC) MODEL 60, 60A/H, V.13	22V10 : TBD
STAG MICROSYSTEMS, INC. 1600 WYATT DRIVE SUITE 3 SANTA CLARA, CALIFORNIA 95054 (408)988–1118	ZL30/30A PROGRAMMER REV. 30A31 PPZ PROGRAMMER TBA	22V10 : TBD

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
SIGNETICS COMPANY 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SLICE 1.0 AND LATER SNAP 1.6 AND LATER
DATA I/O 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	ABEL™ SOFTWARE REV. 1.0 AND LATER
LOGICAL DEVICES, INC. 1201 NORTHWEST 65TH PLACE FORT LAUDERDALE, FLORIDA 33309 (800)331–7766	CUPL™ SOFTWARE REV. 1.01 AND LATER

BiCMOS versatile PAL-type device PLQ22V10–7

PROGRAM TABLE

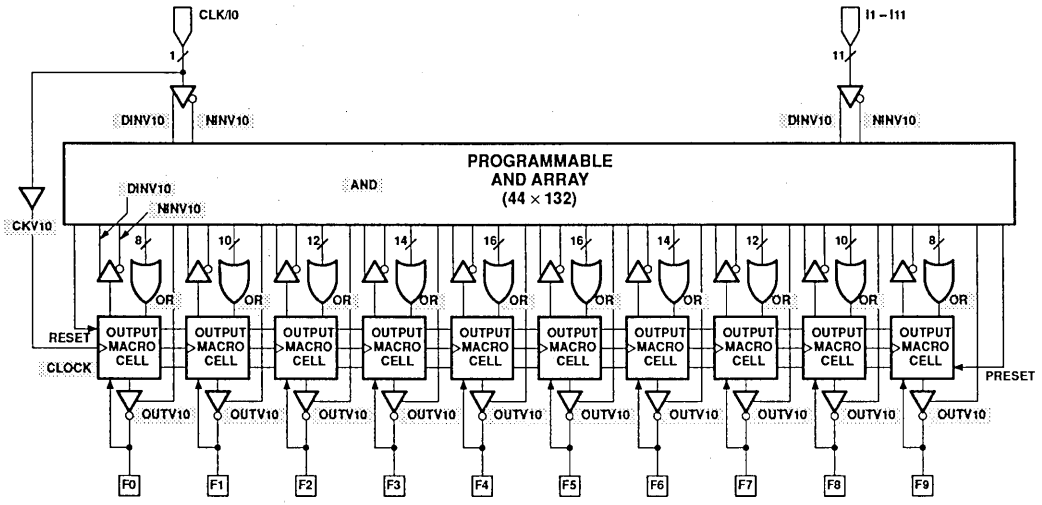
	CONTROL WORD										POLARITY									
TERM	AND																			
	I										F(I)										F(O)									
	11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
0																							D									
1																							A									
2																							A									
3																							A									
4																							A									
5																							A									
6																							A									
7																							A									
8																							A									
9																								D								
10																								A								
11																								A								
12																								A								
13																								A								
14																								A								
15																								A								
16																								A								
17																								A								
18																								A								
19																								A								
20																									D							
21																									A							
22																									A							
23																									A							
24																									A							
25																									A							
26																									A							
27																									A							
28																									A							
29																									A							
30																									A							
31																									A							
32																									A							
33																										D						
34																										A						
35																										A						
36																										A						
37																										A						
38																										A						
39																										A						
40																										A						
41																										A						
42																										A						
43																										A						
44																										A						
45																										A						
46																										A						
47																										A						
48																											D					
49																											A					
50																											A					
51																											A					
52																											A					
53																											A					
54																											A					
55																											A					
56																											A					
57																											A					
58																											A					
59																											A					
60																											A					
61																											A					
62																											A					
63																											A					
64																											A					
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

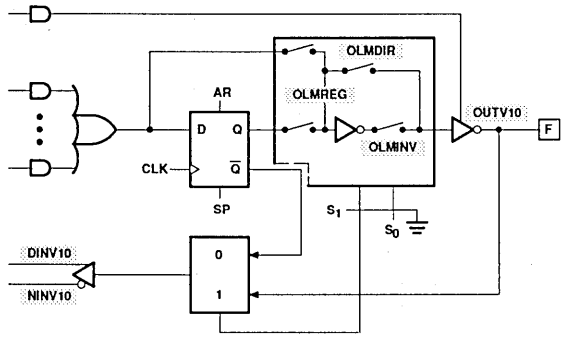
BiCMOS versatile PAL-type device PLQ22V10–7

PROGRAM TABLE (Continued)

TERM	AND I 11	10	9	8	7	6	5	4	3	2	1	0	F(I) 9	8	7	6	5	4	3	2	1	0	F(O) 9	8	7	6	5	4	3	2	1	0
65																												D				
66																												A				
67																												A				
68																												A				
69																												A				
70																												A				
71																												A				
72																												A				
73																												A				
74																												A				
75																												A				
76																												A				
77																												A				
78																												A				
79																												A				
80																												A				
81																												A				
82																													D			
83																													A			
84																													A			
85																													A			
86																													A			
87																													A			
88																													A			
89																													A			
90																													A			
91																													A			
92																													A			
93																													A			
94																													A			
95																													A			
96																													A			
97																														D		
98																														A		
99																														A		
100																														A		
101																														A		
102																														A		
103																														A		
104																														A		
105																														A		
106																														A		
107																														A		
108																														A		
109																														A		
110																															D	
111																															A	
112																															A	
113																															A	
114																															A	
115																															A	
116																															A	
117																															A	
118																															A	
119																															A	
120																															A	
121																																D
122																																A
123																																A
124																																A
125																																A
126																																A
127																																A
128																																A
129																																A
AR																																
SP																																
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

SNAP RESOURCE SUMMARY DESIGNATIONS





Output Macro Cell

Document No.	853–1359
ECN No.	99791
Date of Issue	June 14, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS20R8D/–7 Series

PAL®-type devices

20L8, 20R8, 20R6, 20R4

FEATURES

- Ultra high-speed
  - t_{PD} = 7.5ns and f_{MAX} = 74MHz for the PLUS20R8-7 Series
  - t_{PD} = 10ns and f_{MAX} = 60 MHz for the PLUS20R8D Series
- 100% functionally and pin-for-pin compatible with industry standard 24-pin PAL ICs
- Power-up reset function to enhance state machine design and testability
- Design support provided via AMAZE and other CAD tools for Series 24 PAL devices
- Field-programmable on industry standard programmers
- Security fuse
- Individual 3-State control of all outputs

DESCRIPTION

The Signetics PLUS20XX family consists of ultra high-speed 7.5ns and 10ns versions of Series 24 PAL devices.

The PLUS20XX family is 100% functional and pin-compatible with the 20L8, 20R8, 20R6, and 20R4 Series devices.

The sum of products (AND-OR) architecture is comprised of 64 AND gates and 8 fixed OR gates. Multiple bidirectional pins provide variable input/output pin ratios. Individual 3-State control of all outputs and registers with feedback (R8, R6, R4) is also provided. Proprietary designs can be protected by programming the security fuse.

The PLUS20R8, R6, and R4 have D-type flip-flops which are loaded on the Low-to-High transition of the clock input.

In order to facilitate state machine design and testing, a power-up reset function has been incorporated into these devices to reset all internal registers to active-Low after a specific period of time.

The Signetics State-of-the-Art oxide isolation Bipolar fabrication process is employed to achieve high-performance operation.

The PLUS20XX family of devices are field programmable, enabling the user to quickly generate custom patterns using standard programming equipment. See the programmer chart for qualified programmers.

The AMAZE software package from Signetics supports easy design entry for the PLUS20XX series as well as other PLD devices from Signetics. The PLUS20XX series are also supported by other standard CAD tools for PAL-type devices.

Order codes are listed in the Ordering Information table.

DEVICE NUMBER	DEDICATED INPUTS	COMBINATORIAL OUTPUTS	REGISTERED OUTPUTS
PLUS20L8	14	8 (6 I/O)	0
PLUS20R8	12	0	8
PLUS20R6	12	2 I/O	6
PLUS20R4	12	4 I/O	4

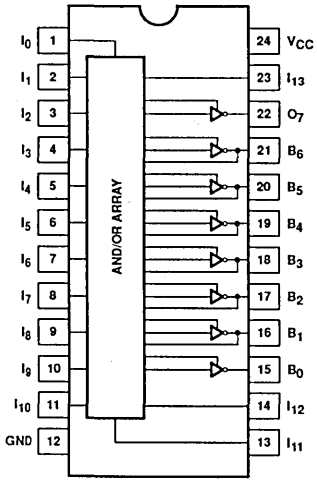
®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

PAL-type devices
20L8, 20R8, 20R6, 20R4

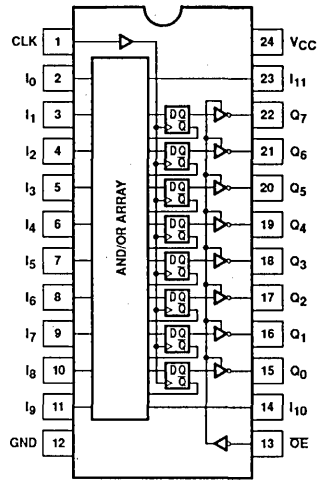
PLUS20R8D/–7 Series

PIN CONFIGURATIONS

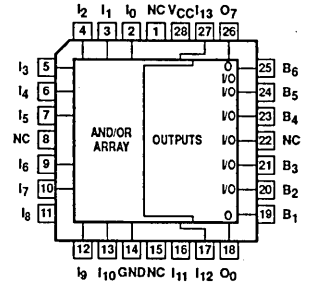
PLUS20L8



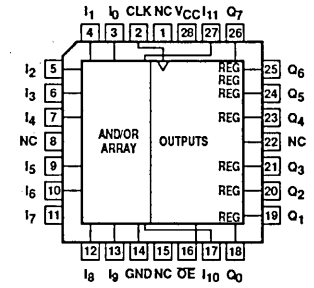
PLUS20R8



PLUS20L8



PLUS20R8



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

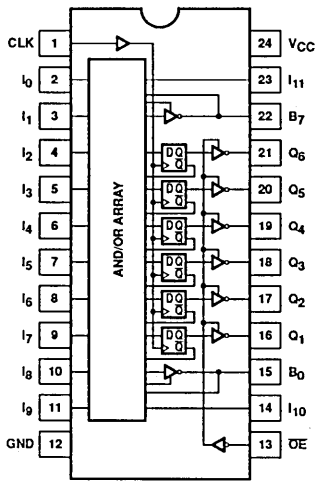
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

PAL-type devices
20L8, 20R8, 20R6, 20R4

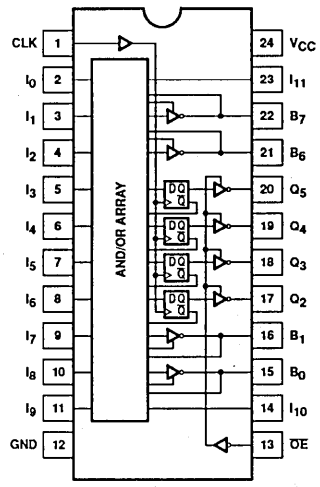
PLUS20R8D/–7 Series

PIN CONFIGURATIONS

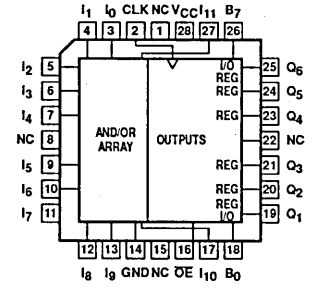
PLUS20R6



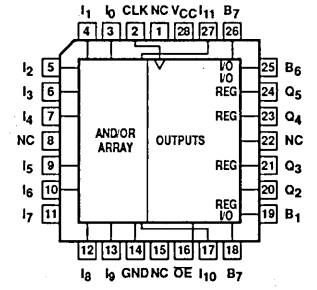
PLUS20R4



PLUS20R6



PLUS20R4

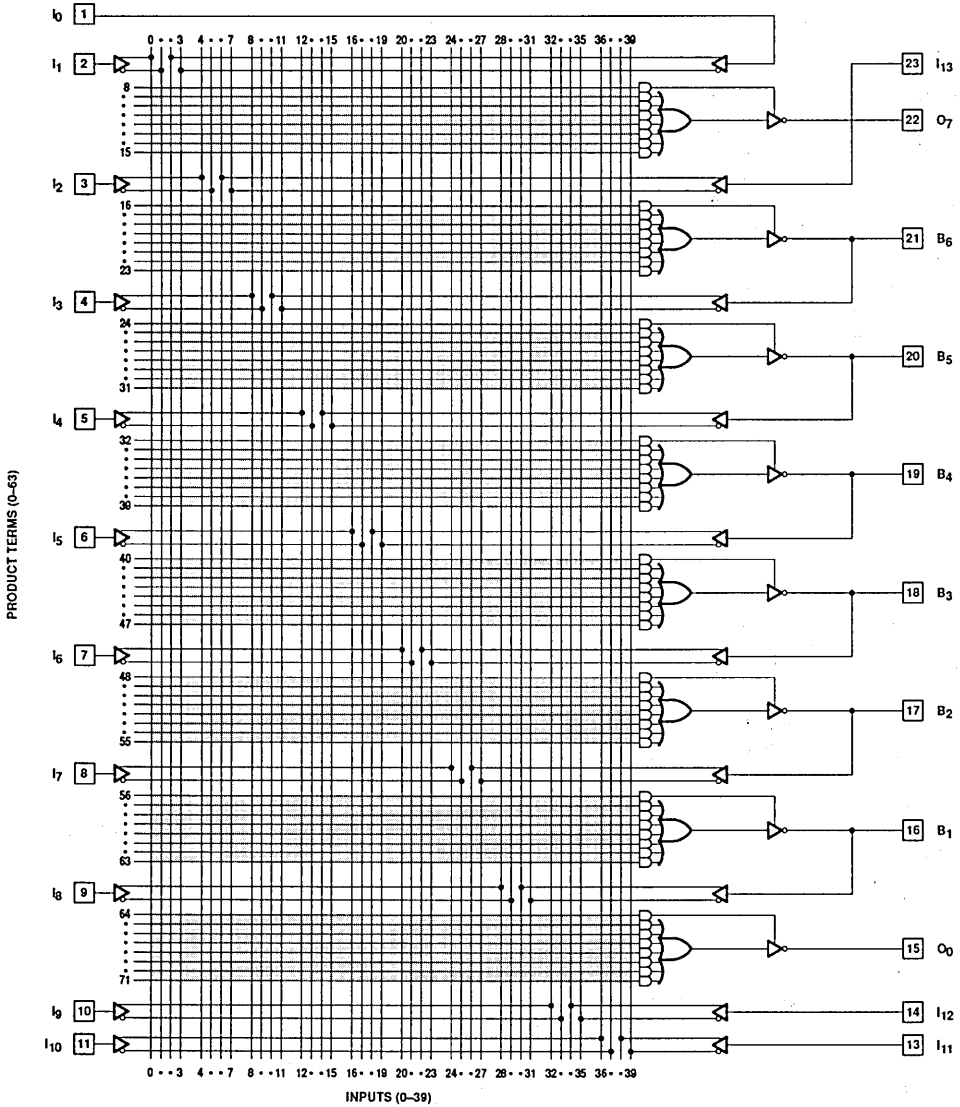


SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

LOGIC DIAGRAM

PLUS20L8



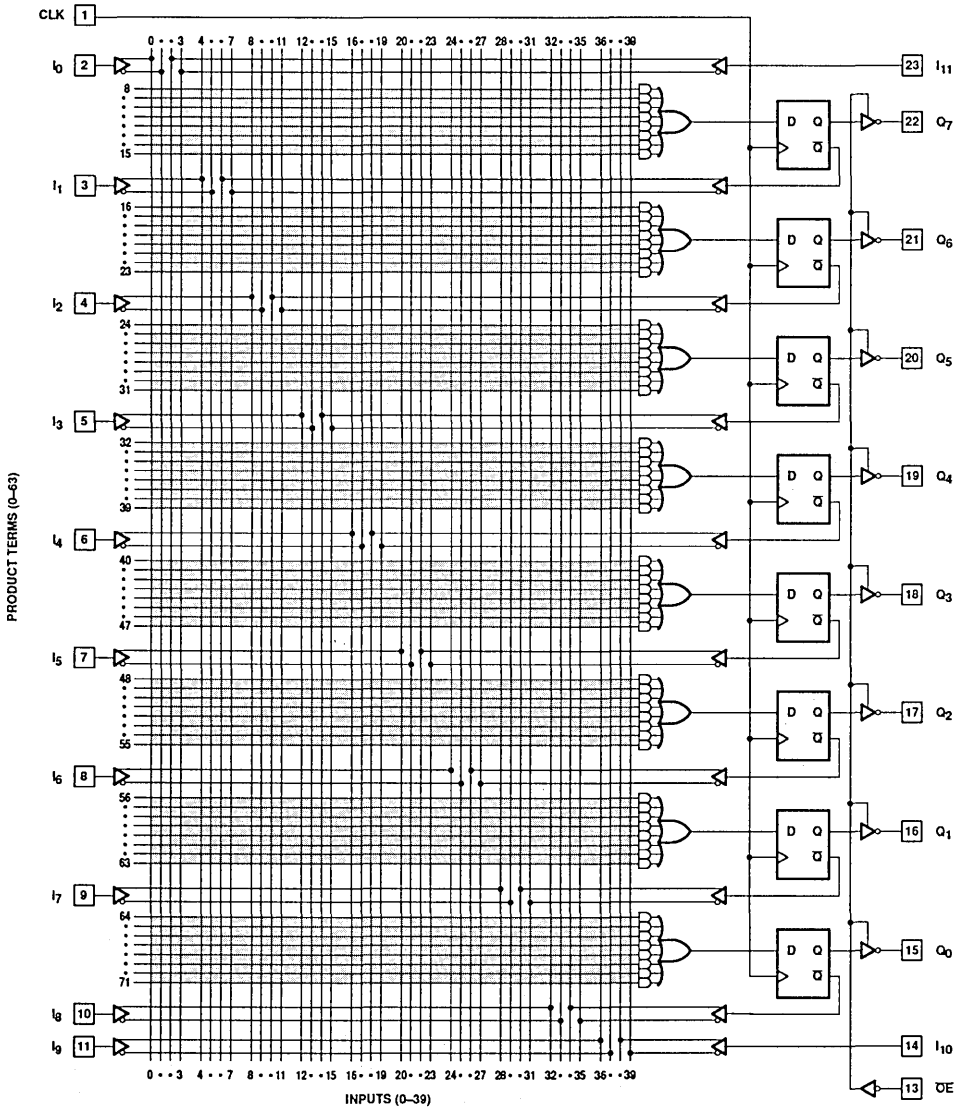
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒▒ Programmable connections.

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLUS20R8D/–7 Series

LOGIC DIAGRAM

PLUS20R8



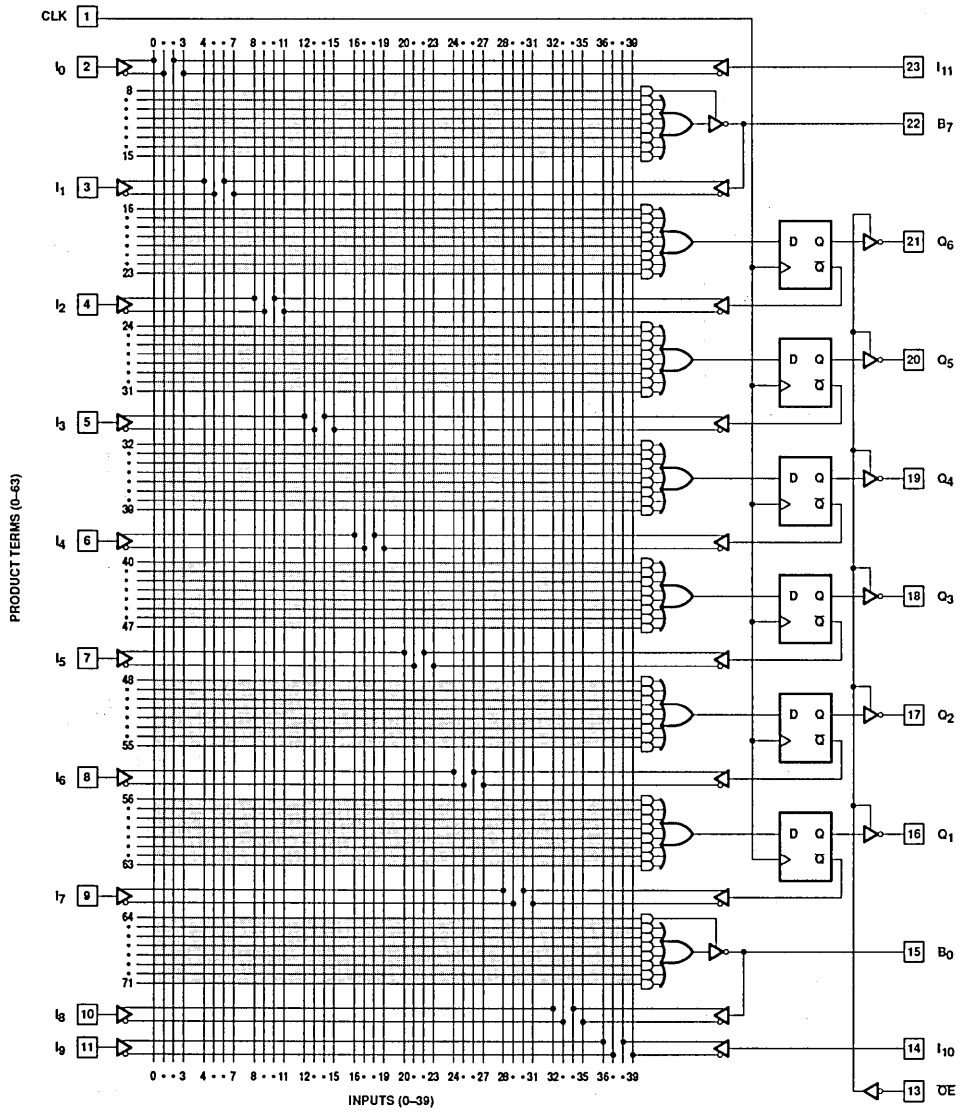
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒▒ Programmable connections.

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLUS20R8D/–7 Series

LOGIC DIAGRAM

PLUS20R6



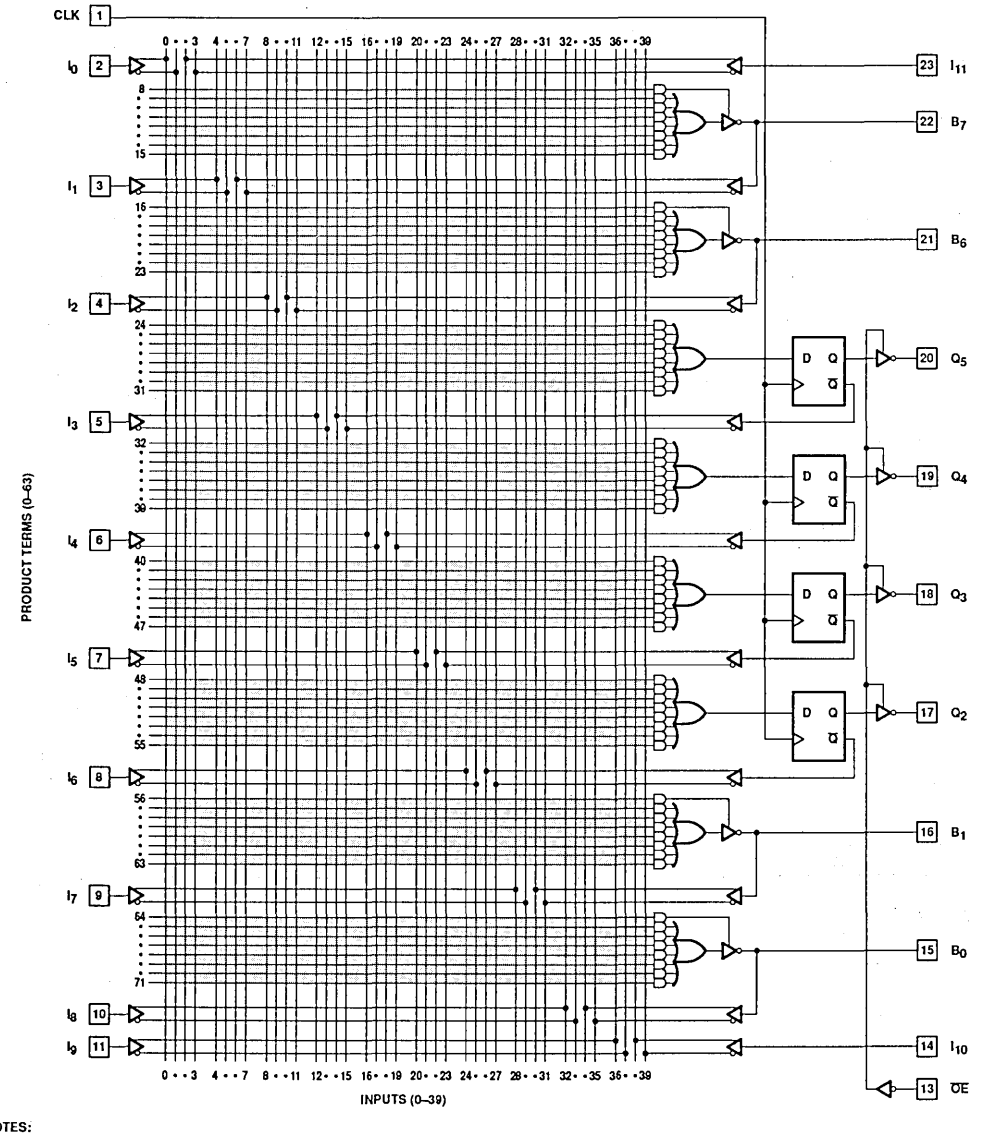
NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

PAL-type devices
20L8, 20R8, 20R6, 20R4

PLUS20R8D/–7 Series

LOGIC DIAGRAM

PLUS20R4



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. Programmable connections.

PAL-type devices 20L8, 20R8, 20R6, 20R4

PLUS20R8D/–7 Series

FUNCTIONAL DESCRIPTIONS

The PLUS20XX series utilizes the familiar sum-of-products implementation consisting of a programmable AND array and a fixed OR array. These devices are capable of replacing an equivalent of four or more SSI/MSI integrated circuits to reduce package count and board area occupancy, consequently improving reliability and design cycle over Standard Cell or gate array options. By programming the security fuse, proprietary designs can be protected from duplication.

The PLUS20XX series consists of four PAL-type devices. Depending on the particular device type, there are a variable number of combinatorial and registered outputs available to the designer. The PLUS20L8 is a combinatorial part with 8 user configurable outputs (6 bidirectional), while the other three devices, PLUS20R8, PLUS20R6, PLUS20R4, have respectively 8, 6, and 4 output registers.

3-State Outputs

The PLUS20XX series devices also feature 3-State output buffers on each output pin which can be programmed for individual control of all outputs. The registered outputs (Qn) are controlled by an external input (/OE), and the combinatorial outputs (On, Bn) use a product term to control the enable function.

Programmable Bidirectional Pins

The PLUS20XX products feature variable Input/Output ratios. In addition to 12 dedicated inputs, each combinatorial output pin of the registered devices can be individually programmed as an input or output. The PLUS20L8 provides 14 dedicated inputs and 6 Bidirectional I/O lines that can be individually configured as inputs or outputs.

Output Registers

The PLUS20R8 has 8 output registers, the 20R6 has 6, and the 20R4 has 4. Each output register is a D-type flip-flop which is loaded on the Low-to-High transition of the clock input. These output registers are capable of feeding the outputs of the registers back into the array to facilitate design of synchronous state machines.

Power-up Reset

By resetting all flip-flops to a logic Low, as the power is turned on, the PLUS20R8, R6, R4 enhance state machine design and initialization capability.

Software Support

Like other Programmable Logic Devices from Signetics, the PLUS20XX series are supported by AMAZE, the PC-based software development tool from Signetics. The PLUS20XX family of devices are also supported by standard CAD tools for PAL devices, including ABEL and CUPL.

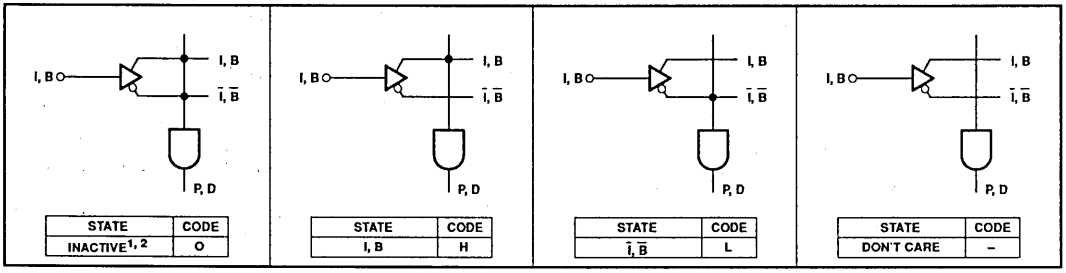
AMAZE is available free of charge to qualified users.

Logic Programming

The PLUS20XX series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLUS20XX architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

AND ARRAY – (I, B)



STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
$\overline{I}$, $\overline{B}$	L

STATE	CODE
DON'T CARE	–

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

PAL-type devices 20L8, 20R8, 20R6, 20R4

PLUS20R8D/–7 Series

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic Dual-In-Line 300mil-wide	PLUS20R8DN PLUS20R6DN PLUS20R4DN PLUS20L8DN PLUS20R8–7N PLUS20R6–7N PLUS20R4–7N PLUS20L8–7N
28-Pin Plastic Leaded Chip Carrier (PLCC)	PLUS20R8DA PLUS20R6DA PLUS20R4DA PLUS20L8DA PLUS20R8–7A PLUS20R6–7A PLUS20R4–7A PLUS20L8–7A

NOTE:
The PLUS20XX series of devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Signetics Military Data Book.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–1.2	+8.0	V_{DC}
V_{OUT}	Output voltage	–0.5	V_{CC} + 0.5V	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free–air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_A \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = 24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low[3]	V_{IN} = 0.40V	–250			µA
I_{IH}	High[3]	V_{IN} = 2.7V			25	µA
I_I	Maximum input current	V_{IN} = V_{CC} = V_{CCMAX}			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage	V_{OUT} = 2.7V			100	µA
I_{OZL}	Output leakage	V_{OUT} = 0.4V	–100			µA
I_{OS}	Short circuit[4, 5]	V_{OUT} = 0V	–30		–90	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		150	210	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5V				
		V_{OUT} = 2.0V		8		pF
C_B	I/O (B)	V_{OUT} = 2V, f = 1MHz		8		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Test one at a time.
5. Duration of short circuit should not exceed 1 second.
6. These parameters are not 100% tested but periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 200\Omega$, $R_2 = 390\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

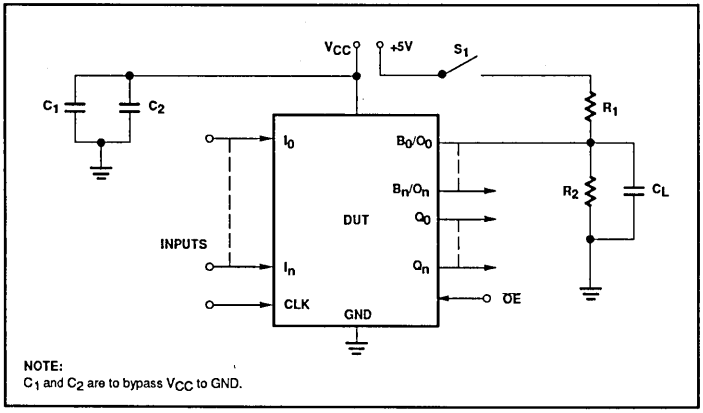
SYMBOL	PARAMETER	FROM	TO	LIMITS					UNIT
				–7			D		
				MIN[1]	TYP	MAX	MIN[1]	MAX	
Pulse Width									
t_{CKH}	Clock High	CK+	CK–	5			7		ns
t_{CKL}	Clock Low	CK–	CK+	5			7		ns
t_{CKP}	Period	CK+	CK+	10			14		ns
Setup & Hold time									
t_{IS}	Input	Input or feedback	CK+	7			9		ns
t_{IH}	Input	CK+	Input or feedback	0			0		ns
Propagation delay									
t_{CKO}	Clock	CK±	Q±	3		6.5	3	7.5	ns
t_{CKF}	Clock[3]	CK±	$\overline{Q}$			3		6.5	ns
t_{PD}	Output (20L8, R6, R4)[2]	I, B	Output	3		7.5	3	10	ns
t_{OE1}	Output enable[4]	$\overline{OE}$	Output enable	3		8	3	10	ns
t_{OE2}	Output enable[4,5]	I	Output enable	3		10	3	10	ns
t_{OD1}	Output disable[4]	$\overline{OE}$	Output disable	3		8	3	10	ns
t_{OD2}	Output disable[4,5]	I	Output disable	3		10	3	10	ns
t_{SKW}	Output	Q	Q			1		1	ns
t_{PPR}	Power-Up Reset	V_{CC}+	Q+			10		10	ns
Frequency (20R8, R6, R4)									
f_{MAX}	No feedback $1/(t_{CKL} + t_{CKH})$[6]				100		71.4		MHz
	Internal feedback $1/(t_{IS} + t_{CKF})$[6]				90		64.5		MHz
	External feedback $1/(t_{IS} + t_{CKO})$[6]				74		60.6		MHz

* For definitions of the terms, please refer to the Timing/Frequency Definitions tables.

NOTES:

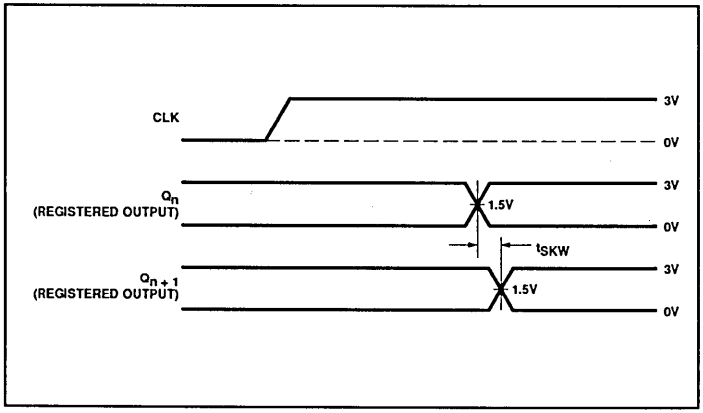
1. CL = 0pF while measuring minimum output delays.
2. t_{PD} test conditions: CL = 50pF (with jig and scope capacitance), $V_{IH} = 3V$, $V_{IL} = 0V$, $V_{OH} = V_{OL} = 1.5V$.
3. t_{CKF} was calculated from measured Internal f_{MAX}.
4. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
5. Same function as t_{OE1} and t_{OD1}, with the difference of using product term control.
6. Not 100% tested, but calculated at initial characterization and at any time a modification in design takes place which may affect the frequency.

TEST LOAD CIRCUIT

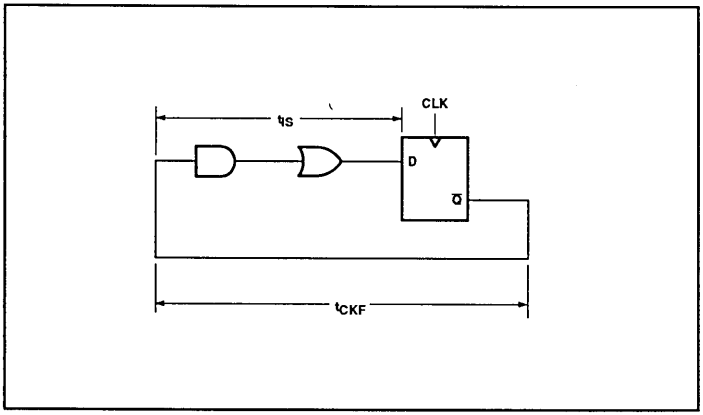


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

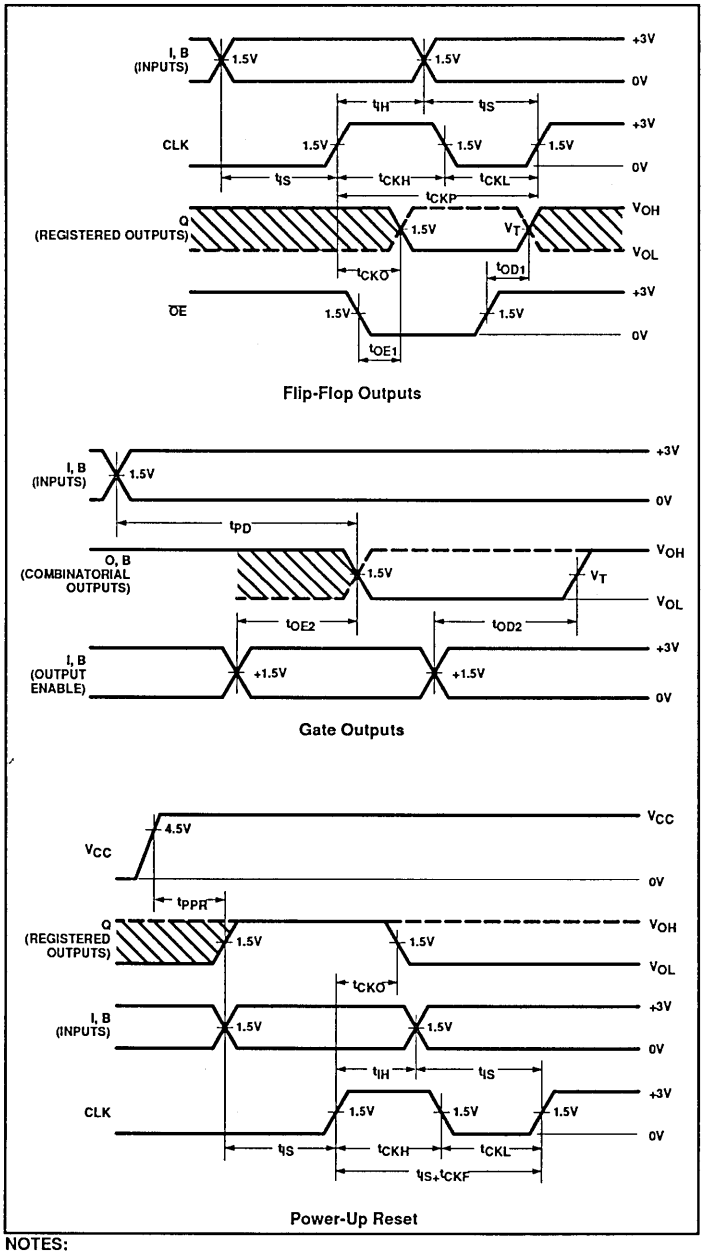
OUTPUT REGISTER SKEW



CLOCK TO FEEDBACK PATH



TIMING DIAGRAMS[1, 2]



NOTES:
1. Input pulse amplitude is 0V to 3V.
2. Input rise and fall times are 2.5ns.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{IS}	Required delay between beginning of valid input and positive transition of clock.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKF}	Delay between positive transition of clock and when internal $\overline{Q}$ output of flip-flop becomes valid.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the Off-State.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the Off-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.

FREQUENCY DEFINITIONS

f_{MAX}	**No feedback:** Determined by the minimum clock period, $1/(t_{CKL} + t_{CKH})$. **Internal feedback:** Determined by the internal delay from flip-flop outputs through the internal feedback and array to the flip-flop inputs, $1/(t_{IS} + t_{CKF})$. **External feedback:** Determined by clock-to-output delay and input setup time, $1/(t_{IS} + t_{CKO})$.

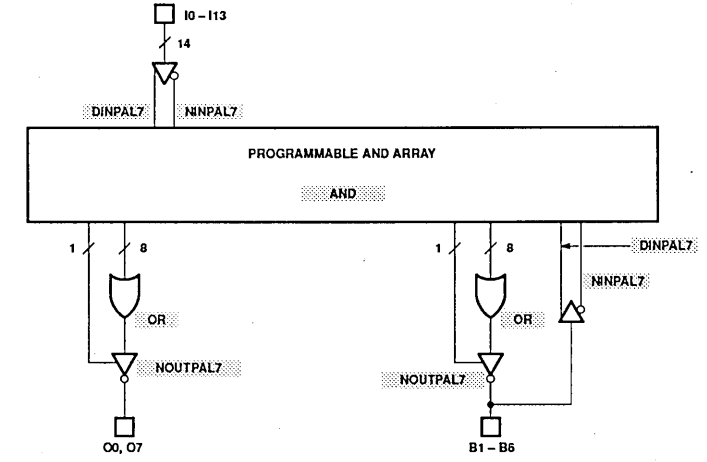
PROGRAMMING

The PLUS20XX Series are programmable on conventional programmers for 24-pin PAL® devices. Refer to the following charts for qualified manufacturers of programmers and software tools:

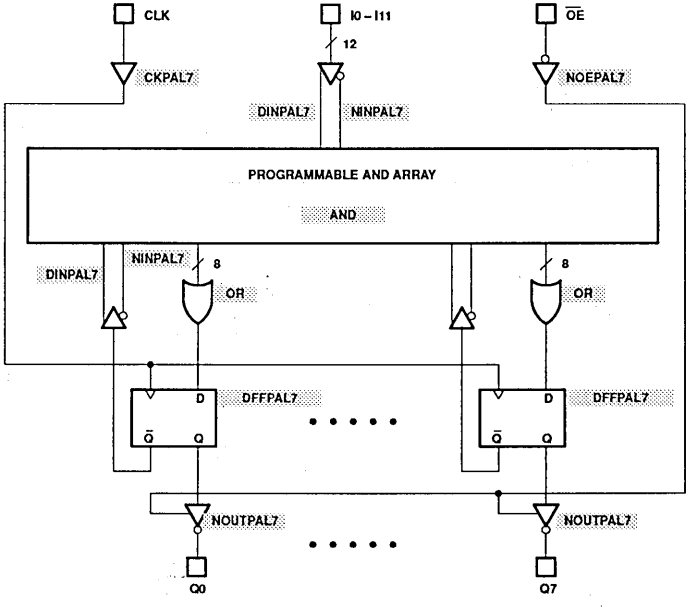
PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	SYSTEM 29B, LogicPak™ 303A–V04 ADAPTER 303A–011A–V08 (DIP) 303A–011B–V04 (PLCC) UNISITE 40/48, V2.3 (DIP) V2.5 (PLCC) MODEL 60, 60A/H, V.15.0	20L8–7/20L8D : 1B/26 20R8–7/20R8D : 1B/27 20R6–7/20R6D : 1B/27 20R4–7/20R4D : 1B/27
STAG MICROSYSTEMS, INC. 1600 WYATT DRIVE SUITE 3 SANTA CLARA, CALIFORNIA 95054 (408)988–1118	ZL30/30A PROGRAMMER REV. 30A31 (DIP) REV. 30A001 (PLCC) PPZ PROGRAMMER TBA	20L8–7/20L8D : 11/56 20R8–7/20R8D : 11/57 20R6–7/20R6D : 11/58 20R4–7/20R4D : 11/59

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
SIGNETICS COMPANY 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SNAP REV. 1.6 AND LATER SLICE REV. 1.0 AND LATER AMAZE SOFTWARE REV. 1.7
DATA I/O 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	ABEL™ SOFTWARE REV. 1.0 AND LATER
LOGICAL DEVICES, INC. 1201 NORTHWEST 65TH PLACE FORT LAUDERDALE, FLORIDA 33309 (800)331–7766	CUPL™ SOFTWARE REV. 1.01 AND LATER

SNAP RESOURCE SUMMARY DESIGNATIONS

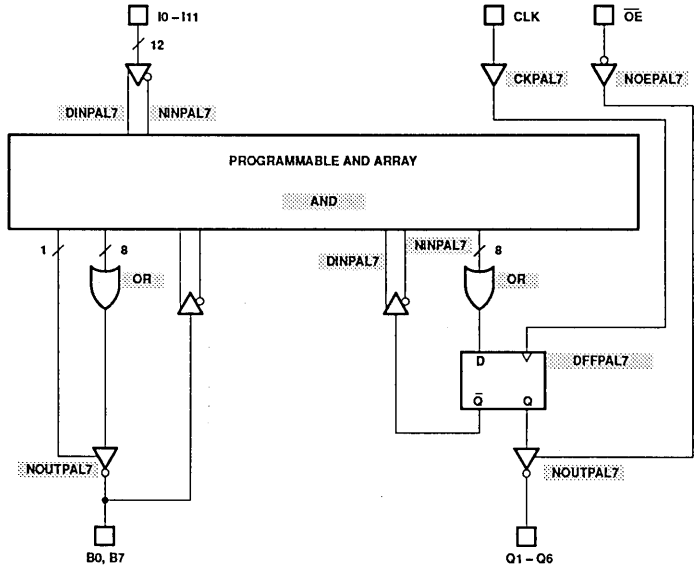


PLUS20L8

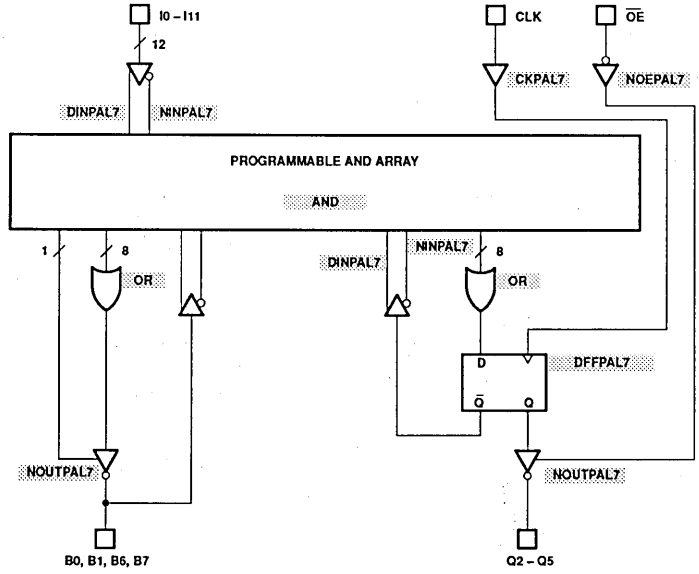


PLUS20R8

SNAP RESOURCE SUMMARY DESIGNATIONS (Continued)



PLUS20R6



PLUS20R4

Document No.	
ECN No.	
Date of Issue	January 1990
Status	Preliminary Specification
Programmable Logic Devices	

PHD48N22–7

Programmable high-speed decoder logic (48 × 73 × 22)

DESCRIPTION

The PHD48N22–7 is an ultra fast Programmable High-speed Decoder featuring a 7.5ns maximum propagation delay. The architecture has been optimized using Philips Components–Signetics state-of-the-art bipolar oxide isolation process coupled with titanium-tungsten fuses to achieve superior speed in any design.

The PHD48N22–7 is a two level logic element comprised of 36 fixed inputs, 73 AND gates, 10 outputs, and 12 bidirectional I/Os. This gives the device the ability to have as many as 48 inputs. Individual 3-State control of all outputs is also provided.

The device is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment. Proprietary designs can be protected by programming the security fuse.

The AMAZE software package from Philips Components–Signetics supports easy design entry for the PHD48N22–7 as well as other PLD devices.

Order codes are listed in the pages following.

FEATURES

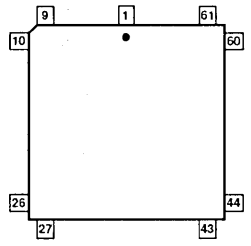
- Ideal for high speed system decoding
- Super high speed at 7.5ns t_{PD}
- 36 dedicated inputs
- 22 outputs
  - 12 bidirectional I/O
  - 10 dedicated outputs
- Security fuse to prevent duplication of proprietary designs.
- Individual 3-State control of all outputs
- Field-programmable on industry standard programmers
- Available in 68-Pin Plastic Leaded Chip Carrier (PLCC)

APPLICATIONS

- High speed memory decoders
- High speed code detectors
- Random logic
- Peripheral selectors
- Machine state decoders

PIN CONFIGURATION

A Package



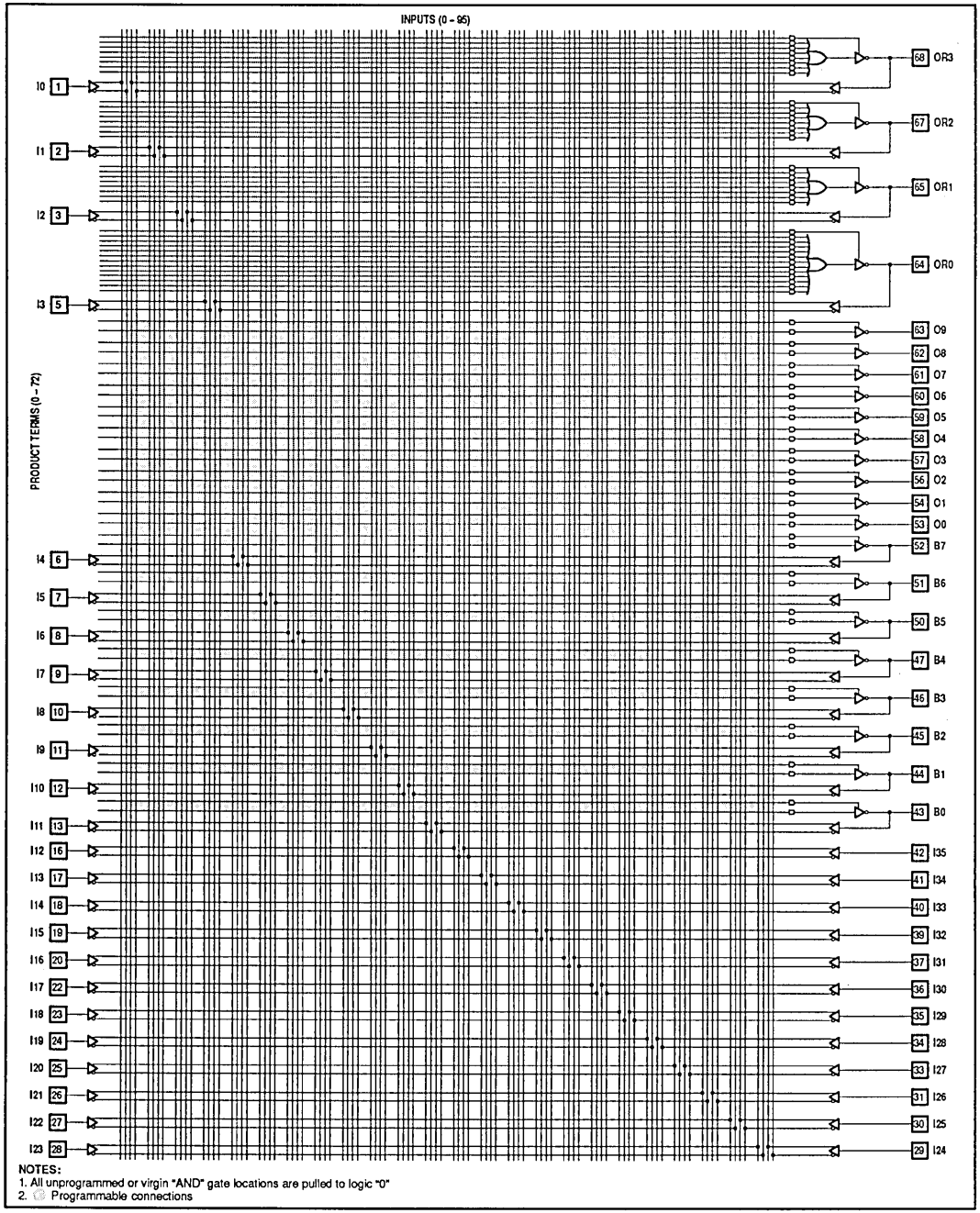
A = Plastic Leaded Chip Carrier

Pin	Function	Pin	Function
1	I0	35	I29
2	I1	36	I30
3	I2	37	I31
4	V_{CC3}	38	V_{CC4}
5	I3	39	I32
6	I4	40	I33
7	I5	41	I34
8	I6	42	I35
9	I7	43	B0
10	I8	44	B1
11	I9	45	B2
12	I10	46	B3
13	I11	47	B4
14	GND5	48	GND6
15	GND1	49	GND2
16	I12	50	B5
17	I13	51	B6
18	I14	52	B7
19	I15	53	O0
20	I16	54	O1
21	V_{CC2}	55	V_{CC1}
22	I17	56	O2
23	I18	57	O3
24	I19	58	O4
25	I20	59	O5
26	I21	60	O6
27	I22	61	O7
28	I23	62	O8
29	I24	63	O9
30	I25	64	OR0
31	I26	65	OR1
32	GND3	66	GND4
33	I27	67	OR2
34	I28	68	OR3

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

LOGIC DIAGRAM

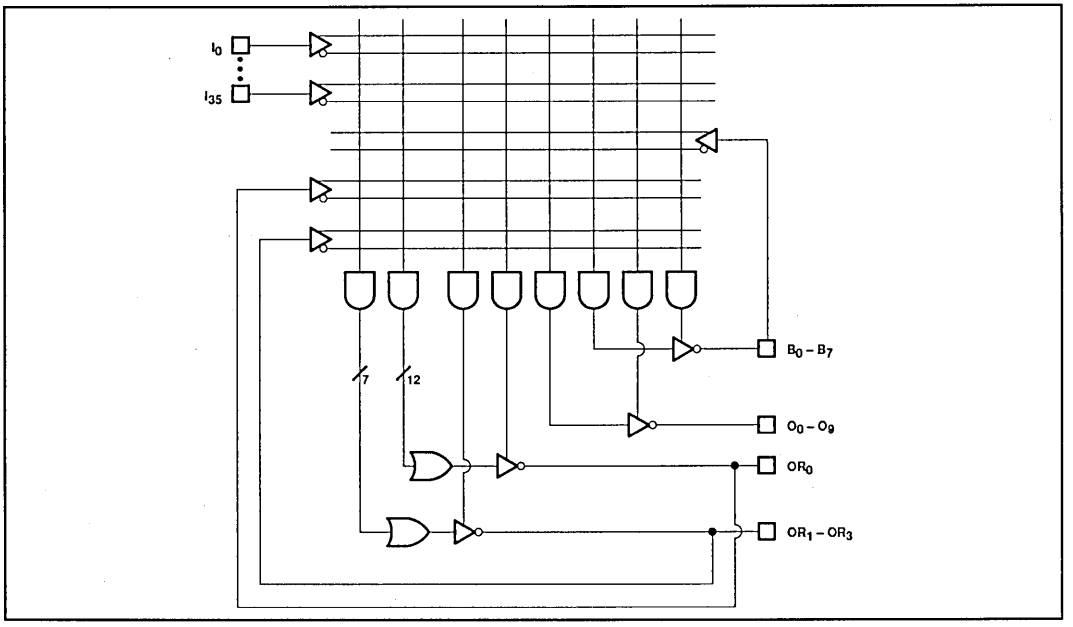


NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ○ Programmable connections

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
68-Pin Plastic Leaded Chip Carrier	PHD48N22–7A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–0.5	+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = +24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = +0.40V		–20	–250	μA
I_{IH}	High	V_{IN} = +2.7V			25	μA
I_I	High	V_{IN} = V_{CC} = $V_{CC\ MAX}$			100	μA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage [3]	V_{OUT} = +2.7V			100	μA
I_{OZL}	Output leakage [3]	V_{OUT} = +0.40V			–100	μA
I_{OS}	Short circuit [4]	V_{OUT} = +0V	–30	–60	–90	mA
I_{CC}	V_{CC} current	V_{CC} = MAX			420	mA
Capacitance[5]						
		V_{CC} = +5V				
C_{IN}	Input	V_{IN} = 2.0V @ f = 1MHz		8		pF
C_{OUT}	I/O	V_{OUT} = 2.0V @ f = 1MHz		8		pF

NOTES:
1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZL}.
4. Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second.
5. These parameters are not 100% tested, but are periodically sampled.

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_A \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 200\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS	LIMITS MIN	LIMITS MIN	UNIT
t_{PD1}[1]	Propagation delay through B/O outputs	(I, B, OR) ±	Output ±	C_L = 50pF		6.5	ns
t_{PD2}[1]	Propagation delay through OR outputs	(I, B, OR) ±	Output ±	C_L = 50pF		7	ns
t_{OE}[2]	Output Enable	(I, B, OR) ±	Output enable	C_L = 50pF		10	ns
t_{OD}[2]	Output Disable	(I, B, OR) ±	Output disable	C_L = 5pF		10	ns

NOTES:

1. $t_{PD1, 2}$ are tested with switch S_1 closed and C_L = 50pF.
2. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

VIRGIN STATE

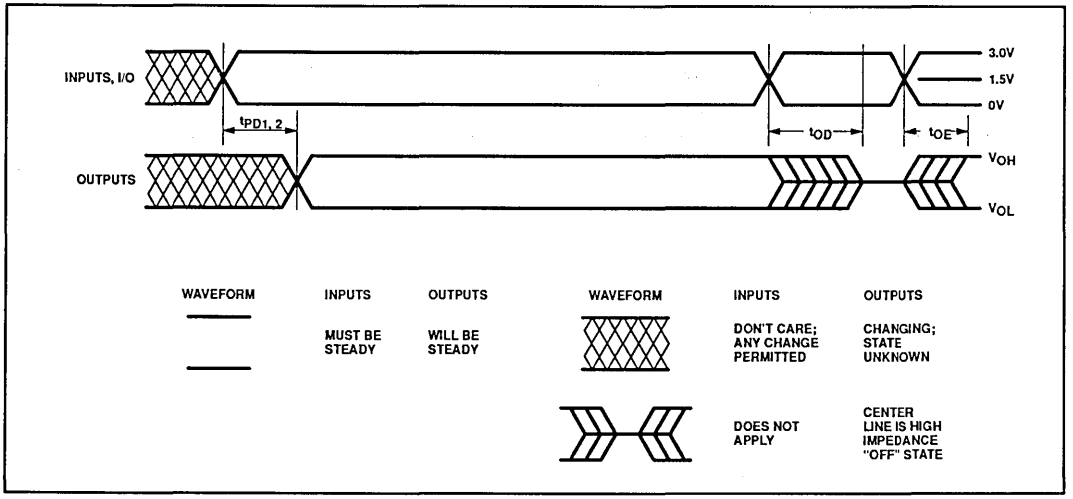
A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are disabled.
2. All p-terms are disabled in the AND array.

TIMING DEFINITIONS

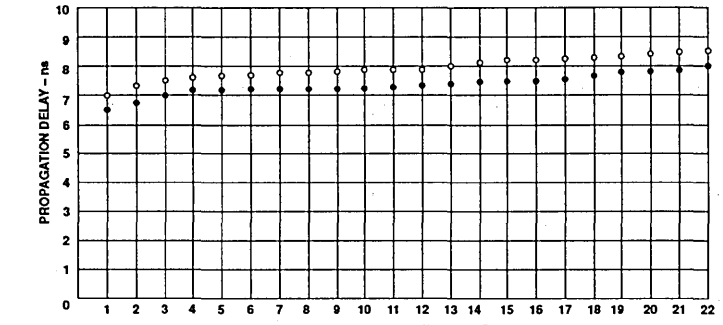
SYMBOL	PARAMETER
t_{PD1}	Input to output propagation delay (through B/O outputs).
t_{PD2}	Input to output propagation delay (through OR outputs).
t_{OD}	Input to Output Disable (3-State) delay (Output Disable).
t_{OE}	Input to Output Enable delay (Output Enable).

TIMING DIAGRAM



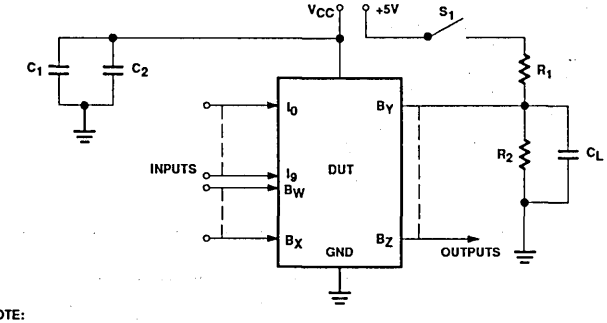
Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7



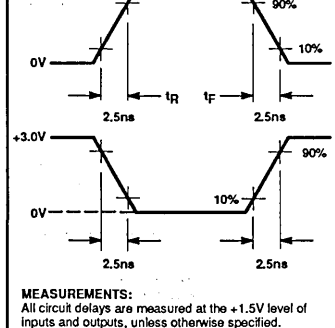
Worst-Case Propagation Delay vs. Number of Outputs Switching

AC TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

LOGIC PROGRAMMING

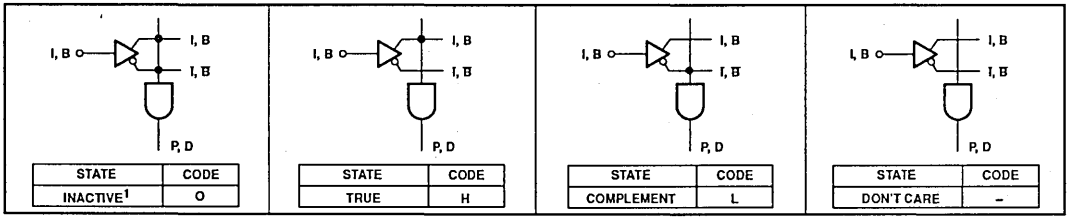
The PHD48N22–7 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ CUPL™ and PALASM® 90 design software packages also support the architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PHD48N22–7 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, each logic variable (I, B, P and D) from the logic equations is assigned a symbol. TRUE (High), COMPLEMENT (Low), DON'T CARE and INACTIVE symbols are defined below.

"AND" ARRAY – (I, B)



STATE	CODE
INACTIVE[1]	O

STATE	CODE
TRUE	H

STATE	CODE
COMPLEMENT	L

STATE	CODE
DON'T CARE	–

NOTE:
1. This is the initial state.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

PROGRAM TABLE

TERM	INPUTS																																														
	I																																			B								OR			
	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0
0																																																
1																																																
2																																																
3																																																
4																																																
5																																																
6																																																
7																																																
8																																																
9																																																
10																																																
11																																																
12																																																
13																																																
14																																																
15																																																
16																																																
17																																																
18																																																
19																																																
20																																																
21																																																
22																																																
23																																																
24																																																
25																																																
26																																																
27																																																
28																																																
29																																																
30																																																
31																																																
32																																																
33																																																
34																																																
35																																																
36																																																
37																																																
38																																																
39																																																
40																																																
41																																																
42																																																
43																																																
44																																																
45																																																
46																																																
47																																																
48																																																
49																																																
50																																																
51																																																
52																																																
53																																																
54																																																
55																																																
56																																																
57																																																
58																																																
59																																																
60																																																
61																																																
62																																																
63																																																
64																																																
65																																																
66																																																
67																																																
68																																																
69																																																
70																																																
71																																																
72																																																
PIN	42	41	40	39	37	36	35	34	33	31	30	29	28	27	26	25	24	23	22	20	19	18	17	16	13	12	11	10	9	8	7	6	5	3	2	1	52	51	50	47	46	45	44	43	68	67	65	64
VARIABLE NAME																																																

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

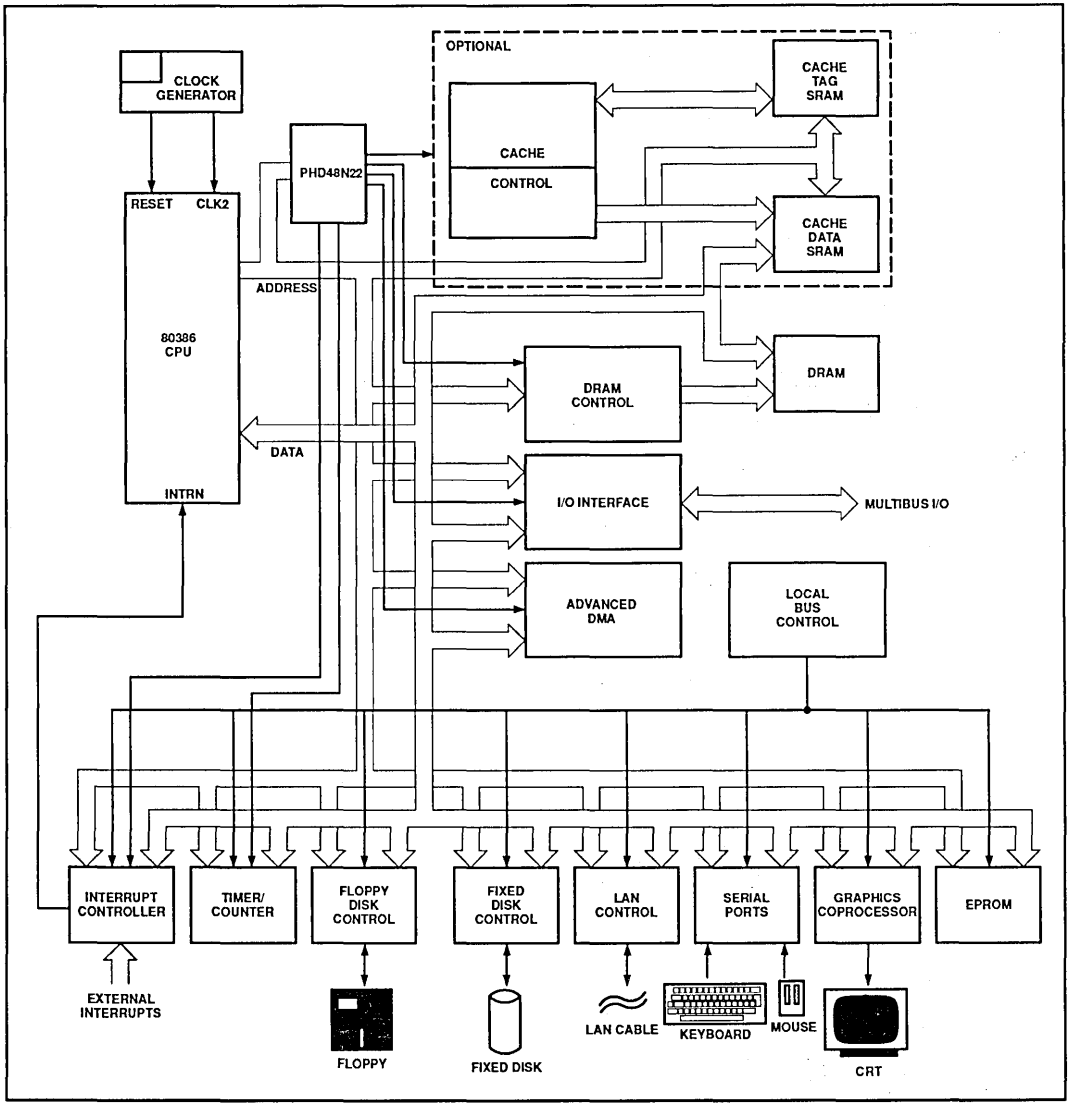
PROGRAM TABLE (Continued)

TERM	OUTPUTS																					
	O									B							OR			
	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0
0																			D			
1																			A			
2																			A			
3																			A			
4																			A			
5																			A			
6																			A			
7																			A			
8																				D		
9																				A		
10																				A		
11																				A		
12																				A		
13																				A		
14																				A		
15																				A		
16																					D	
17																					A	
18																					A	
19																					A	
20																					A	
21																					A	
22																					A	
23																					A	
24																						D
25																						A
26																						A
27																						A
28																						A
29																						A
30																						A
31																						A
32																						A
33																						A
34																						A
35																						A
36																						A
37	D																					
38	A																					
39		D																				
40		A																				
41			D																			
42			A																			
43				D																		
44				A																		
45					D																	
46					A																	
47						D																
48						A																
49							D															
50							A															
51								D														
52								A														
53									D													
54									A													
55										D												
56										A												
57											D											
58											A											
59												D										
60												A										
61													D									
62													A									
63														D								
64														A								
65															D							
66															A							
67																D						
68																A						
69																	D					
70																	A					
71																		D				
72																		A				
PIN	63	62	61	60	59	58	57	56	54	53	52	51	50	47	46	45	44	43	68	67	65	64
VARIABLE NAME																						

Programmable high-speed decoder logic ($48 \times 73 \times 22$)

PHD48N22–7

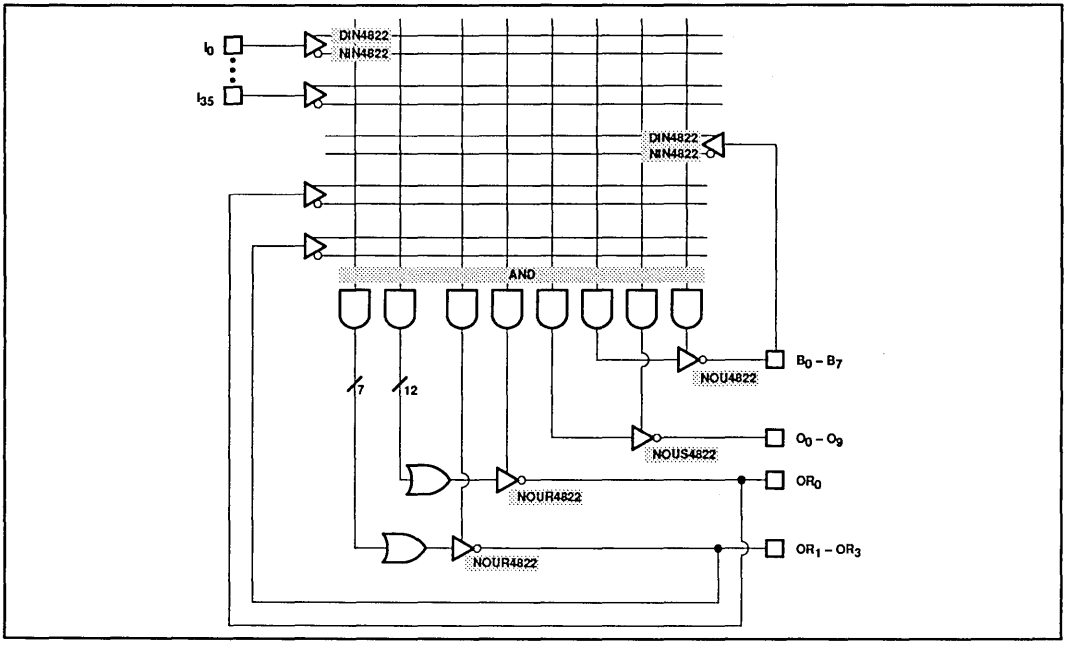
TYPICAL SYSTEM APPLICATION



Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

SNAP RESOURCE SUMMARY DESIGNATIONS



Section 4
Programmable Logic Array Device Data Sheets

INDEX

Document No.	853–0311
ECN No.	00749
Date of Issue	October 18, 1990
Status	Product Specification
Programmable Logic Devices	

PLS153/A

Programmable logic arrays (18 × 42 × 10)

DESCRIPTION

The PLS153 and PLS153A are two-level logic elements, consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

On-chip T/C buffers couple either True (I, B) or Complement ($\overline{I}$, $\overline{B}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS153 and PLS153A are field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are listed in the Ordering Information Table.

FEATURES

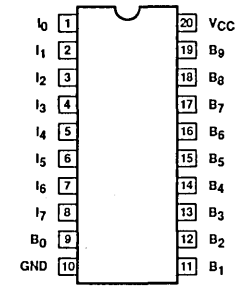
- Field-Programmable (Ni-Cr links)
- 8 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay:
  - PLS153: 40ns (max)
  - PLS153A: 30ns (max)
- Input loading: −100μA (max)
- Power dissipation: 650mA (typ)
- 3-State outputs
- TTL compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

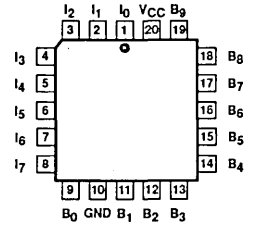
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

LOGIC FUNCTION

TYPICAL PRODUCT TERM:

$Pn = A \cdot B \cdot C \cdot D \cdot \ldots$

TYPICAL LOGIC FUNCTION:

AT OUTPUT POLARITY = H

$Z = P0 + P1 + P2 \ldots$

AT OUTPUT POLARITY = L

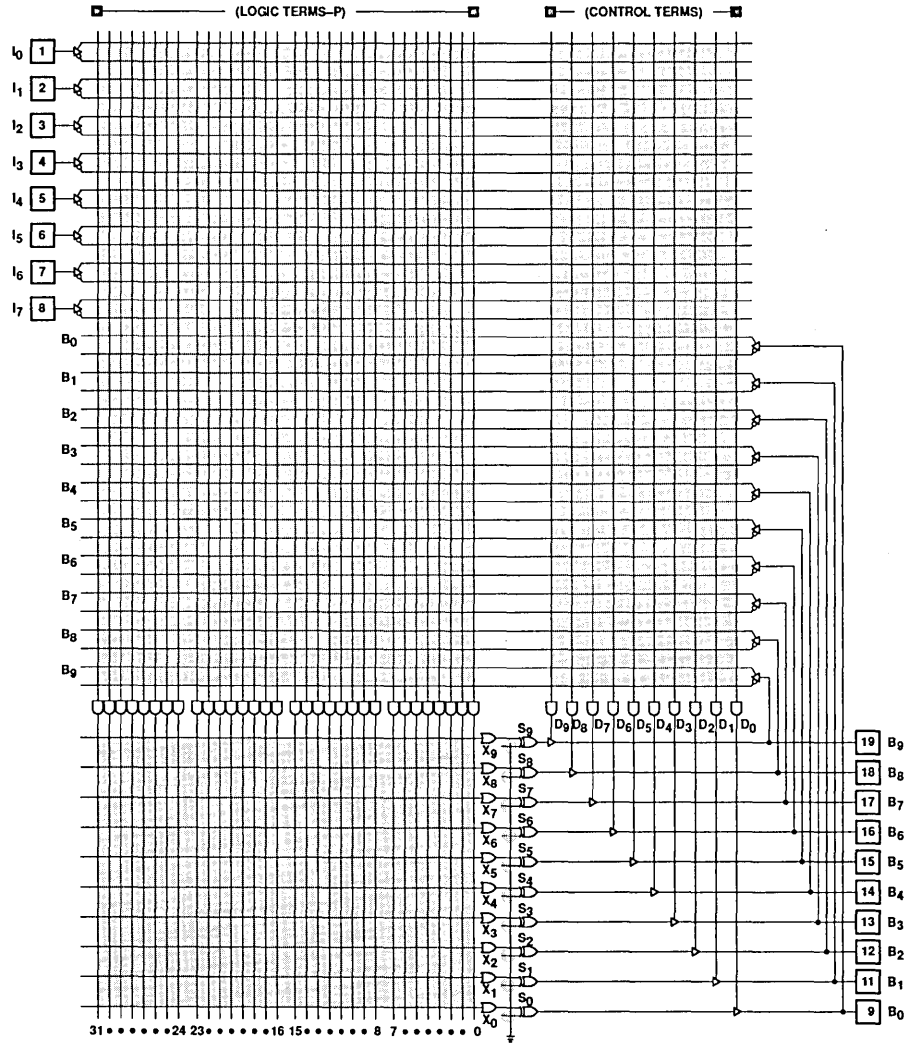
$\overline{Z} = \overline{P0 + P1 + P2 + \ldots}$

$Z = \overline{P0} \cdot \overline{P1} \cdot \overline{P2} \cdot \ldots$

NOTES:
1. For each of the 10 outputs, either function Z (Active-High) or $\overline{Z}$ (Active-Low) is available, but not both. The desired output polarity is programmed via the Ex-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

Programmable logic arrays (18 × 42 × 10) PLS153/A

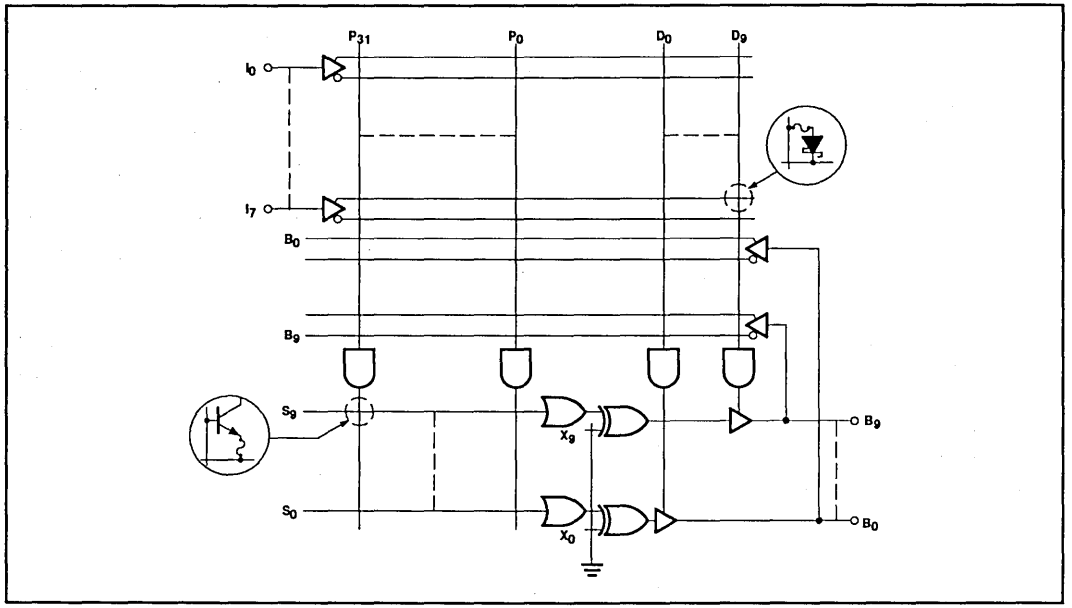
LOGIC DIAGRAM



NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ⊗ Programmable connection.

Programmable logic arrays (18 × 42 × 10) — PLS153/A

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic Dual In-Line, 300mil-wide	PLS153N, PLS153AN
20-Pin Plastic Leaded Chip Carrier	PLS153A, PLS153AA

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS153/A devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Handbook.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = 5.5V			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 5.5V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		130	155	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with +10V applied to I_7.
5. Measured with +10V applied to I_{0-7}. Output sink current is supplied through a resistor to V_{CC}.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with I_0, I_1 at 0V, $I_2 - I_7$ and B_{0-9} at 4.5V.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I_0 - I_7$).

Programmable logic arrays (18 × 42 × 10) — PLS153/A

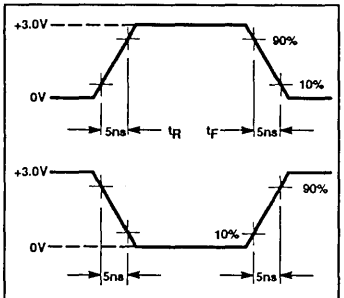
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS						UNIT
					PLS153			PLS153A			
					MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
t_{PD}	Propagation delay	Input ±	Output ±	C_L = 30pF		30	40		20	30	ns
t_{OE}	Output enable[2]	Input ±	Output –	C_L = 30pF		25	35		20	30	ns
t_{OD}	Output disable[2]	Input ±	Output +	C_L = 5pF		25	35		20	30	ns

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays are measured and specified under worst case conditions.

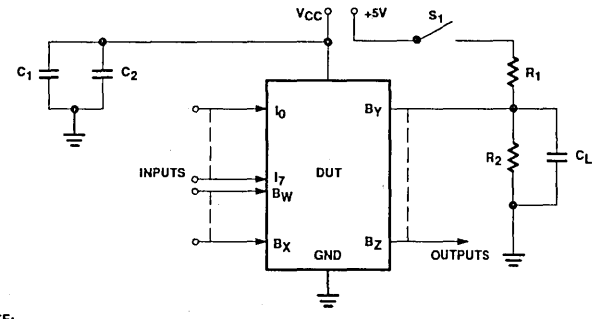
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

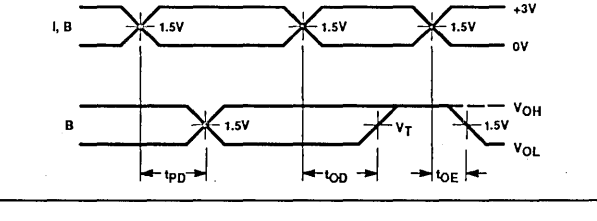


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic arrays (18 × 42 × 10) — PLS153/A

LOGIC PROGRAMMING

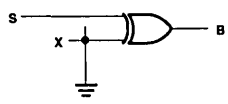
The PLS153/A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, Data I/O's ABEL™ and Logical Devices, Inc. CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

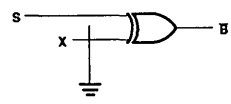
PLS153/A logic designs can also be generated using the program table entry format detailed on the following page. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

OUTPUT POLARITY – (B)

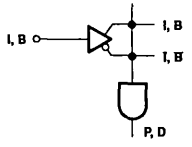


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

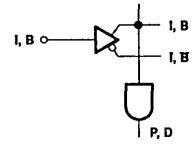


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

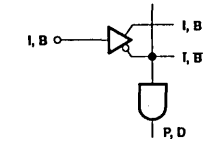
AND ARRAY – (I, B)



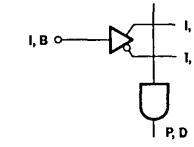
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

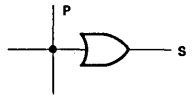


STATE	CODE
I, B	L

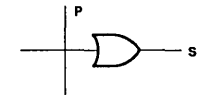


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate P_n will be unconditioanlly inhibited if both the True and Complement of an input (either I or B) are left intact.

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

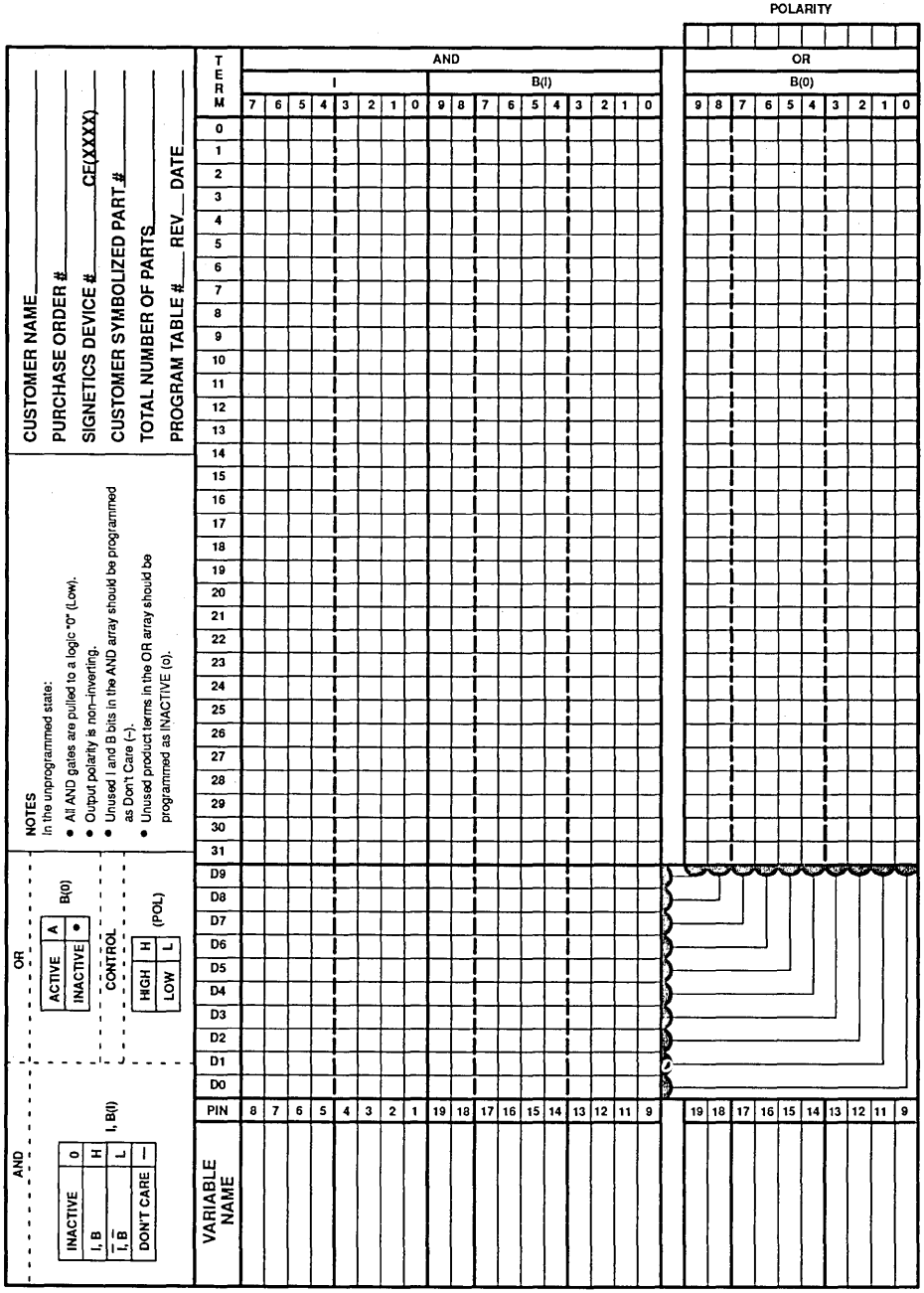
CAUTION: PLS153A TEST COLUMNS

The PLS153A incorporates two columns not shown in the logic block diagram. These columns are used for in-house testing of the device in the unprogrammed state. These columns must be disabled prior to using the PLS153A in your application. If you are using a Signetics-approved programmer, the disabling is accomplished during the device programming sequence. If these columns are not disabled, abnormal operation is possible.

Furthermore, because of these test columns, the PLS153A cannot be programmed using the programmer algorithm for the PLS153.

Programmable logic arrays (18 × 42 × 10) PLS153/A

PROGRAM TABLE



Philips Components–Signetics

Document No.	853–1285
ECN No.	00751
Date of Issue	October 18, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS153B/D

Programmable logic arrays (18 × 42 × 10)

DESCRIPTION

The PLUS153 PLDs are high speed, combinatorial Programmable Logic Arrays. The Signetics state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce propagation delays as short as 12ns.

The 20-pin PLUS153 devices have a programmable AND array and a programmable OR array. Unlike PAL® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS153 devices can support up to 32 input wide OR functions.

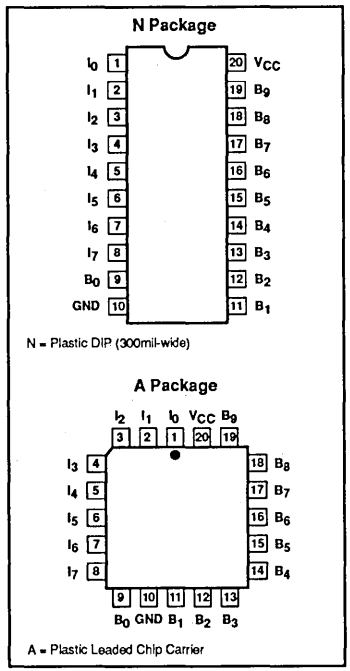
The polarity of each output is user-programmable as either active-High or active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS153 devices are user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

- I/O propagation delays (worst case)
  - PLUS153B – 15ns max.
  - PLUS153D – 12ns max.
- Functional superset of 16L8 and most other 20-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 8 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 750mW (typ.)
- TTL Compatible

PIN CONFIGURATIONS



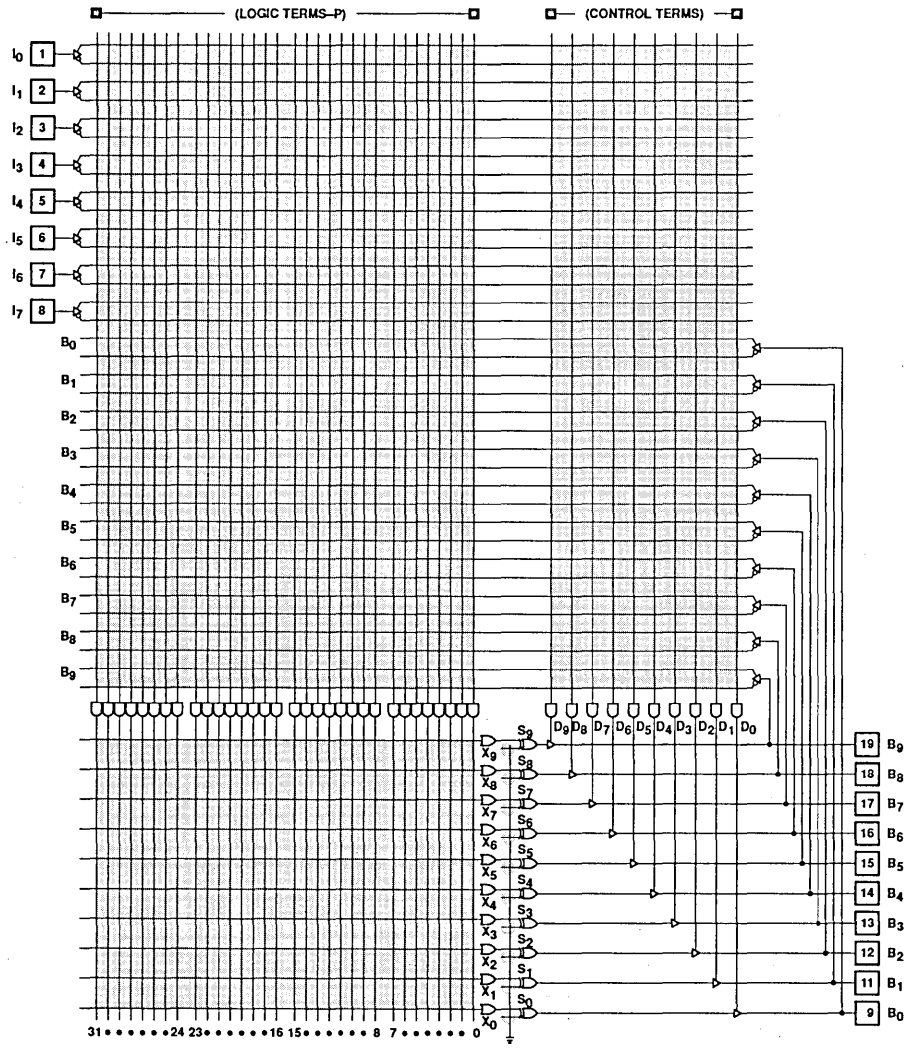
APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices Corporation.

Programmable logic arrays (18 × 42× 10) PLUS153B/D

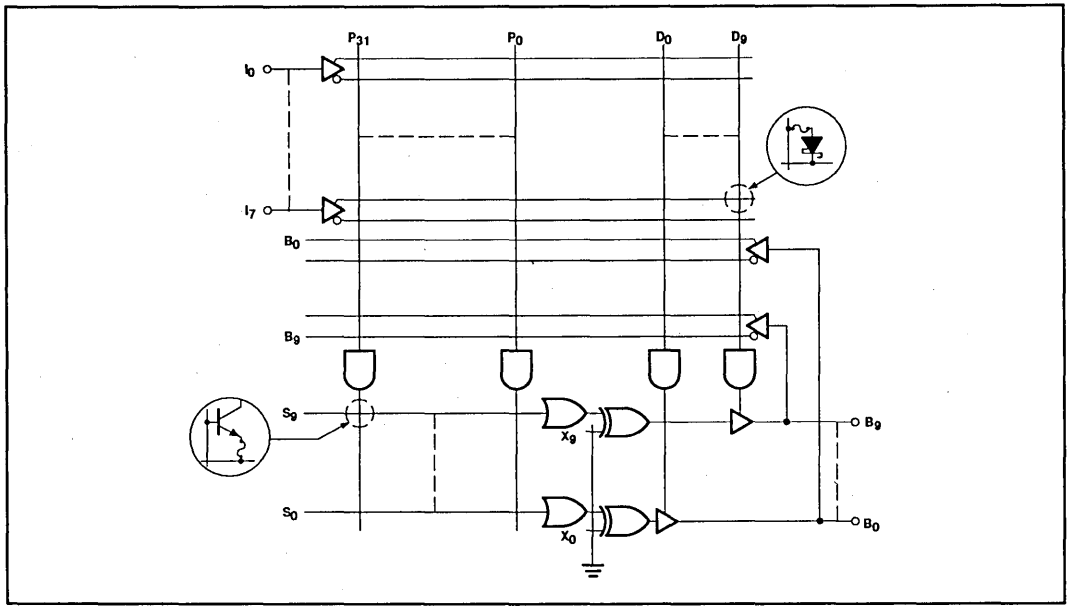
LOGIC DIAGRAM



NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ◌ Programmable connection.

Programmable logic arrays (18 × 42× 10) PLUS153B/D

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	t_{PD} (MAX)	ORDER CODE
20-Pin Plastic Dual-In-Line 300mil-wide	15ns	PLUS153BN
20-Pin Plastic Dual-In-Line 300mil-wide	12ns	PLUS153DN
20-Pin Plastic Leaded Chip Carrier	15ns	PLUS153BA
20-Pin Plastic Leaded Chip Carrier	12ns	PLUS153DA

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING MIN	RATING MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	200	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs $I_0 - I_2$ = 0V, inputs $I_3 - I_5$ = 4.5V, inputs I_7 = 4.5V and I_6 = 10V. For outputs $B_0 - B_4$ and for outputs $B_5 - B_9$ apply the same conditions except I_7 = 0V.
5. Same conditions as Note 4 except I_7 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs $I_0 - I_7$ and $B_0 - B_9$ = 0V.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I_0 - I_7$).

Programmable logic arrays (18 × 42× 10) PLUS153B/D

AC ELECTRICAL CHARACTERISTICS

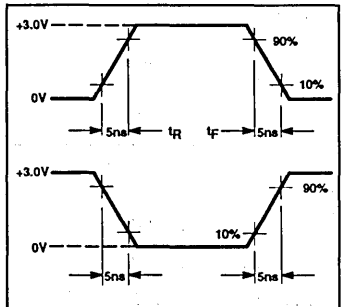
0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, R_1 = 300Ω, R_2 = 390Ω

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS						UNIT
					PLUS153B			PLUS153D			
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		11	15		10	12	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		11	15		10	12	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		11	15		10	12	ns

NOTES:

1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

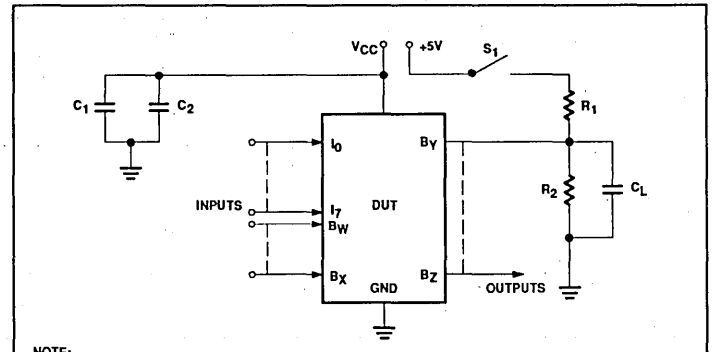
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

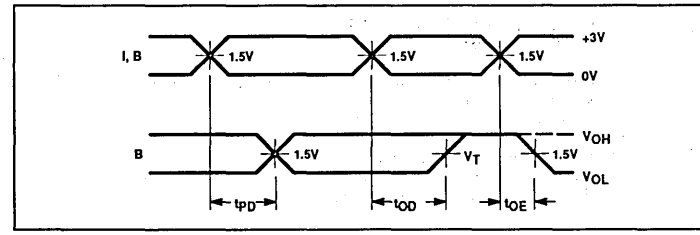


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic arrays (18 × 42× 10) — PLUS153B/D

LOGIC PROGRAMMING

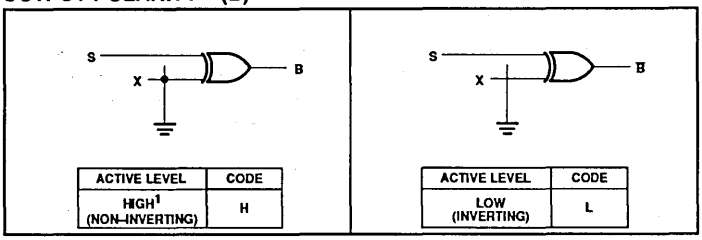
The PLUS153B/D is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS153B/D architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS153B/D logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

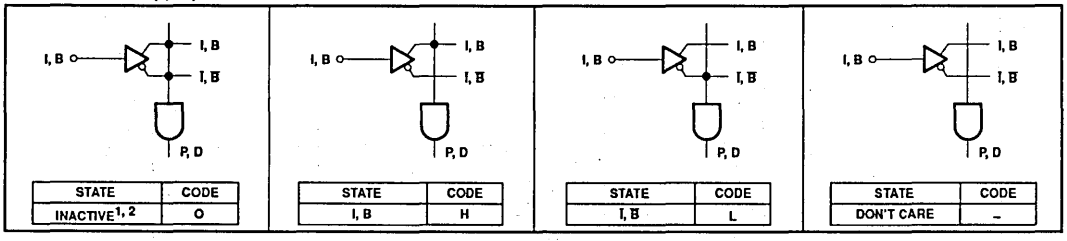
OUTPUT POLARITY – (B)



ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

ACTIVE LEVEL	CODE
LOW (INVERTING)	L

AND ARRAY – (I, B)



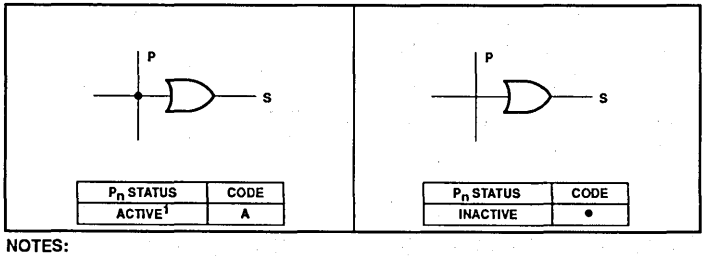
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
I, B̄	L

STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A

P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate P_n will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

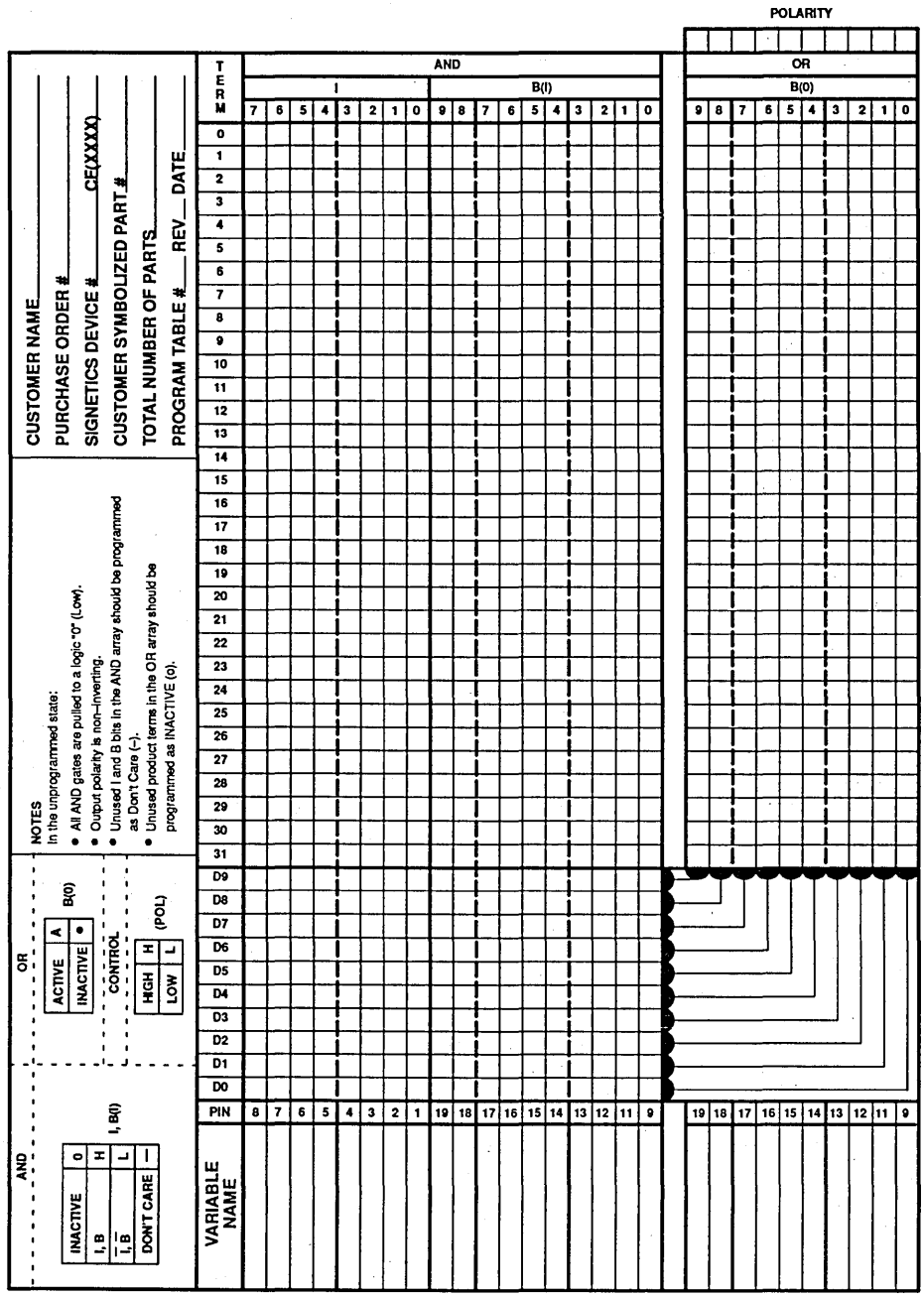
VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:
1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

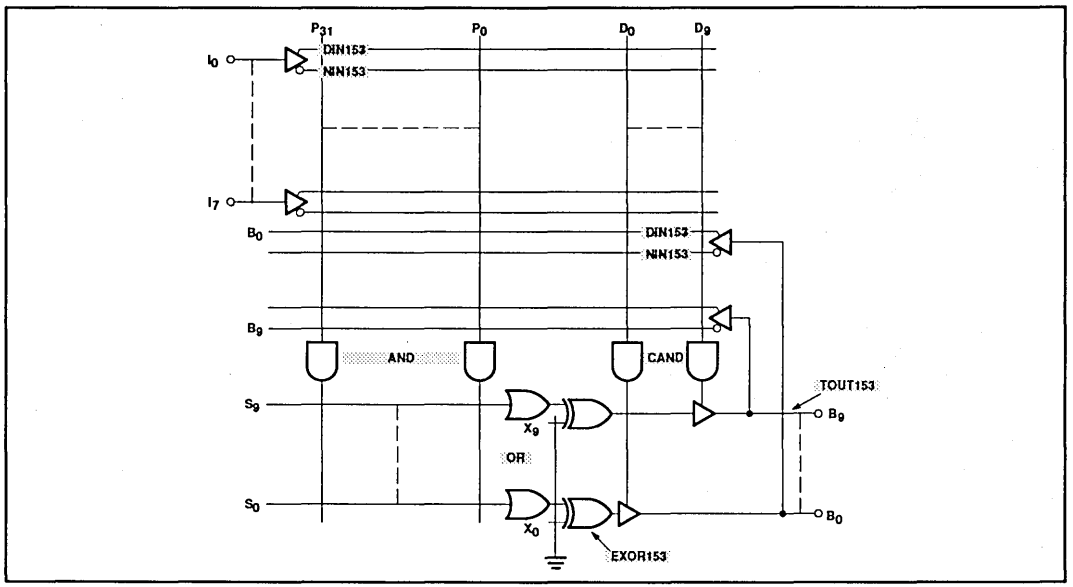
ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic arrays (18 × 42× 10) PLUS153B/D

PROGRAM TABLE



SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–1508
ECN No.	00750
Date of Issue	October 18, 1990
Status	Product Status
Programmable Logic Devices	

PLUS153–10

Programmable logic array

(18 × 42 × 10)

DESCRIPTION

The PLUS153–10 PLD is a high speed, combinatorial Programmable Logic Array. The Signetics state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce maximum propagation delays of 10ns or less.

The 20-pin PLUS153 device has a programmable AND array and a programmable OR array. Unlike PAL® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS153–10 can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either Active-High or Active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS153–10 device is user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

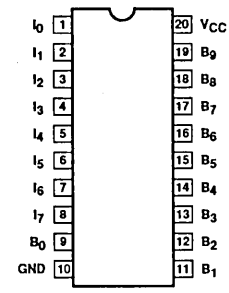
- I/O propagation delays (worst case)
  - PLUS153–10 – 10ns max.
- Functional superset of 16L8 and most other 20-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 8 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 750mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

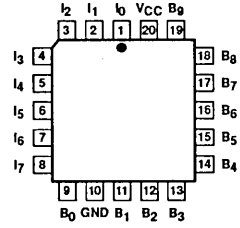
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package

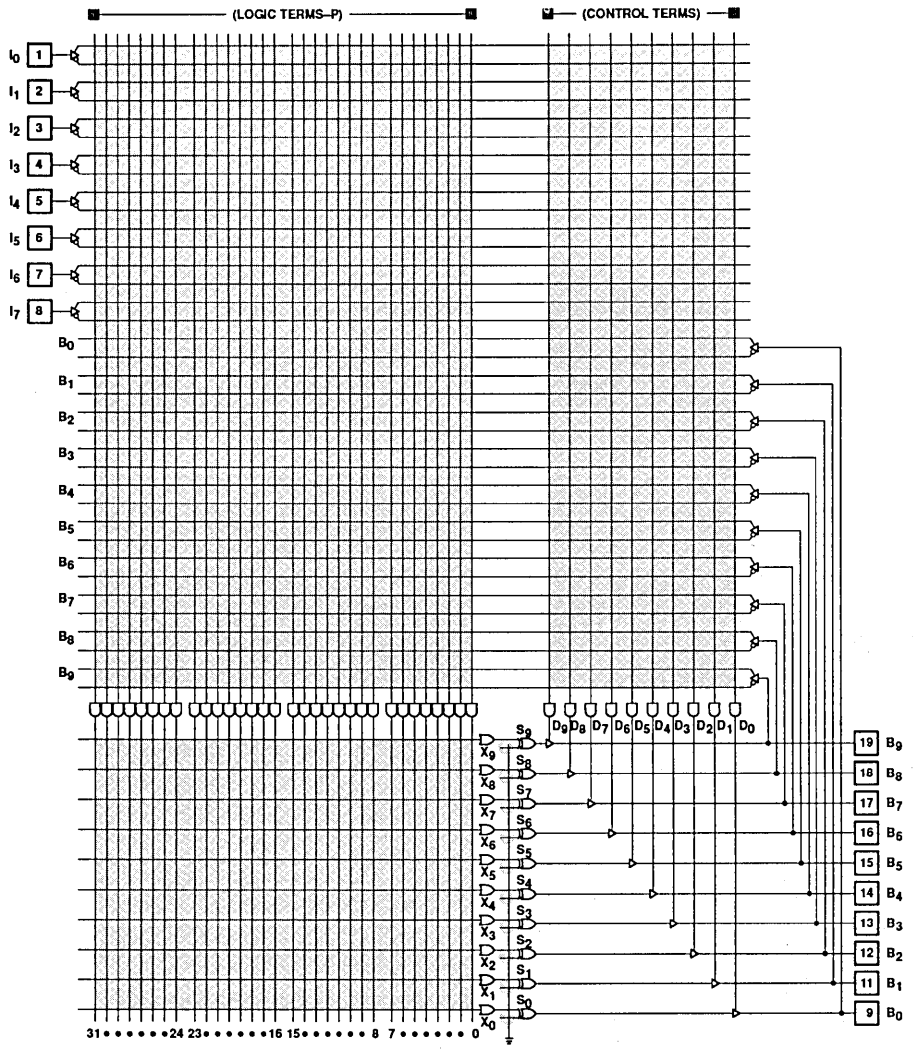


A = Plastic Leaded Chip Carrier

®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices Corporation.

Programmable logic array (18 × 42 × 10) | PLUS153–10

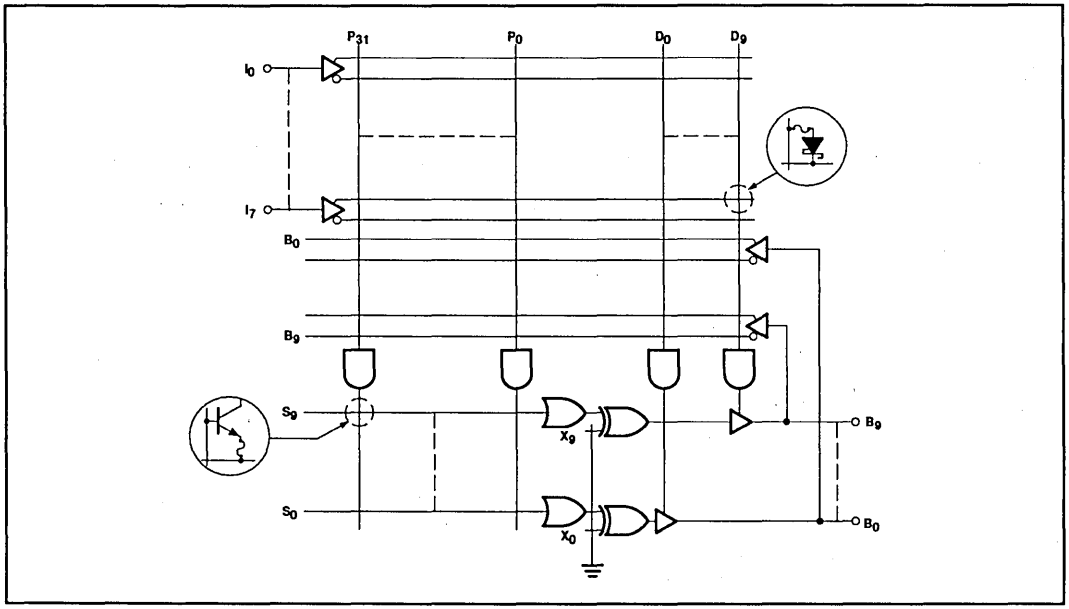
LOGIC DIAGRAM



NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. Programmable connection.

Programmable logic array (18 × 42 × 10) PLUS153–10

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	t_{PD} (MAX)	ORDER CODE
20-Pin Plastic Dual-In-Line 300mil-wide	10ns	PLUS153–10N
20-Pin Plastic Leaded Chip Carrier	10ns	PLUS153–10A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING MIN	RATING MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{in}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

Programmable logic array (18 × 42 × 10) PLUS153–10

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	200	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs $I_0 - I_2$ = 0V, inputs $I_3 - I_5$ = 4.5V, inputs I_7 = 4.5V and I_6 = 10V. For outputs $B_0 - B_4$ and for outputs $B_5 - B_9$ apply the same conditions except I_7 = 0V.
5. Same conditions as Note 4 except I_7 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs $I_0 - I_7$ and $B_0 - B_9$ = 0V.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I_0 - I_7$).

Programmable logic array (18 × 42 × 10) PLUS153–10

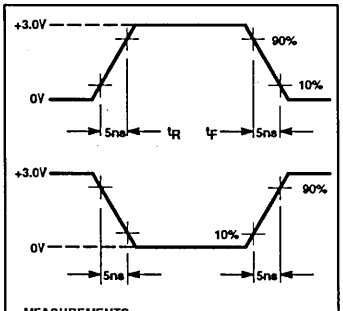
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		8	10	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		8	10	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		8	10	ns

NOTES:
1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

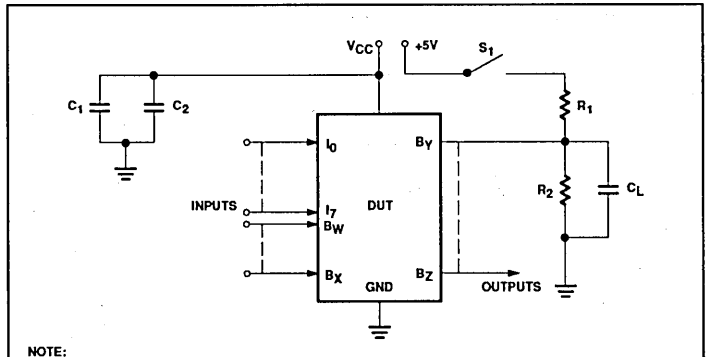
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

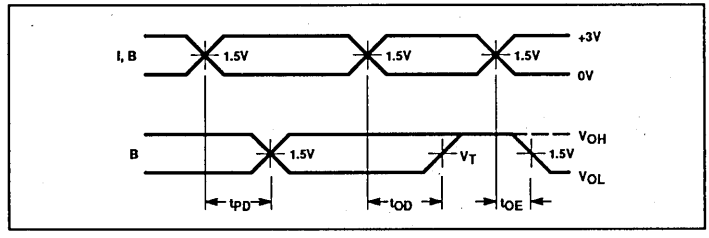


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic array (18 × 42 × 10) PLUS153–10

LOGIC PROGRAMMING

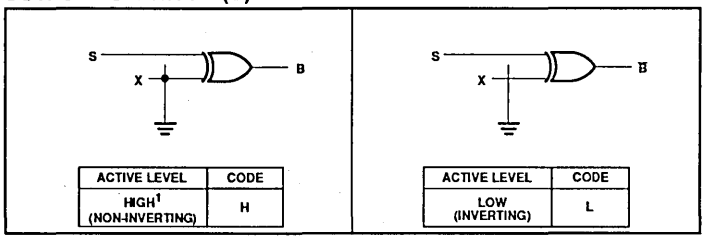
The PLUS153–10 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS153–10 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS153–10 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

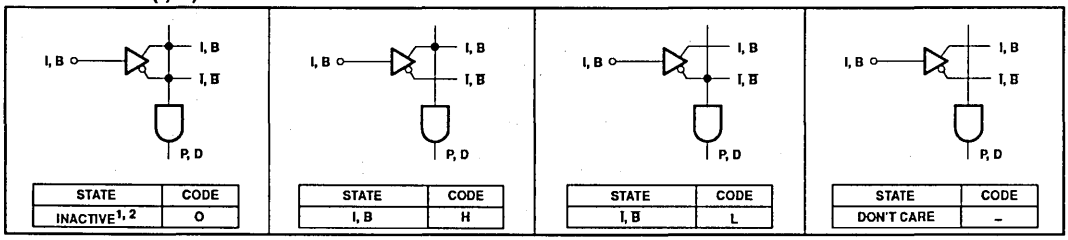
OUTPUT POLARITY – (B)



ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

ACTIVE LEVEL	CODE
LOW (INVERTING)	L

AND ARRAY – (I, B)



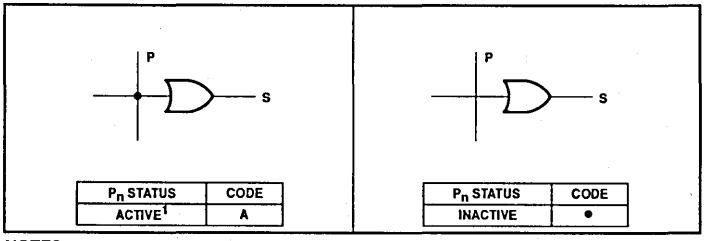
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
I, B̄	L

STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A

P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate P_n will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

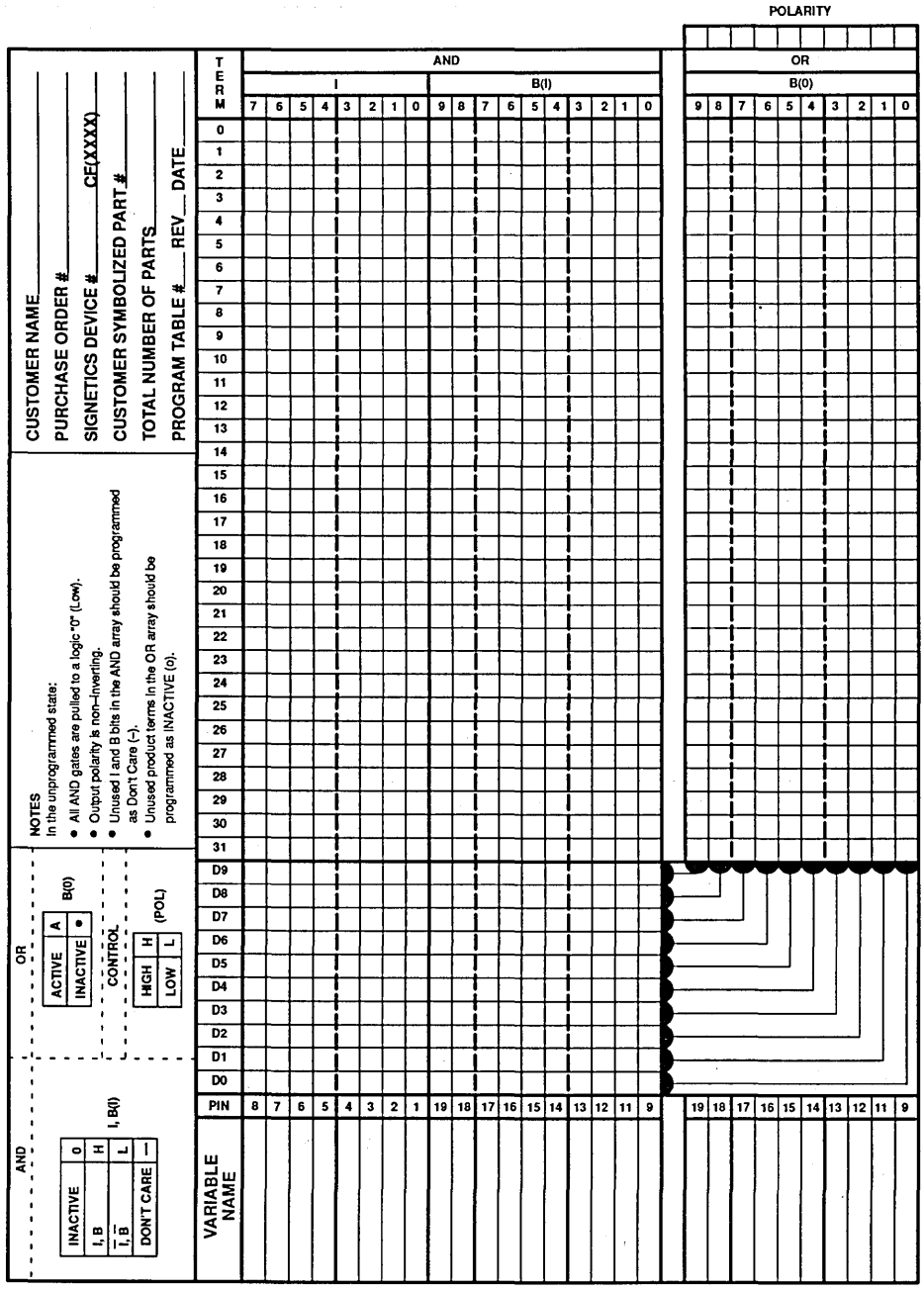
VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:
1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

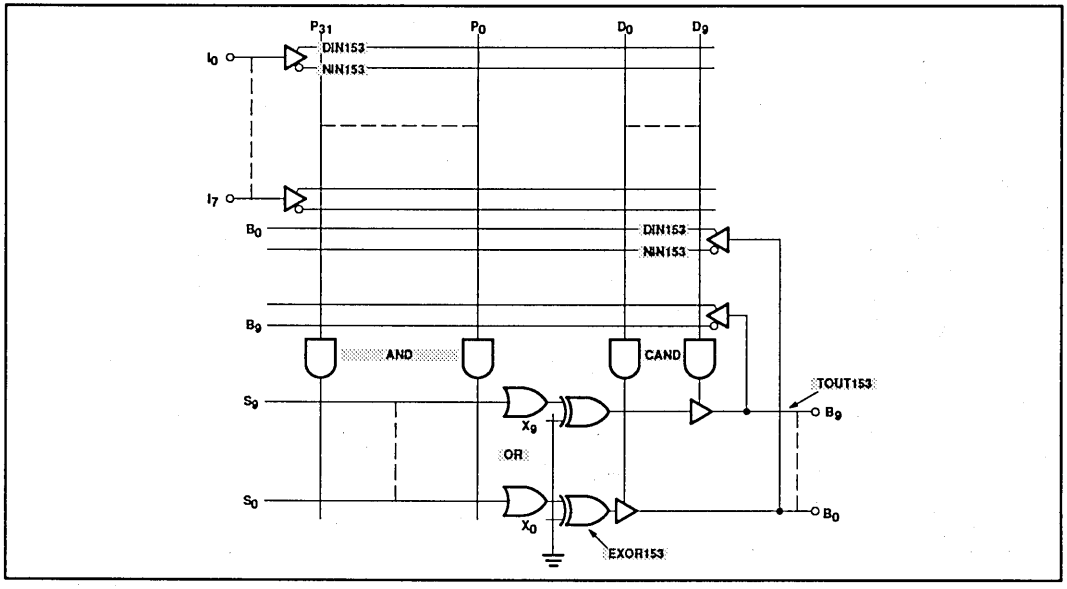
Programmable logic array (18 × 42 × 10) PLUS153–10

PROGRAM TABLE



Programmable logic array (18 × 42 × 10) PLUS153–10

SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–0324
ECN No.	00752
Date of Issue	October 18, 1990
Status	Product Specification
Programmable Logic Devices	

PLS173

Programmable logic array (22 × 42 × 10)

DESCRIPTION

The PLS173 is a two-level logic element consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 12 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 22 inputs to 10 outputs.

On-chip T/C buffers couple either True (I, B) or Complement ($\bar{I}$, $\bar{B}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS173 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes for this device are contained in the pages following.

FEATURES

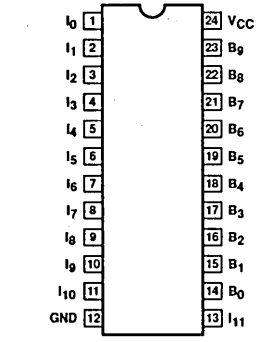
- I/O propagation delay: 30ns (max.)
- 12 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- Ni-Cr programmable links
- Input loading: –100μA (max.)
- Power dissipation: 750mW (typ.)
- 3-State outputs
- TTL compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

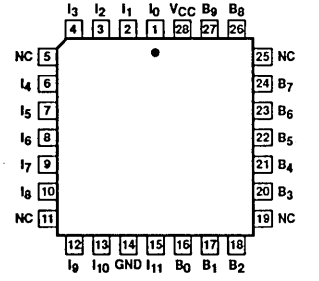
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

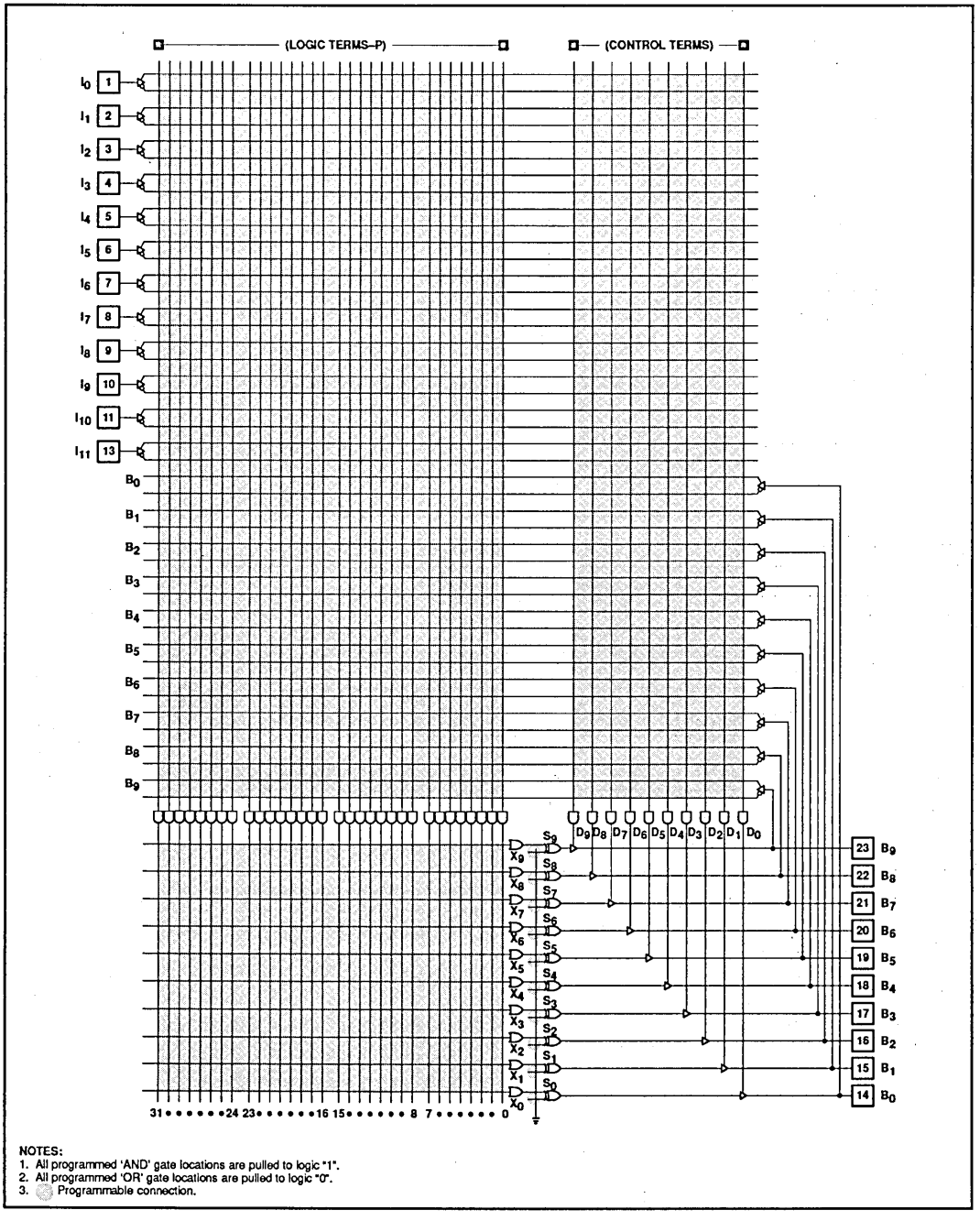
A Package



A = Plastic Leaded Chip Carrier

Programmable logic array (22 × 42 × 10) PLS173

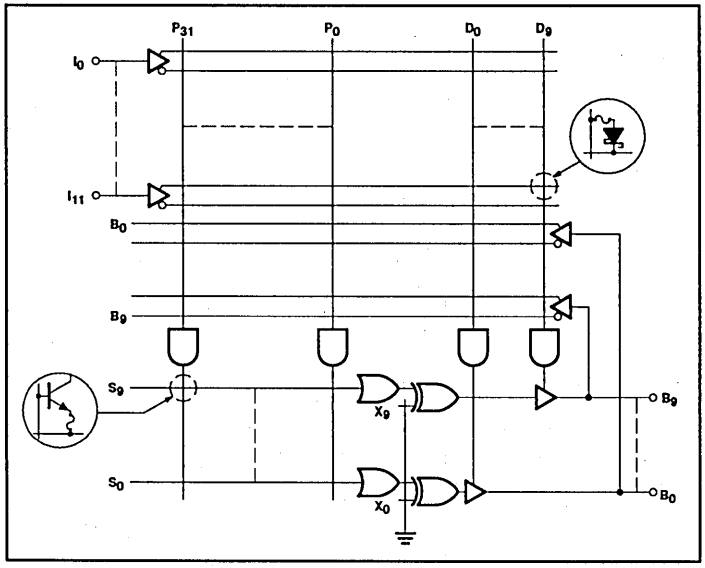
LOGIC DIAGRAM



NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ◌ Programmable connection.

Programmable logic array (22 × 42 × 10) — PLS173

FUNCTIONAL DIAGRAM



LOGIC FUNCTION

TYPICAL PRODUCT TERM:
$Pn = A \cdot \overline{B} \cdot C \cdot D \cdot \ldots$

TYPICAL LOGIC FUNCTION:
AT OUTPUT POLARITY = H
$Z = P0 + P1 + P2 \ldots$

AT OUTPUT POLARITY = L
$Z = \overline{P0 + P1 + P2 + \ldots}$
$Z = \overline{P0} \cdot \overline{P1} \cdot \overline{P2} \cdot \ldots$

NOTES:
1. For each of the 10 outputs, either function Z (Active-High) or Z (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. ZX, A, B, C, etc. are user defined connections to fixed inputs (I), and bidirectional pins (B).

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic Dual-In-Line 300mil-wide	PLS173N
28-Pin Plastic Leaded Chip Carrier	PLS173A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING Min	RATING Max	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS173 is also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Signetics Military Data Handbook.

Programmable logic array (22 × 42 × 10) PLS173

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 5.5V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	170	mA
Capacitance						
		V_{CC} = 5V				
I_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs V_{IL} applied to I_{11}. Pins 1–5 = 0V, Pins 6–10 = 4.5V, Pin 11 = 0V and Pin 13 = 10V.
5. Same conditions as Note 4 except Pin 11 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with I_0 and I_1 = 0V, and $I_2 - I_{11}$ and $B_0 - B_9$ = 4.5V. Part in Virgin State.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I_0 - I_{11}$).

Programmable logic array (22 × 42 × 10) PLS173

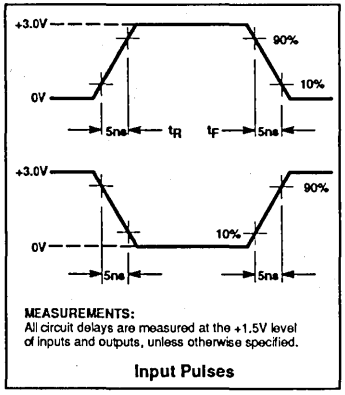
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
t_{PD}	Propagation delay[2]	Input ±	Output ±	C_L = 30pF		20	30	ns
t_{OE}	Output enable[1]	Input ±	Output –	C_L = 30pF		20	30	ns
t_{OD}	Output disable[1]	Input ±	Output +	C_L = 5pF		20	30	ns

NOTES:
1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

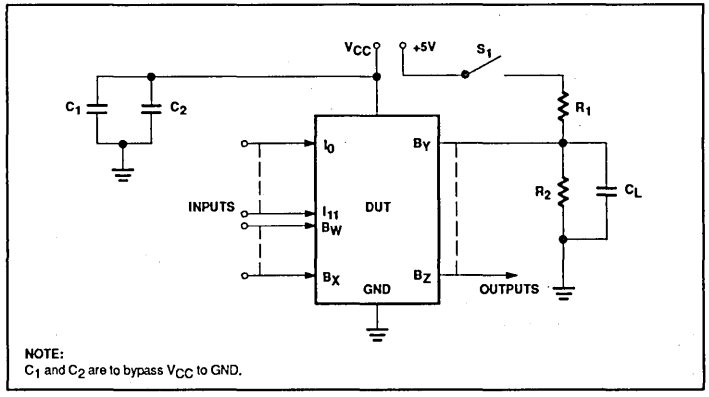
VOLTAGE WAVEFORM



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

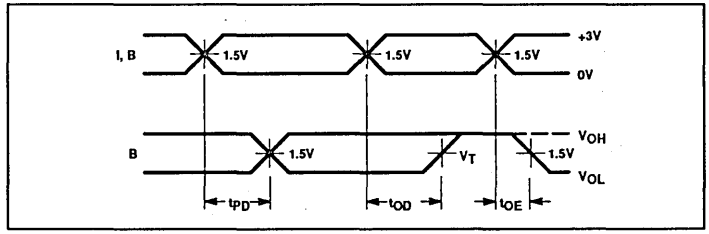


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic array (22 × 42 × 10) PLS173

LOGIC PROGRAMMING

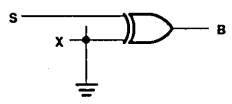
The PLS173 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, Data I/O Corporation's ABEL™, and Logical Devices Incorporated's CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

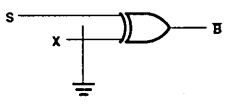
PLS173 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

OUTPUT POLARITY – (B)

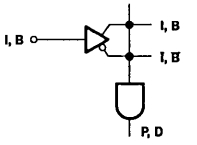


ACTIVE LEVEL	CODE
HIGH[1] (NON-NVERTING)	H

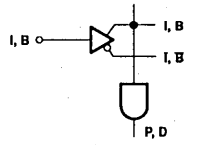


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

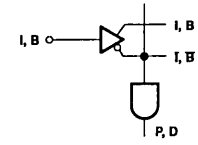
AND ARRAY – (I, B)



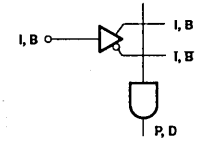
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

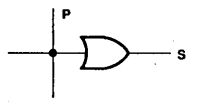


STATE	CODE
I, B̄	L

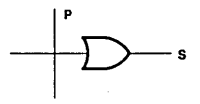


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P_n, D_n.
2. Any gate P_n, D_n will be unconditionally inhibited if both the True and Complement of any input (I, B) are left intact.

VIRGIN STATE

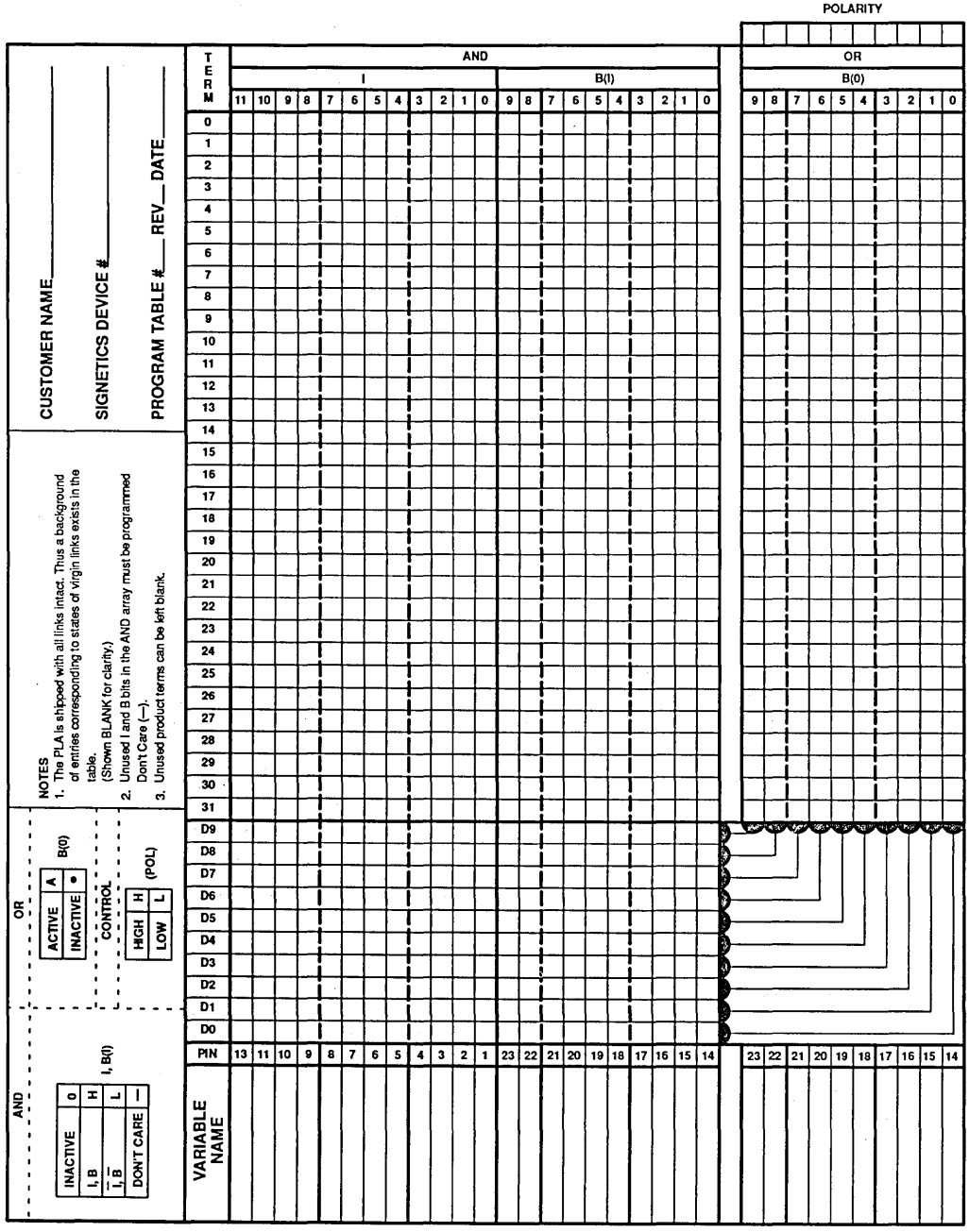
A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic array (22 × 42 × 10) PLS173

PROGRAM TABLE



Document No.	853–1298
ECN No.	00472
Date of Issue	September 20, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS173B/D

Programmable logic arrays (22 × 42 × 10)

DESCRIPTION

The PLUS173 PLDs are high speed, combinatorial Programmable Logic Arrays. The Signetics state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce propagation delays as short as 12ns.

The 24-pin PLUS173 devices have a programmable AND array and a programmable OR array. Unlike PAL ® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS173 devices can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either Active-High or Active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS173 devices are user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

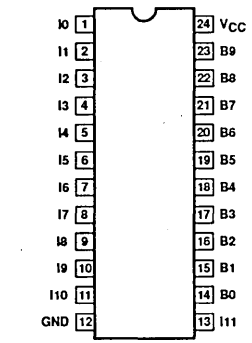
- I/O propagation delays (worst case)
  - PLUS173B – 15ns max.
  - PLUS173D – 12ns max.
- Functional superset of 20L10 and most other 24-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 12 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 750mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

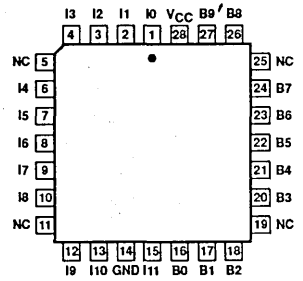
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



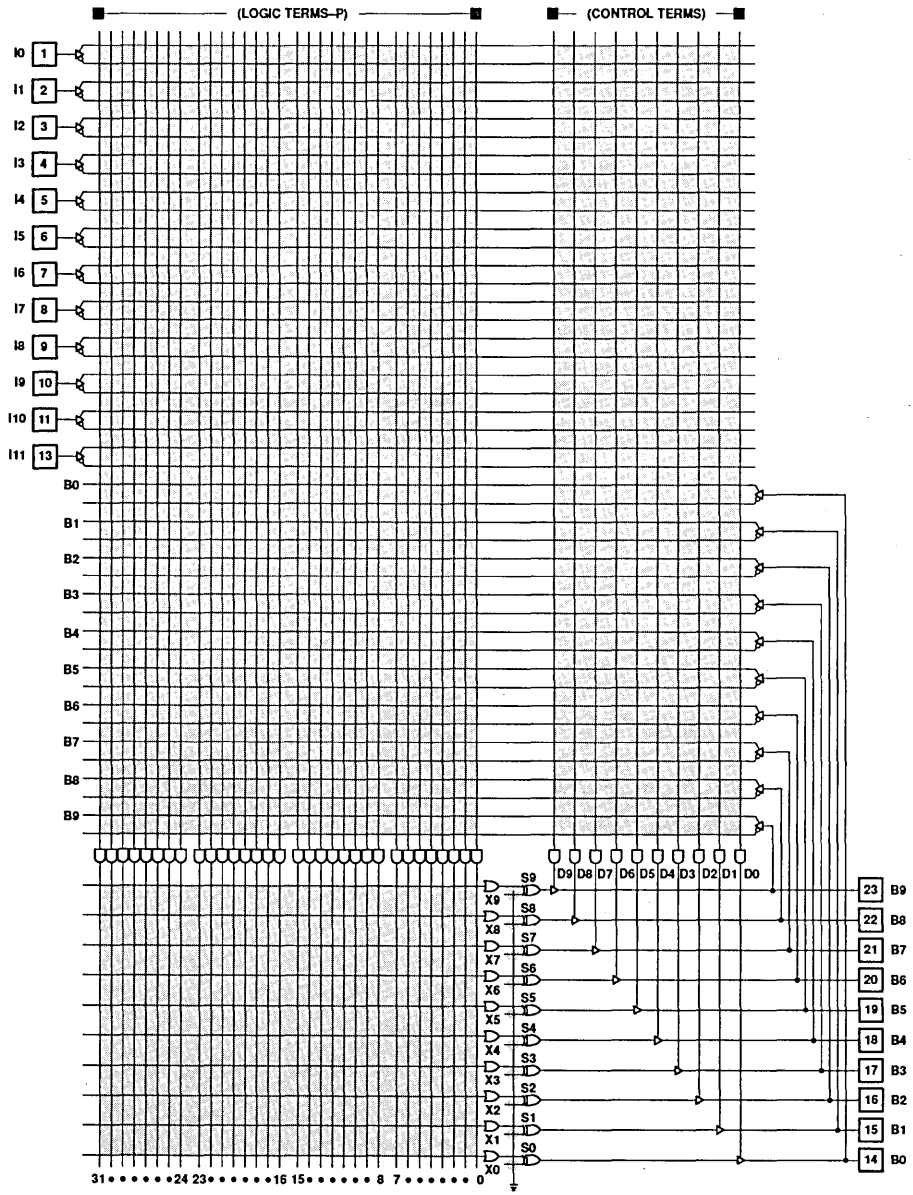
A = Plastic Leaded Chip Carrier

®PAL is a registered trademark of Advanced Micro Devices Corporation.

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

LOGIC DIAGRAM

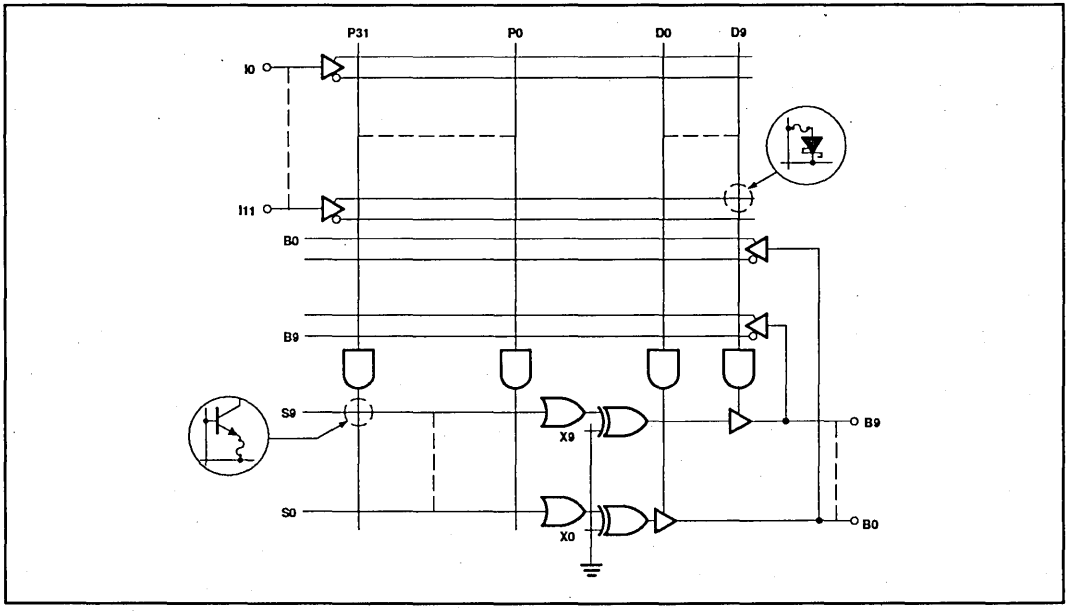


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	t_{PD} (MAX)	ORDER CODE
24-Pin Plastic DIP 300mil-wide	15ns	PLUS173BN
24-Pin Plastic DIP 300mil-wide	12ns	PLUS173DN
28-Pin Plastic Leaded Chip Carrier	15ns	PLUS173BA
28-Pin Plastic Leaded Chip Carrier	12ns	PLUS173DA

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING MIN	RATING MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{In}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	200	mA
Capacitance						
		V_{CC} = 5V				
I_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs I0 – I4 = 0V, inputs I5 – I9 = 4.5V, I11 = 4.5V and I19 = 10V. For outputs B0 – B4 and for outputs B5 – B9 apply the same conditions except I11 = 0V.
5. Same conditions as Note 4 except input I11 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs I0 – I11 and B0 – B9 = 0V. Part in Virgin State.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only (I0 – I11).

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

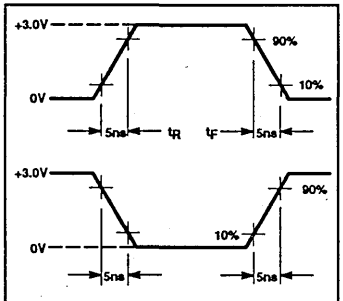
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS						UNIT
					PLUS173B			PLUS173D			
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		11	15		10	12	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		11	15		10	12	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		11	15		10	12	ns

NOTES:
1. For 3-State outputs; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

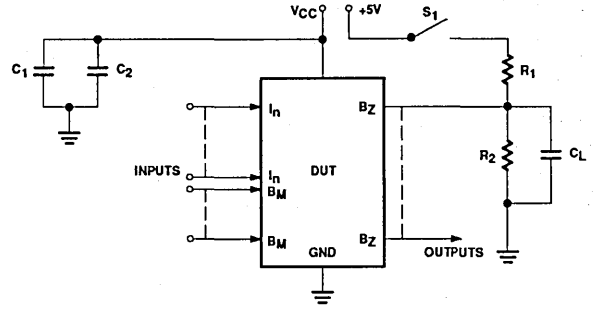
VOLTAGE WAVEFORM



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT



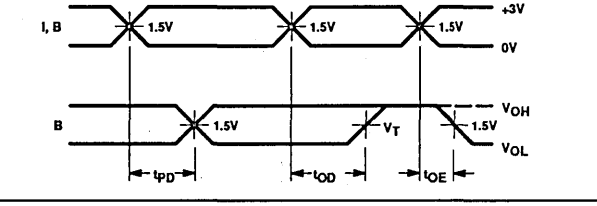
NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Test Load Circuit

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

LOGIC PROGRAMMING

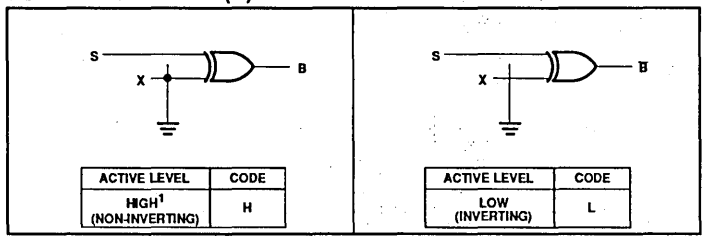
The PLUS173 series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS173 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS173 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

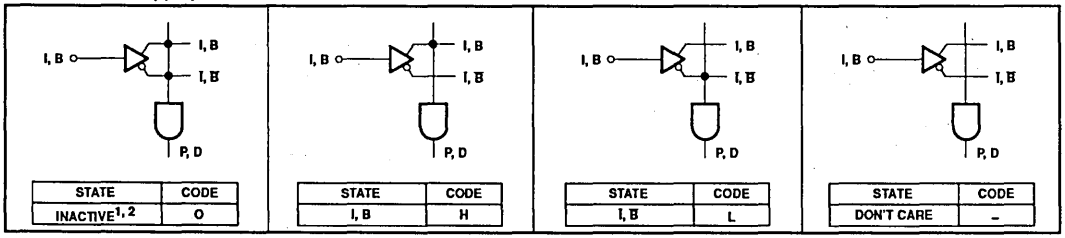
OUTPUT POLARITY – (B)



ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

ACTIVE LEVEL	CODE
LOW (INVERTING)	L

AND ARRAY – (I, B)



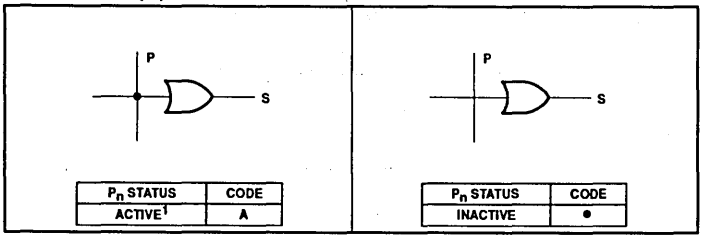
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
I, B̄	L

STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A

P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P_n, D_n.
2. Any gate P_n, D_n will be unconditionally inhibited if both the true and complement of any input (I, B) are left intact.

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

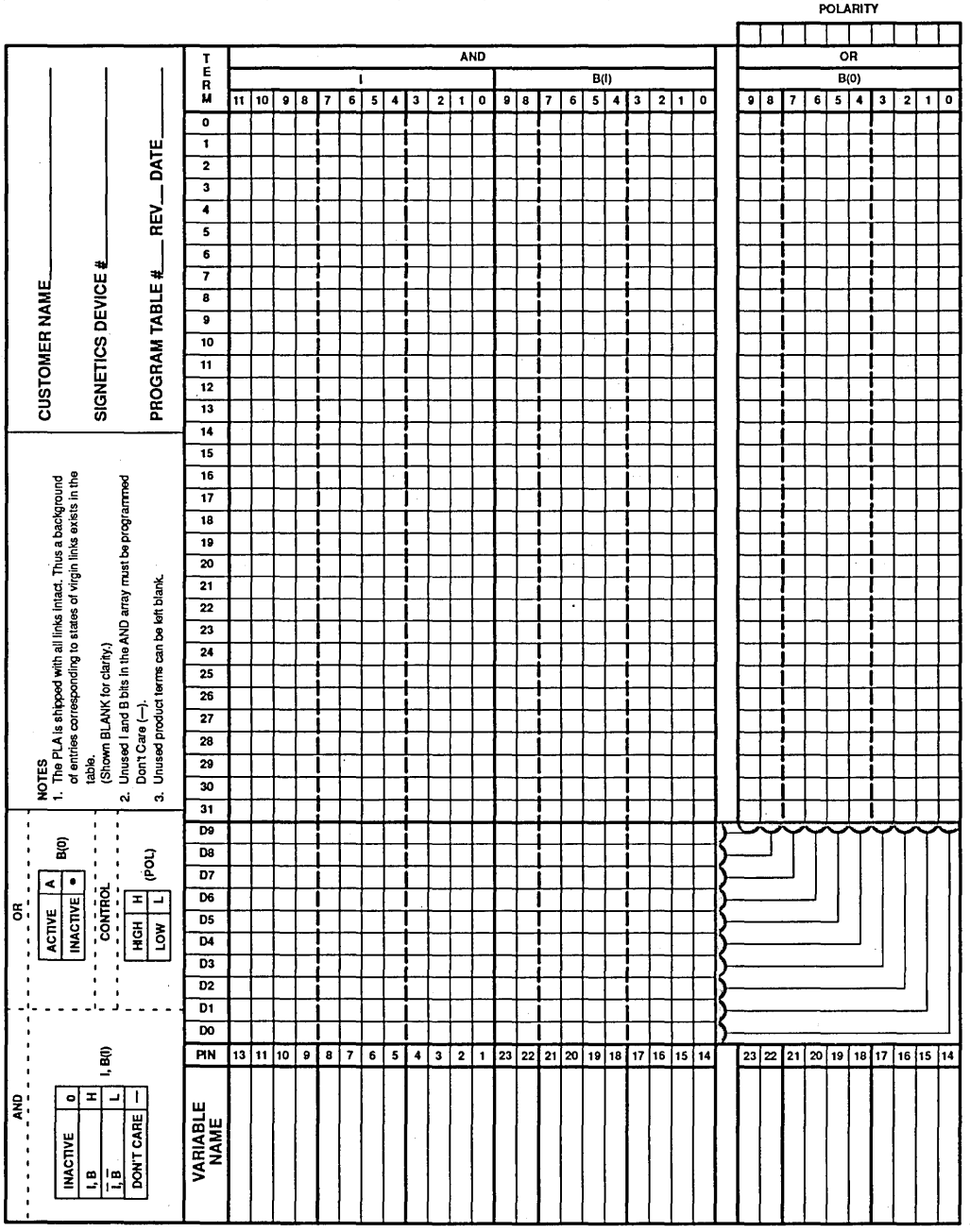
1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

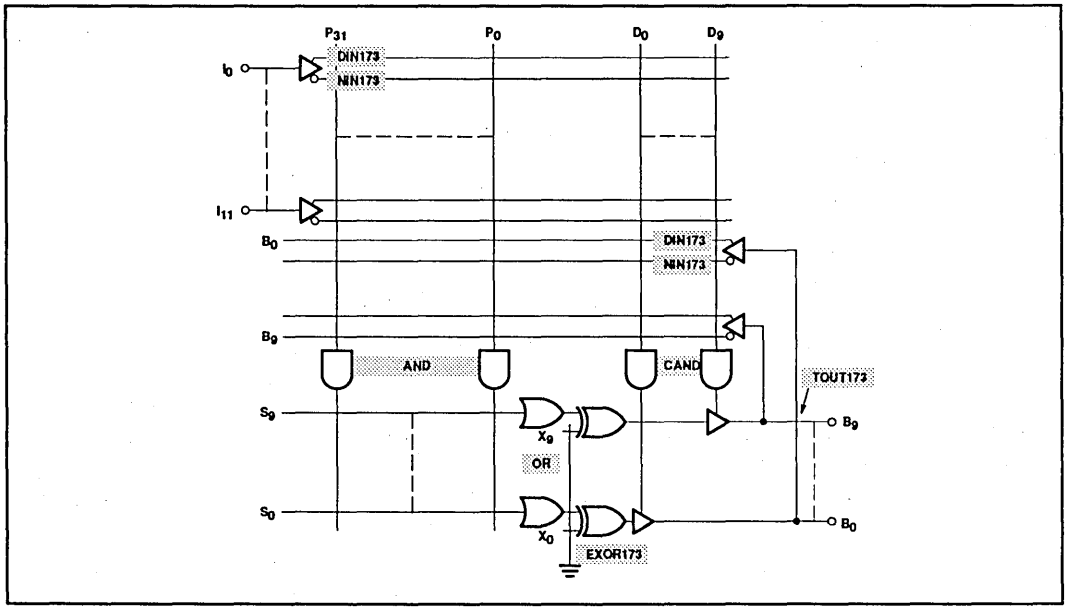
PROGRAM TABLE



Programmable logic arrays ($22 \times 42 \times 10$)

PLUS173B/D

SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–1422
ECN No.	99372
Date of Issue	April 17, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS173–10

Programmable logic array (22 × 42 × 10)

DESCRIPTION

The PLUS173–10 PLD is a high speed, combinatorial Programmable Logic Array. The Signetics state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce maximum propagation delays of 10ns or less.

The 24-pin PLUS173–10 device has a programmable AND array and a programmable OR array. Unlike PAL® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS173–10 device can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either Active-High or Active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS173–10 device is user-programmable using one of several commercially available, industry standard PLD programmers.

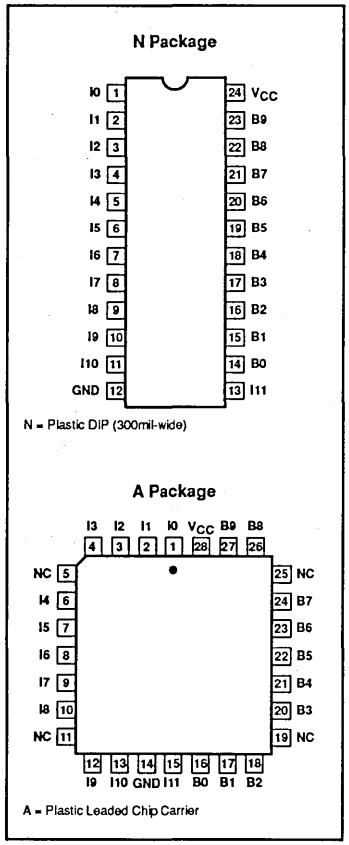
FEATURES

- I/O propagation delays
  - 10ns (worst case)
- Functional superset of 20L10 and most other 24-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 12 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 825mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

PIN CONFIGURATIONS

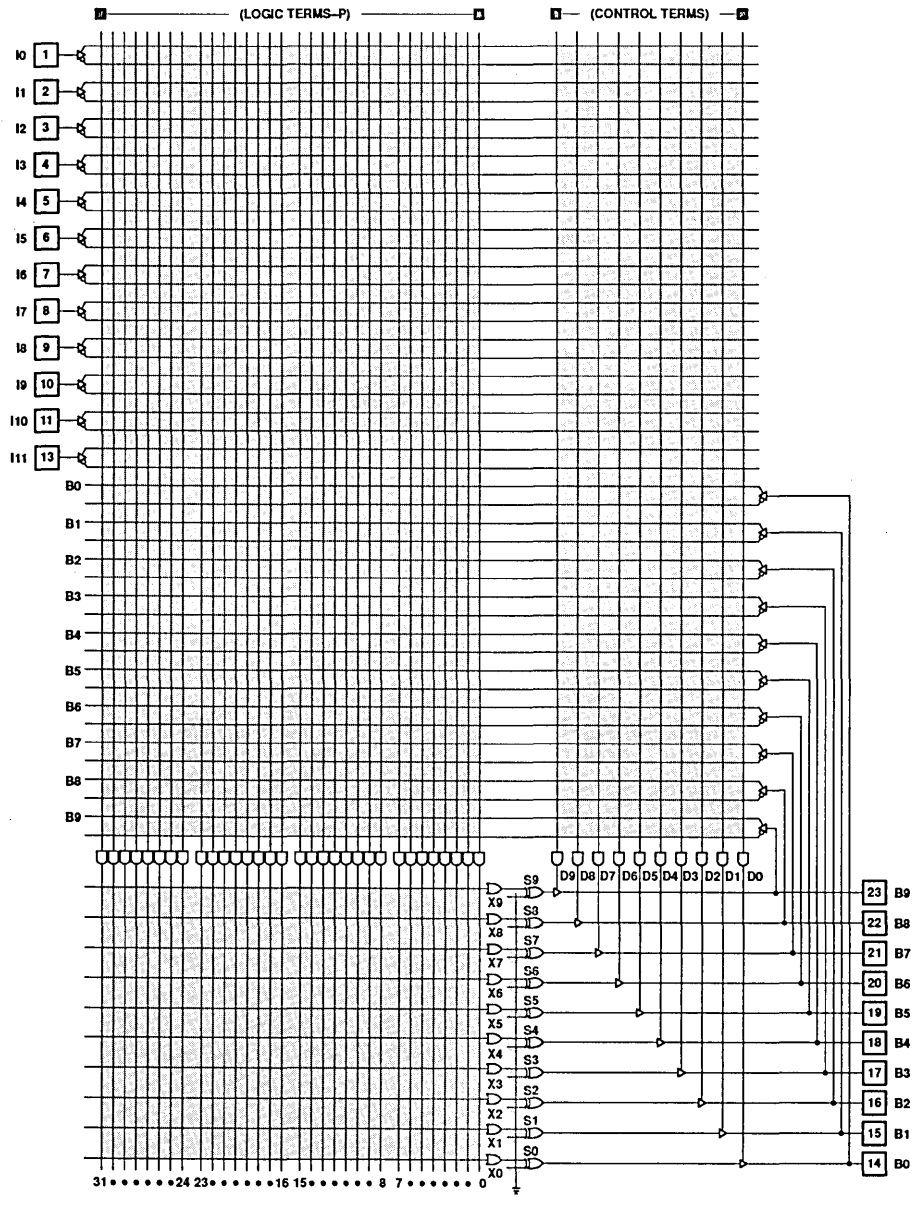


®PAL is a registered trademark of Advanced Micro Devices Corporation.

Programmable logic array (22 × 42 × 10)

PLUS173–10

LOGIC DIAGRAM

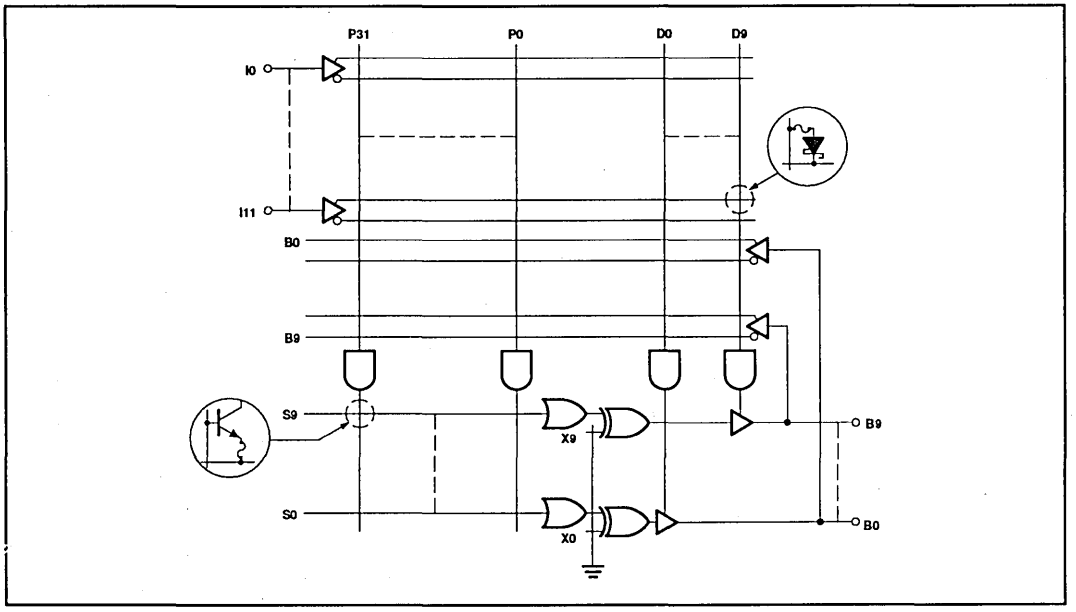


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic array (22 × 42 × 10)

PLUS173–10

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	t_{PD} (MAX)	ORDER CODE
24-Pin Plastic DIP 300mil-wide	10ns	PLUS173–10N
28-Pin Plastic Leaded Chip Carrier	10ns	PLUS173–10A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING Min	RATING Max	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{In}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100.0	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

Programmable logic array (22 × 42 × 10)

PLUS173–10

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = −2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			−100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V			80	µA
		V_{OUT} = 0.45V			−140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		165	210	mA
Capacitance						
		V_{CC} = 5V				
I_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs I0 – I4 = 0V, inputs I5 – I9 = 4.5V, I11 = 4.5V and I10 = 10V. For outputs B0 – B4 and for outputs B5 – B9 apply the same conditions except I11 = 0V.
5. Same conditions as Note 4 except input I11 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs I0 – I11 and B0 – B9 = 0V. Part in Virgin State.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I0 - I_{11}$).

Programmable logic array (22 × 42 × 10)

PLUS173–10

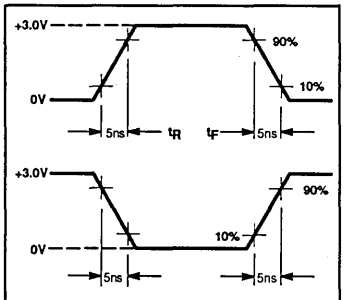
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		8	10	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		8	10	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		8	10	ns

NOTES:
1. For 3-State outputs; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

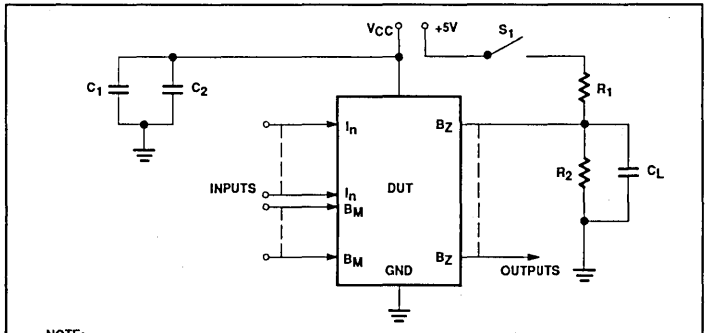
VOLTAGE WAVEFORM



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT



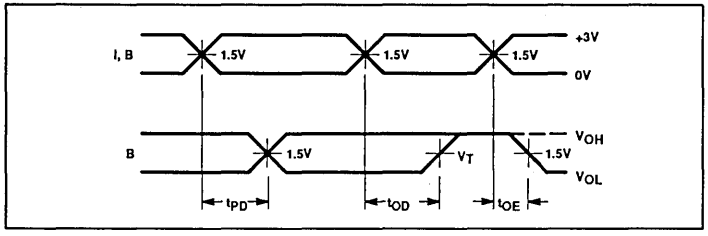
NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Test Load Circuit

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic array (22 × 42 × 10)

PLUS173–10

LOGIC PROGRAMMING

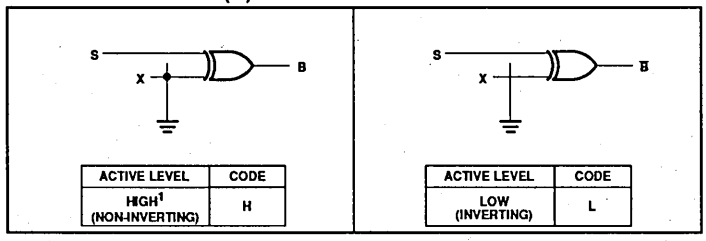
The PLUS173–10 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS173–10 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS173–10 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

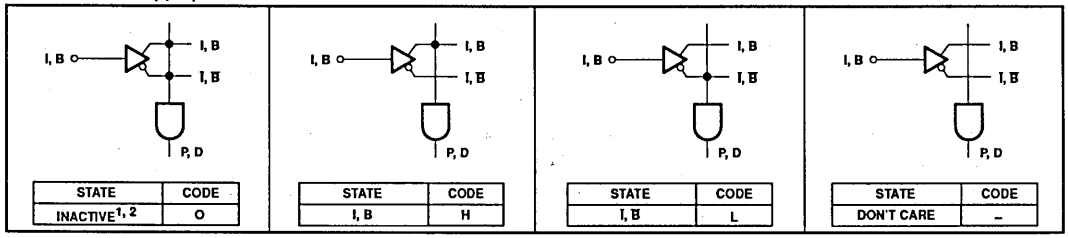
OUTPUT POLARITY – (B)



ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

ACTIVE LEVEL	CODE
LOW (INVERTING)	L

AND ARRAY – (I, B)



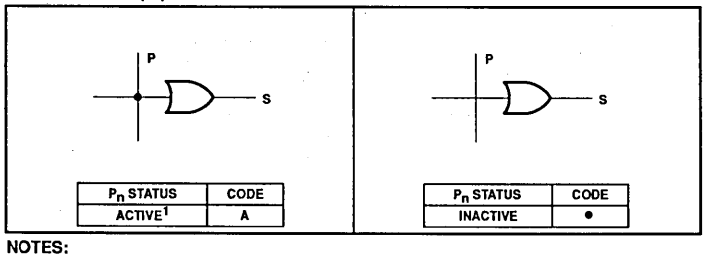
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
I, B̄	L

STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A

P_n STATUS	CODE
INACTIVE	•

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

NOTES:

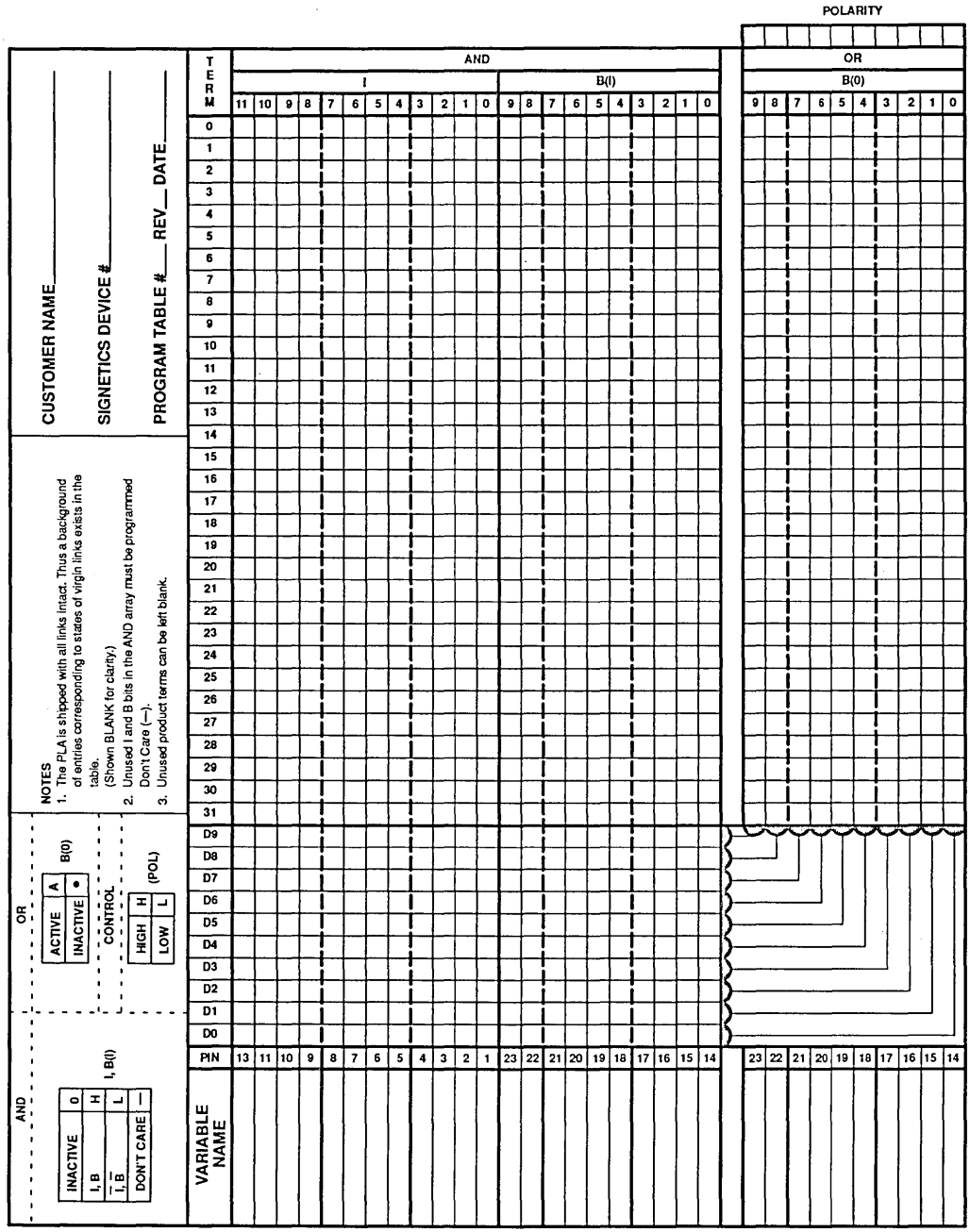
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P_n, D_n.
2. Any gate P_n, D_n will be unconditionally inhibited if both the true and complement of any input (I, B) are left intact.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic array (22 × 42 × 10)

PLUS173–10

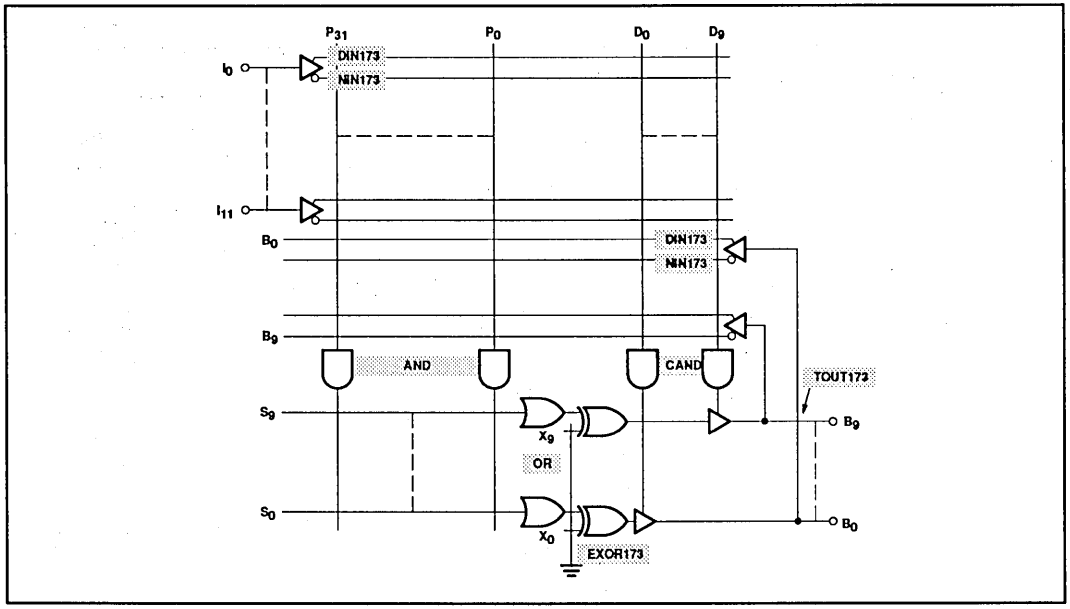
PLA PROGRAM TABLE



Programmable logic array (22 × 42 × 10)

PLUS173–10

SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–0308
ECN No.	93256
Date of Issue	May 11, 1988
Status	Product Specification
Programmable Logic Devices	

PLS100 / PLS101

Programmable logic arrays

(16 × 48 × 8)

DESCRIPTION

The PLS100 (3-State) and PLS101 (Open Collector) are bipolar, fuse Programmable Logic Arrays (PLAs). Each device utilizes the standard AND/OR/Invert architecture to directly implement custom sum of product equations.

Each device consists of 16 dedicated inputs and 8 dedicated outputs. Each output is capable of being actively controlled by any or all of the 48 product terms. The True, Complement, or Don't Care condition of each of the 16 inputs and be ANDed together to comprise one P-term. All 48 P-terms can be selectively ORed to each output.

The PLS100 and PLS101 are fully TTL compatible, and chip enable control for expansion of input variables and output inhibit. They feature either Open Collector or 3-State outputs for ease of expansion of product terms and application in bus-organized systems.

Order codes are listed in the Ordering Information Table.

FEATURES

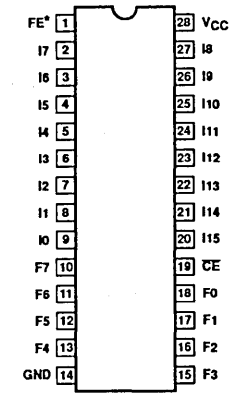
- Field-programmable (Ni-Cr link)
- Input variables: 16
- Output functions: 8
- Product terms: 48
- I/O propagation delay: 50ns (max.)
- Power dissipation: 600mW (typ.)
- Input loading: –100μA (max.)
- Chip Enable input
- Output option:
  - PLS100: 3-State
  - PLS101: Open-Collector
- Output disable function:
  - 3-State: Hi-Z
  - Open-Collector: High

APPLICATIONS

- CRT display systems
- Code conversion
- Peripheral controllers
- Function generators
- Look-up and decision tables
- Microprogramming
- Address mapping
- Character generators
- Data security encoders
- Fault detectors
- Frequency synthesizers
- 16-bit to 8-bit bus interface
- Random logic replacement

PIN CONFIGURATIONS

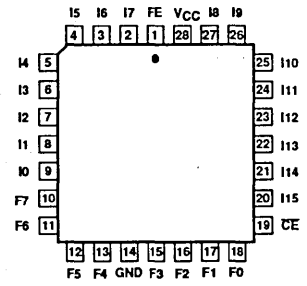
N Package



* Fuse Enable Pin: It is recommended that this pin be left open or connected to ground during normal operation.

N = Plastic DIP (600mil-wide)

A Package

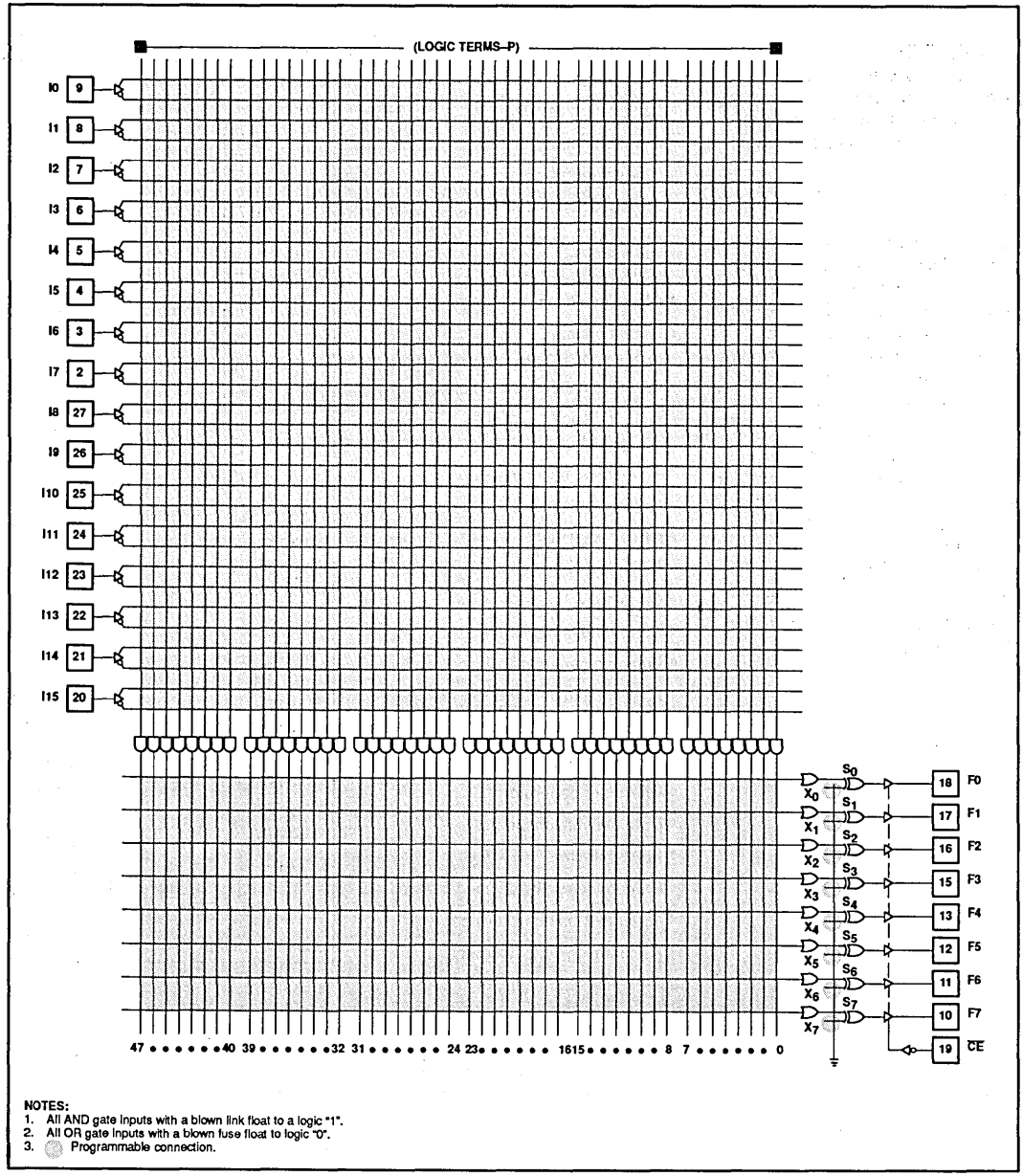


A = Plastic Leaded Chip Carrier

Programmable logic arrays (16 × 48 × 8)

PLS100 / PLS101

LOGIC DIAGRAM

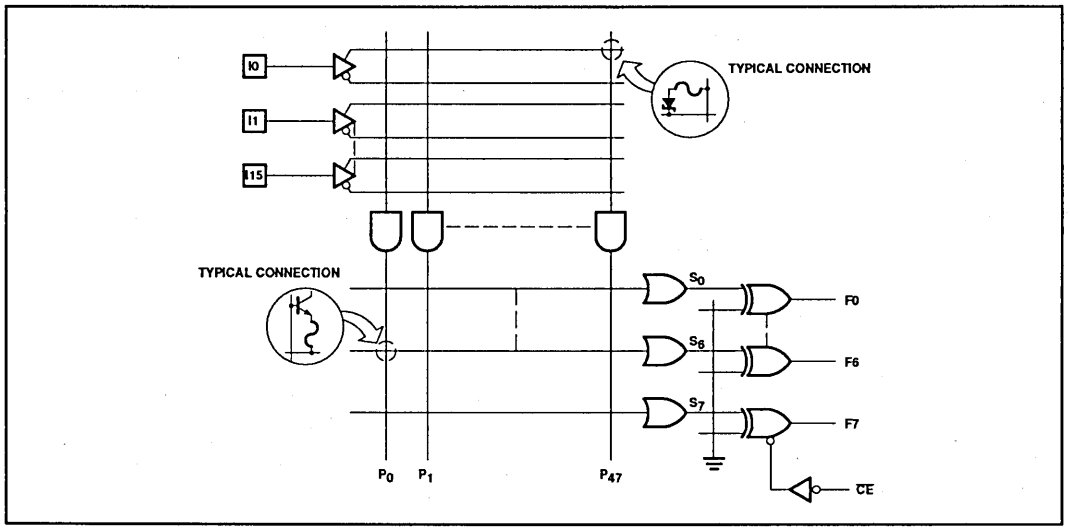


NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ◌ Programmable connection.

Programmable logic arrays (16 × 48 × 8)

PLS100 / PLS101

FUNCTIONAL DIAGRAM



ORDERING INFORMATION

DESCRIPTION	3-STATE	OPEN COLLECTOR
28-Pin Plastic Dual In-Line 600mil-wide	PLS100N	PLS101N
28-Pin Plastic Leaded Chip Carrier	PLS100A	PLS101A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7.0	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Output voltage	+5.5	V_{DC}
I_{IN}	Input current	±30	mA
I_{OUT}	Output current	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS100 device is also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Philips Components–Signetics Military Data Handbook.

Programmable logic arrays (16 × 48 × 8)

PLS100 / PLS101

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High (PLS100)[4]	I_{OH} = –2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		< 1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		–10	–100	µA
Output current						
$I_{O(OFF)}$	Hi-Z state (PLS100)	$\overline{CE}$ = High, V_{CC} = MAX				
		V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit (PLS100) [3,6]	$\overline{CE}$ = Low, V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		120	170	mA
Capacitance						
		$\overline{CE}$ = High, V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		17		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one pin at a time.
4. Measured with V_{IL} applied to $\overline{CE}$ and a logic high stored.
5. Measured with a programmed logic condition for which the output test is at a low logic level. Output sink current is applied through a resistor to V_{CC}.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with the Chip Enable input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic arrays (16 × 48 × 8)

PLS100 / PLS101

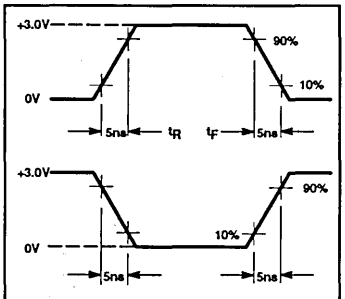
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	TO	FROM	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Propagation delay[2]							
t_{PD}	Input	Output	Input		35	50	ns
t_{CE}	Chip Enable[3]	Output	Chip Enable		15	30	ns
Disable time							
t_{CD}	Chip Disable[3]	Output	Chip Enable		15	30	ns

NOTES:
1. All typical values are at $V_{CC} = 5V$. $T_{amb} = +25°C$.
2. All propagation delays are measured and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

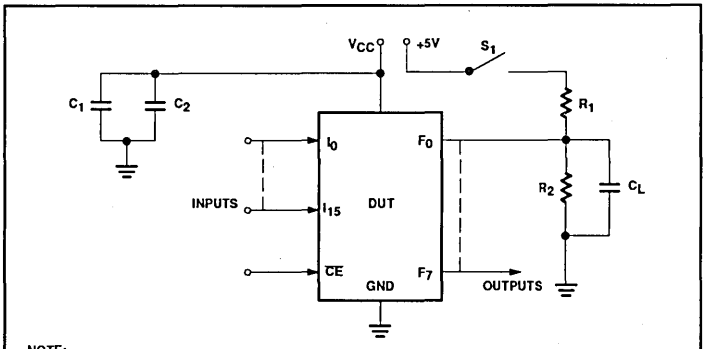
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

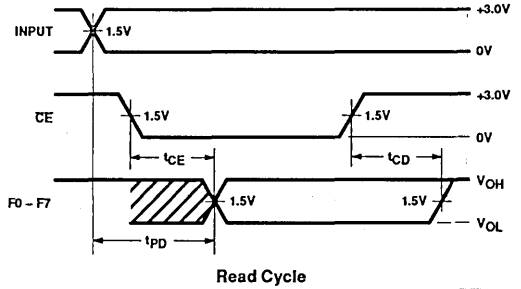


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CE}	Delay between beginning of Chip Enable Low (with Input valid) and when Data Output becomes valid.
t_{CD}	Delay between when Chip Enable becomes High and Data Output is in off state (Hi-Z or High).
t_{PD}	Delay between beginning of valid Input (with Chip Enable Low) and when Data Output becomes valid.

TIMING DIAGRAM



Read Cycle

Programmable logic arrays (16 × 48 × 8)

PLS100 / PLS101

LOGIC PROGRAMMING

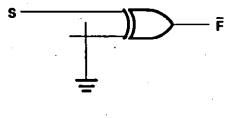
PLS100/PLS101 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE, Data I/O Corporation's ABEL and Logical Devices Inc.'s CUPL design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

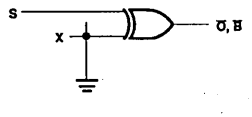
PLS100/PLS101 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics' AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The sumbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

OUTPUT POLARITY – (F)

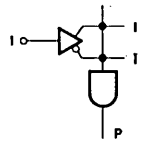


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

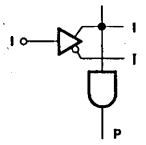


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

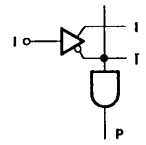
"AND" ARRAY – (I)



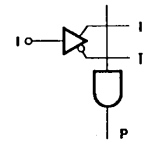
STATE	CODE
INACTIVE[1,2]	O



STATE	CODE
I	H

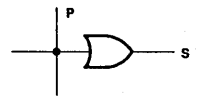


STATE	CODE
I	L

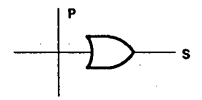


STATE	CODE
DON'T CARE	–

"OR" ARRAY – (F)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all links. It is normally associated with all unused (inactive) AND gates P_n.
2. Any gate P_n will be unconditionally inhibited if any one of its (I) link pairs is left intact.

VIRGIN STATE

The PLS100/101 virgin devices are factory shipped in an unprogrammed state, with all fuses intact, such that:

1. All P_n terms are disabled (inactive) in the AND array.
2. All P_n terms are active in the OR array.
3. All outputs are Active-High.

Programmable logic arrays (16 × 48 × 8)

PLS100 / PLS101

PROGRAM TABLE

PROGRAM TABLE ENTRIES

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # _____ CF (XXXX)

CUSTOMER SYMBOLIZED PART # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____ REV ____ DATE ____

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	Im	Don't Care	Prod. Term Present in Fp	Prod. Term Not Present in Fp	Active High	Active Low
H	L	– (dash)	A	• (period)	H	L

NOTE
Enter (–) for unused inputs of used P-terms.

NOTES
1. Entries independent of output polarity.
2. Enter (A) for unused outputs of used P-terms.

NOTES
1. Polarity programmed once only.
2. Enter (H) for all unused outputs.

AND

TERM	INPUT (I_m)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
PIN NO.	20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9
VARIABLE NAME																

POLARITY

OR

OUTPUT (F_p)

7	6	5	4	3	2	1	0
10	11	12	13	15	16	17	18

Section 5
Programmable Logic Sequencer Device Data Sheets

INDEX

Document No.	853–0317
ECN No.	93255
Date of Issue	May 11, 1988
Status	Product Specification
Programmable Logic Devices	

PLS155

Programmable logic sequencer ($16 \times 45 \times 12$)

DESCRIPTION

The PLS155 is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate F_C. It features 4 registered I/O outputs (F) in conjunction with 8 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates can drive bidirectional I/O lines (B), whose output polarity is individually programmable through a set of Ex-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the AND gates can drive the J-K inputs of all flip-flops. The Asynchronous Preset and Reset lines (P, R), are driven from the OR matrix.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS155 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

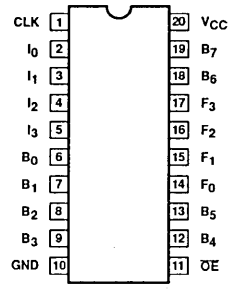
Order codes are listed in the Ordering Information Table.

FEATURES

- f_{MAX} = 14MHz
  - 18.2MHz clock rate
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 8 bidirectional I/O lines
- 4 bidirectional registers
- J-K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Input loading: −100μA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- 3-State outputs

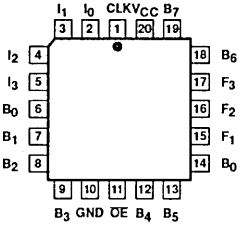
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



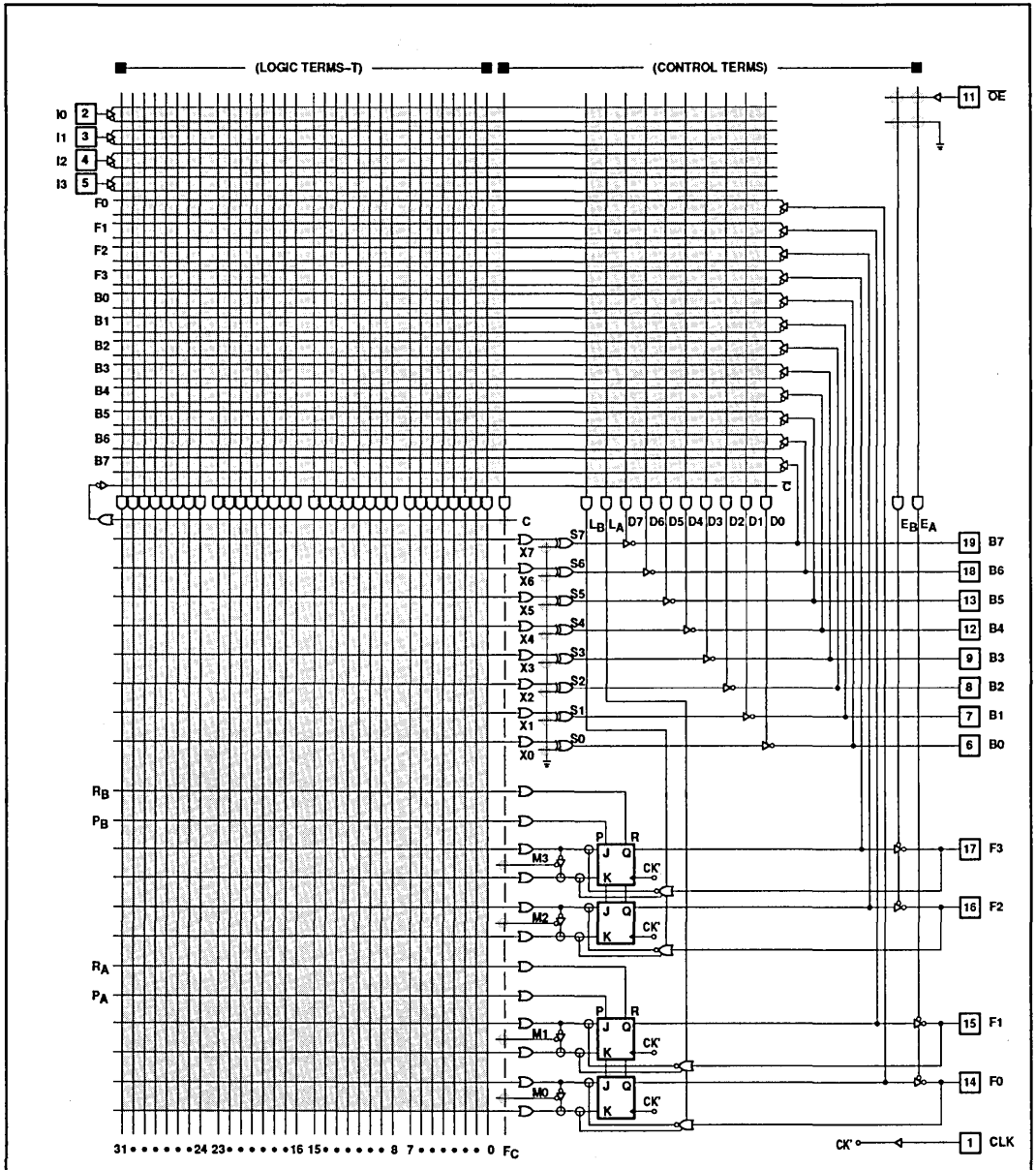
A = Plastic Leaded Chip Carrier

APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

Programmable logic sequencer (16 × 45 × 12) PLS155

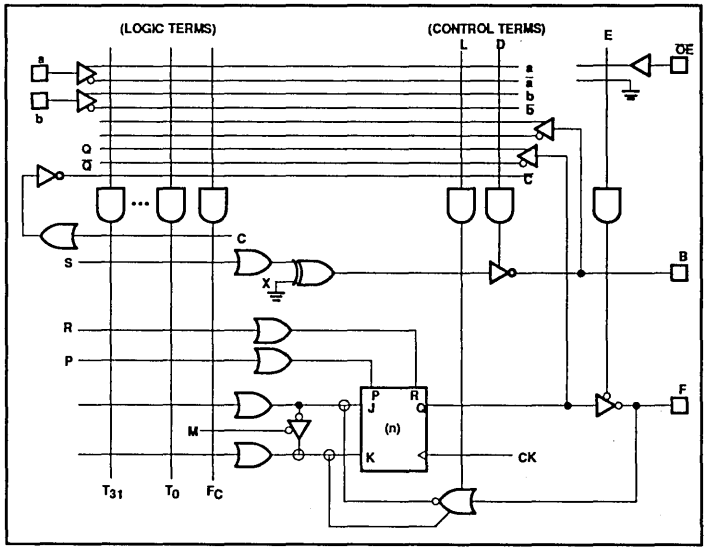
LOGIC DIAGRAM



NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ◌ Programmable connection.

Programmable logic sequencer (16 × 45 × 12) PLS155

FUNCTIONAL DIAGRAM

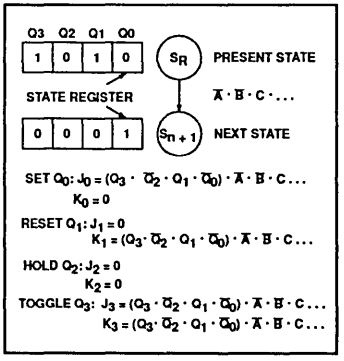


FLIP-FLOP TRUTH TABLE

OE	L	CK	P	R	J	K	Q	F
H								Hi-Z
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

NOTES:
1. Positive Logic:
   $J\text{-}K = T_0 + \overline{T}_1 + T_2 \ldots\ldots\ldots\ldots T_{31}$
   $T_n = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. OE is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

Programmable logic sequencer (16 × 45 × 12) PLS155

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic DIP (300mil-wide)	PLS155N
20-Pin Plastic Leaded Chip Carrier	PLS155A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS Min	RATINGS Max	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

Programmable logic sequencer (16 × 45 × 12) PLS155

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 10mA		0.35	0.5	V
Input current[5]						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[5, 6]	V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V		−1	−140	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[4]	V_{CC} = MAX		150	190	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
5. Leakage values are a combination of input and output leakage.
6. Measured with V_{IH} applied to $\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.

Programmable logic sequencer (16 × 45 × 12) PLS155

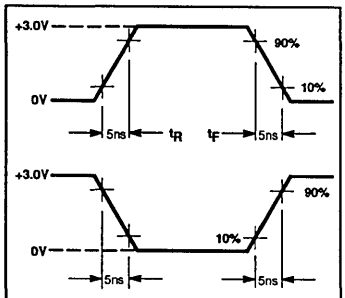
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	25	20		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	30	20		ns
t_{CKP}	Period	CK +	CK +	C_L = 30pF	70	50		ns
t_{PRH}	Preset/Reset pulse	(I,B) –	(I,B) +	C_L = 30pF	40	30		ns
Setup time[5]								
t_{IS1}	Input	(I,B) ±	CK +	C_L = 30pF	40	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	20	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I,B) ±	CK +	C_L = 30pF	65	40		ns
Hold time								
t_{IH1}	Input	(I,B) ±	CK +	C_L = 30pF	0	–10		ns
t_{IH2}	Input	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delays								
t_{CKO}	Clock	CK +	F ±	C_L = 30pF		25	30	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I,B) ±	B ±	C_L = 30pF		40	50	ns
t_{OE2}	Output enable[3]	(I,B) +	B ±	C_L = 30pF		35	55	ns
t_{OD2}	Output disable[3]	(I,B) –	B +	C_L = 5pF		30	35	ns
t_{PRO}	Preset/Reset	(I,B) +	F ±	C_L = 30pF		50	55	ns

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 95ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.
6. For test circuits, waveforms and timing diagrams see the following pages.

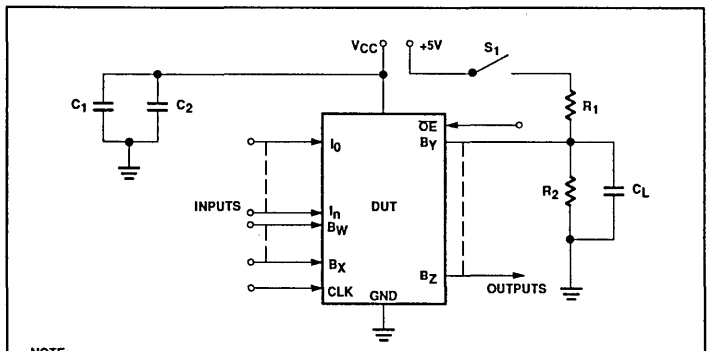
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

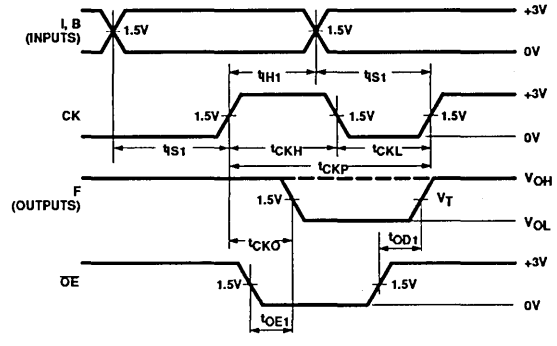
TEST LOAD CIRCUIT



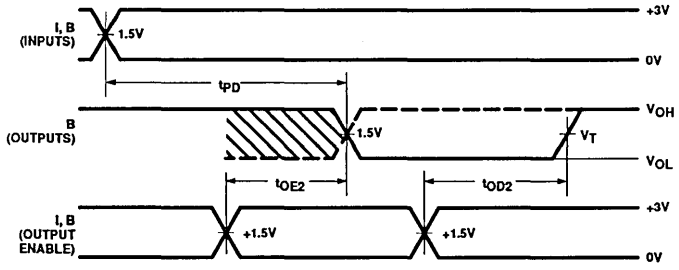
NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Programmable logic sequencer (16 × 45 × 12) PLS155

TIMING DIAGRAMS



Flip–Flop Outputs

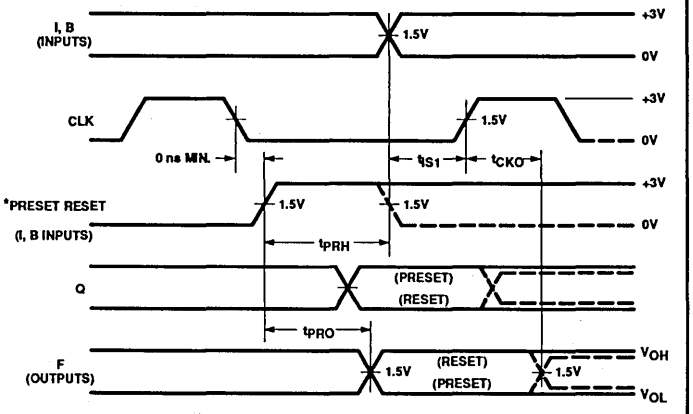


Gate Outputs

TIMING DEFINITIONS

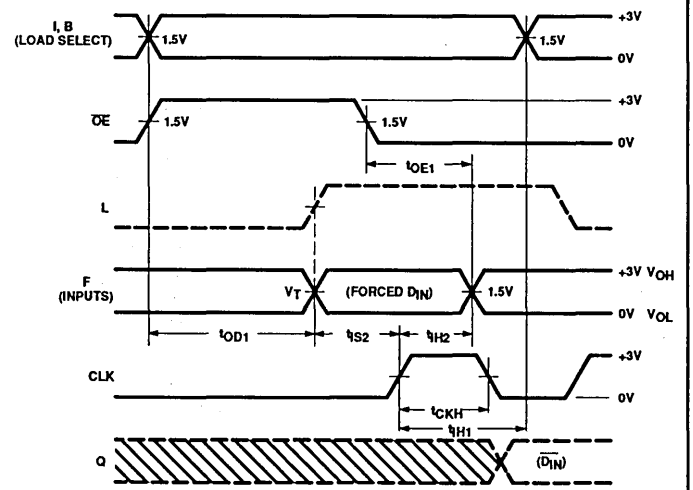
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip–flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip–flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF–State.
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the OFF–State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip–flop outputs become valid.

TIMING DIAGRAMS (Continued)



* The leading edge of preset/reset must occur only when the input clock is "low", and must remain "high" as long as required to override clock. The falling edge of preset/reset can never go "low" when the input clock is "high".

Asynchronous Preset/Reset



Flip-Flop Input Mode

Programmable logic sequencer (16 × 45 × 12) PLS155

LOGIC PROGRAMMING

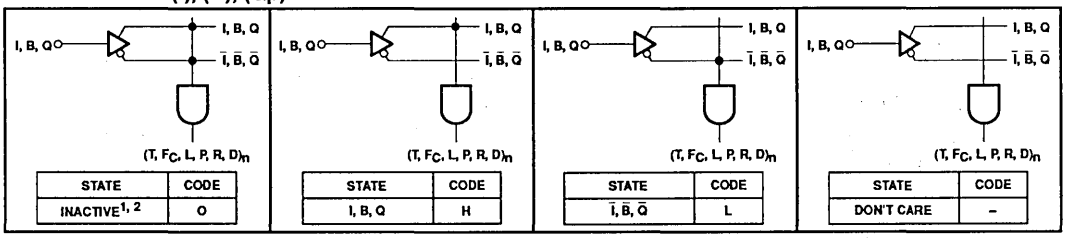
The PLS155 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Singetics' AMAZE, Data I/O Corporation's ABEL™, and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS155 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

"AND" ARRAY – (I), (B), (Qp)



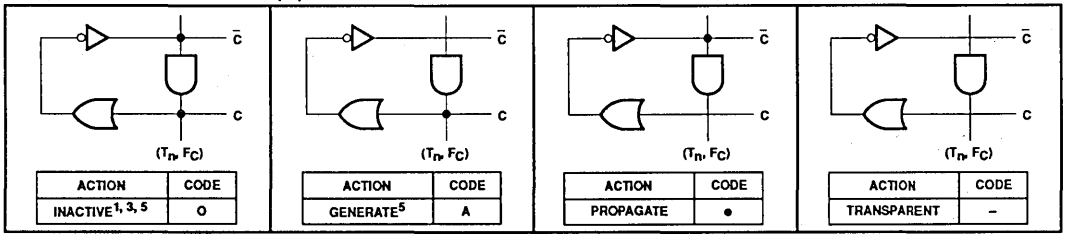
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
Ī, B̄, Q̄	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



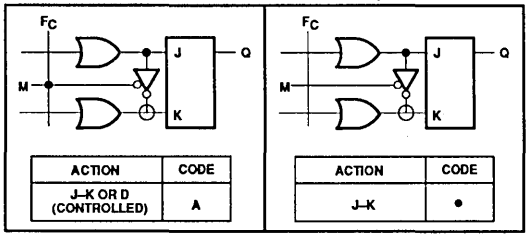
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)	A

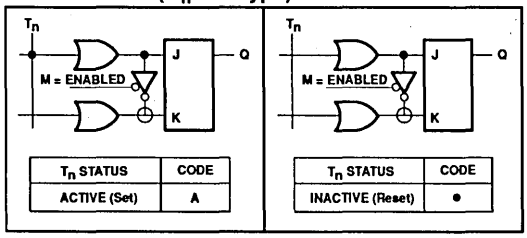
ACTION	CODE
J–K	•

Notes on following page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic sequencer (16 × 45 × 12) PLS155

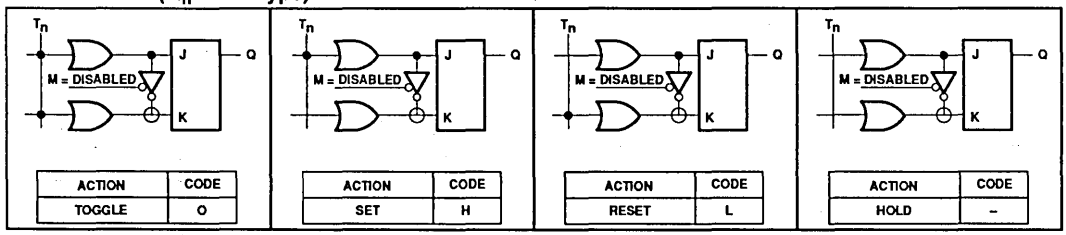
"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)	A

T_n STATUS	CODE
INACTIVE (Reset)	•

"OR" ARRAY – (Q_n = J-K Type)



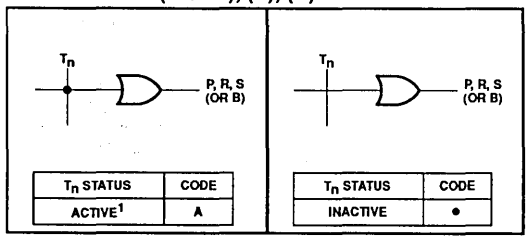
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

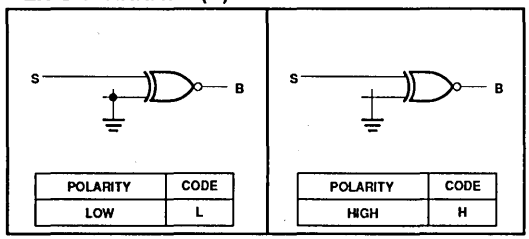
"OR" ARRAY – (S or B), (P), (R)



T_n STATUS	CODE
ACTIVE[1]	A

T_n STATUS	CODE
INACTIVE	•

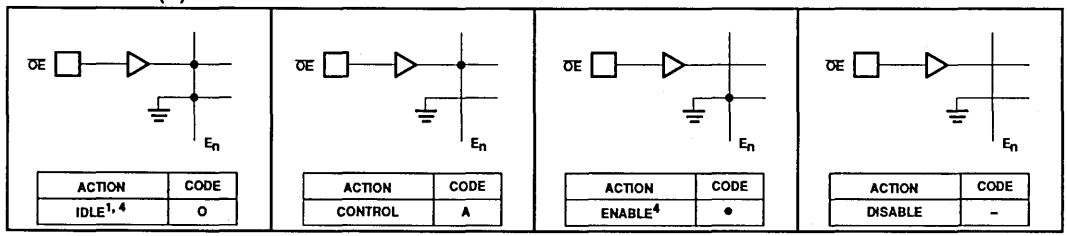
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

ACTION	CODE
ENABLE[4]	•

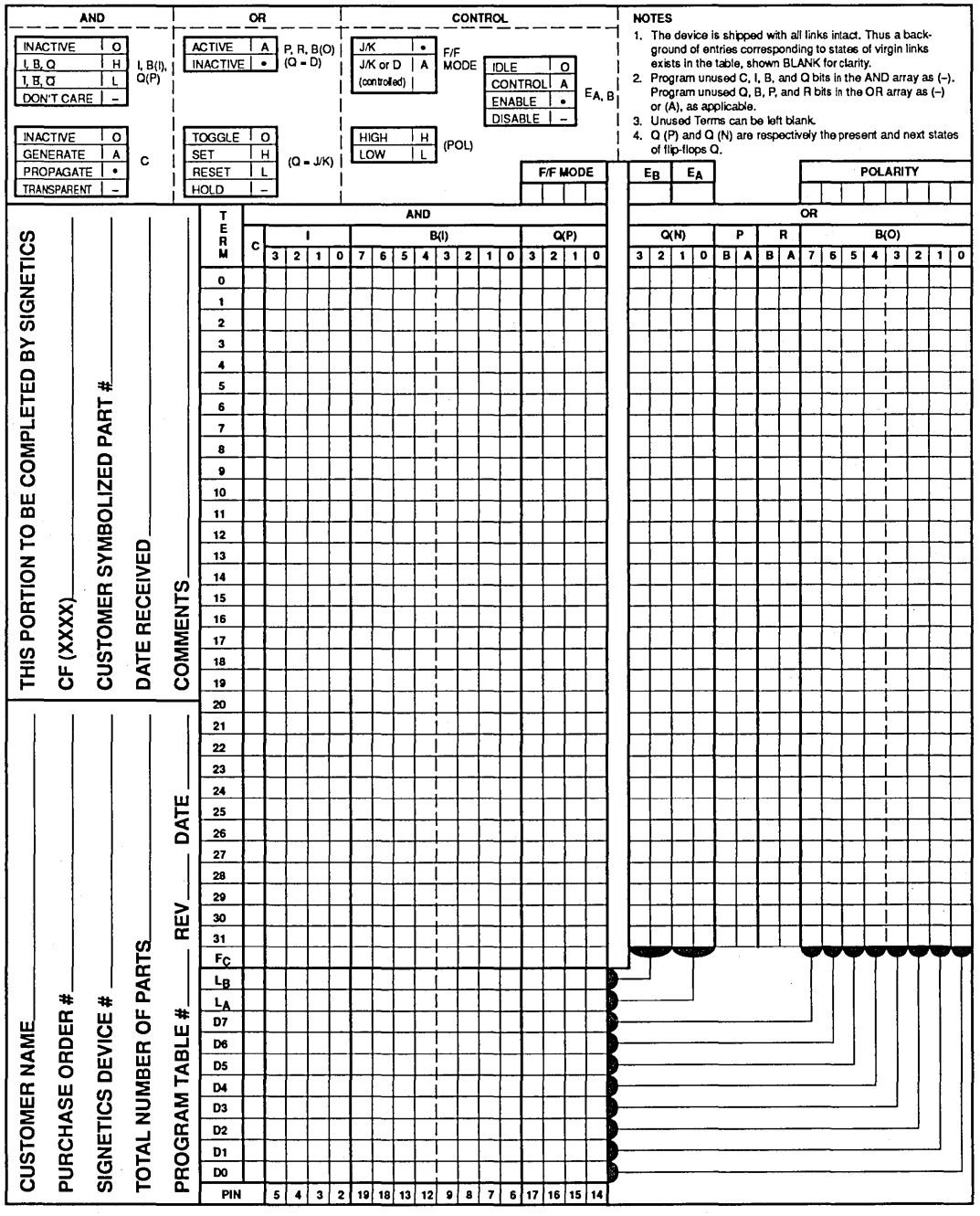
ACTION	CODE
DISABLE	–

NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate $(T, F_C, L, P, R, D)_n$ will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates $(L, P, R, D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (16 × 45 × 12) PLS155

PROGRAM TABLE



Philips Components–Signetics

Document No.	853–0318
ECN No.	93255
Date of Issue	May 11, 1988
Status	Product Specification
Programmable Logic Devices	

PLS157

Programmable logic sequencer (16 × 45 × 12)

DESCRIPTION

The PLS157 is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate F_C. It features 6 registered I/O outputs (F) in conjunction with 6 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement (I, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of Ex-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the 32 AND gates can drive the J-K inputs of all flip-flops. The Asynchronous Preset and Reset lines (P, R), are driven from the AND array for 4 of the 8 registers. The Preset and Reset lines (P, R) controlling the lower four registers are driven from the OR matrix.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS157 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

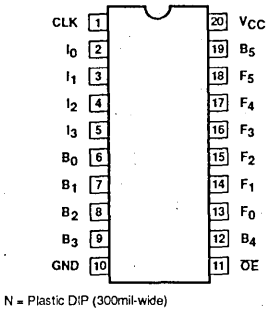
Order codes are listed in the Ordering Information Table.

FEATURES

- f_{MAX} = 14MHz
  - 18.2MHz clock rate
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 6 bidirectional I/O lines
- 6 bidirectional registers
- J-K, T, or D-type flip-flops
- 3-State outputs
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Input loading: –100μA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible

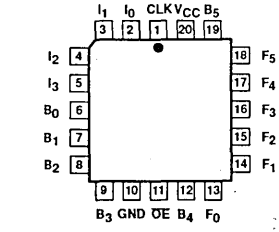
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



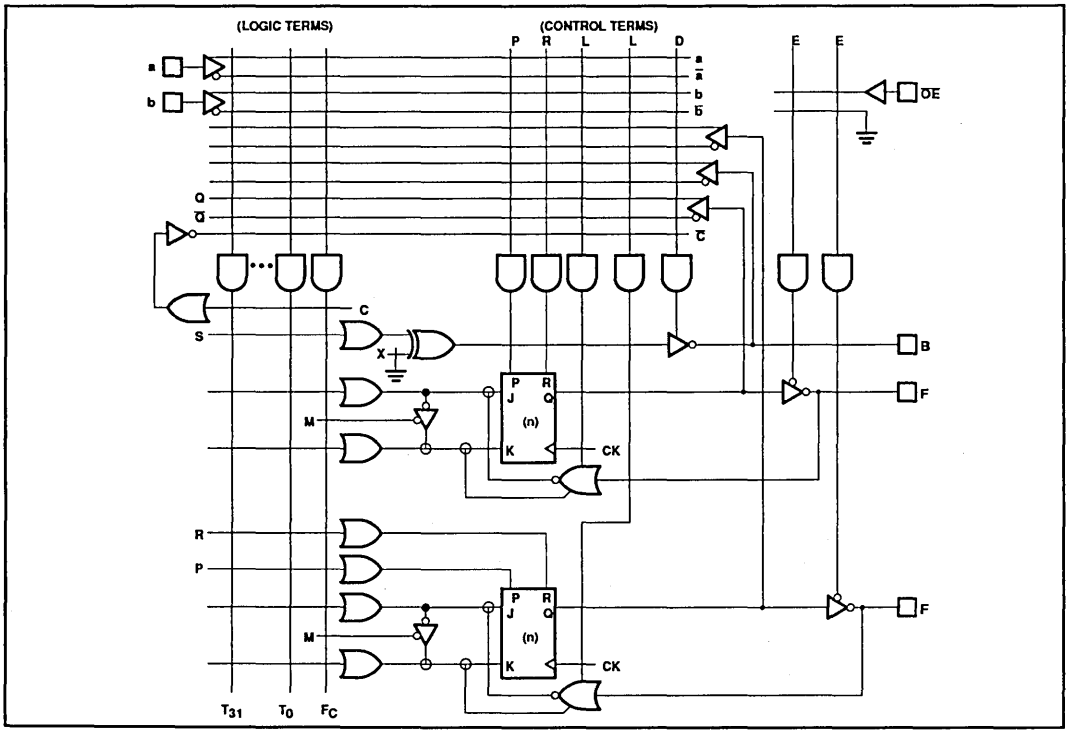
A = Plastic Leaded Chip Carrier

APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

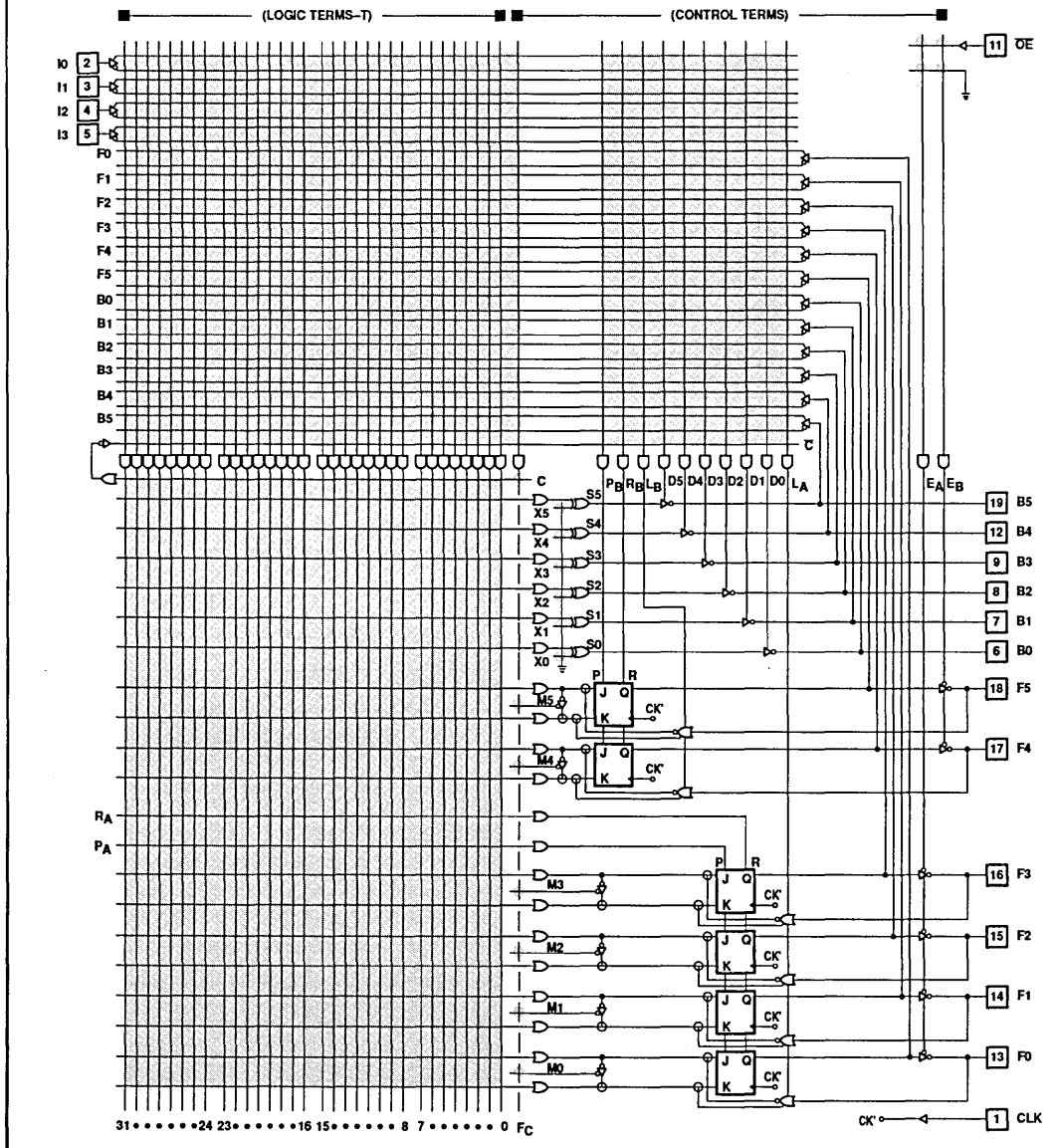
Programmable logic sequencer (16 × 45 × 12) PLS157

FUNCTIONAL DIAGRAM



Programmable logic sequencer (16 × 45 × 12) PLS157

LOGIC DIAGRAM



NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ◌ Programmable connection.

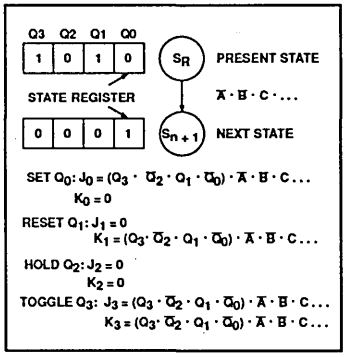
Programmable logic sequencer (16 × 45 × 12) — PLS157

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								Hi-Z
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H* *
	X	↑	X	X	H	L	H	L* *

NOTES:

1. Positive Logic:
   $J\text{-}K = T_0 + T_1 + T_2 \ldots\ldots\ldots\ldots T_{31}$
   $T_n = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. * * = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic DIP (300mil-wide)	PLS157N
20-Pin Plastic Leaded Chip Carrier	PLS157A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

Programmable logic sequencer (16 × 45 × 12) PLS157

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 10mA		0.35	0.5	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[5, 6]	V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V		−1	−140	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[4]	V_{CC} = MAX		150	190	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
5. Leakage values are a combination of input and output leakage.
6. Measured with V_{IH} applied to $\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.

Programmable logic sequencer (16 × 45 × 12) PLS157

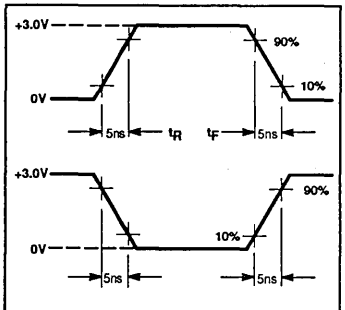
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	25	20		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	30	20		ns
t_{CKP}	Period	CK +	CK +	C_L = 30pF	70	50		ns
t_{PRH}	Preset/Reset pulse	(I,B) –	(I,B) +	C_L = 30pF	40	30		ns
Setup time[5]								
t_{IS1}	Input	(I,B) ±	CK +	C_L = 30pF	40	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	20	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I,B) ±	CK +	C_L = 30pF	65	40		ns
Hold time								
t_{IH1}	Input	(I,B) ±	CK +	C_L = 30pF	0	–10		ns
t_{IH2}	Input	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delays								
t_{CKO}	Clock	CK +	F ±	C_L = 30pF		25	30	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I,B) ±	B ±	C_L = 30pF		40	50	ns
t_{OE2}	Output enable[3]	(I,B) +	B ±	C_L = 30pF		35	55	ns
t_{OD2}	Output disable[3]	(I,B) –	B +	C_L = 5pF		30	35	ns
t_{PRO}	Preset/Reset	(I,B) +	F ±	C_L = 30pF		50	55	ns

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 95ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.
6. For test circuits, waveforms and timing diagrams see the following pages.

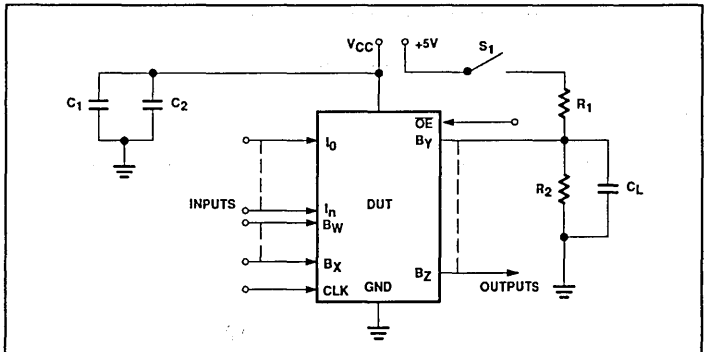
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

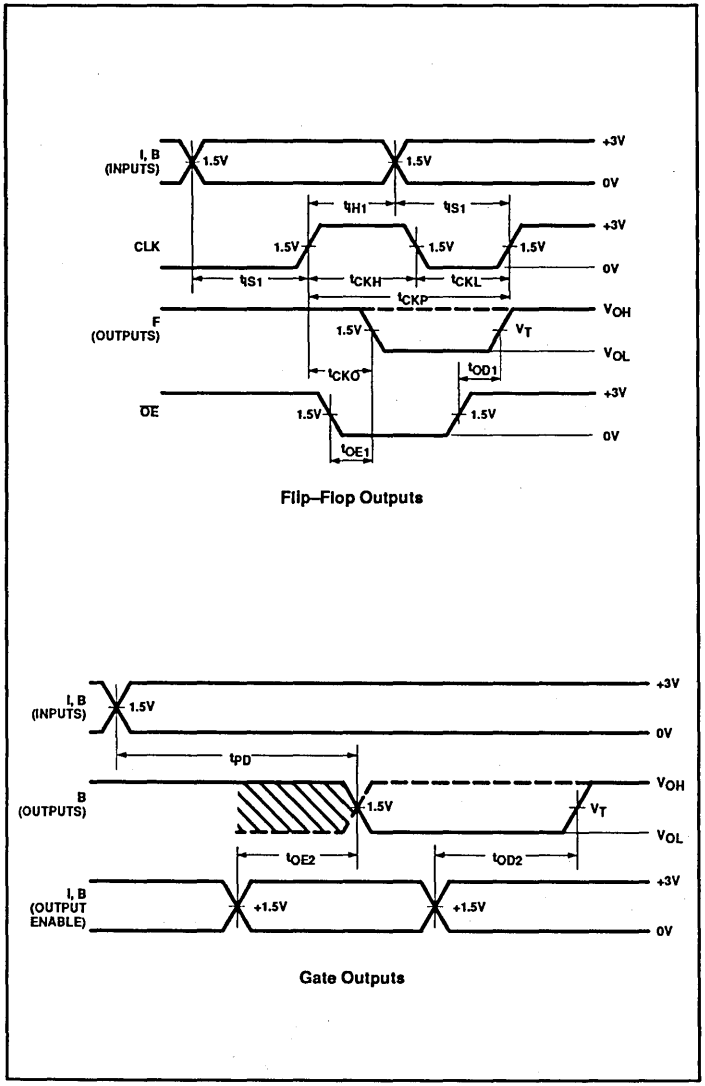
TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Programmable logic sequencer (16 × 45 × 12) PLS157

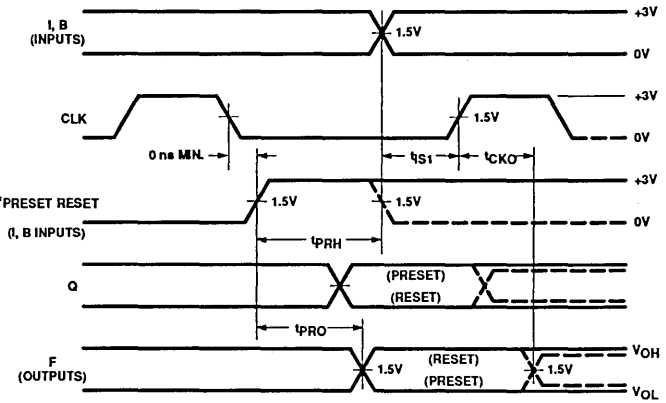
TIMING DIAGRAMS



TIMING DEFINITIONS

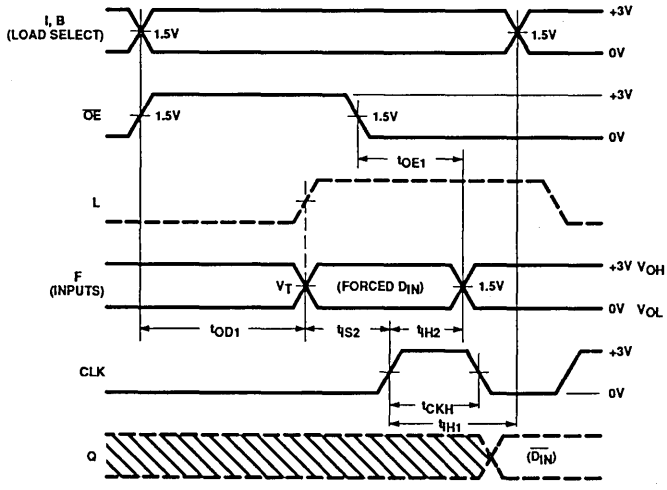
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

TIMING DIAGRAMS (Continued)



* The leading edge of preset/reset must occur only when the input clock is "low", and must remain "high" as long as required to override clock. The falling edge of preset/reset can never go "low" when the input clock is "high".

Asynchronous Preset/Reset



Flip-Flop Input Mode

Programmable logic sequencer (16 × 45 × 12) PLS157

LOGIC PROGRAMMING

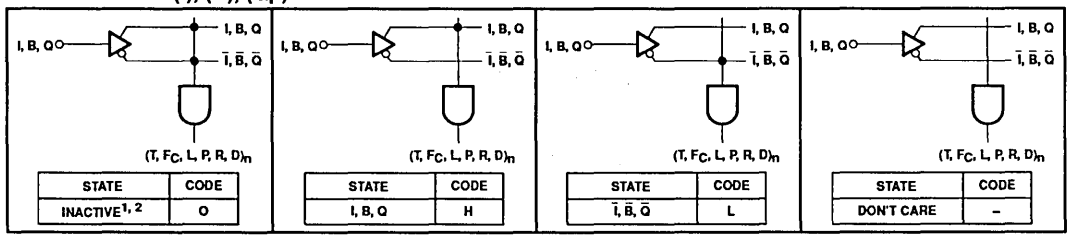
The PLS157 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS157 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

"AND" ARRAY – (I), (B), (Qp)



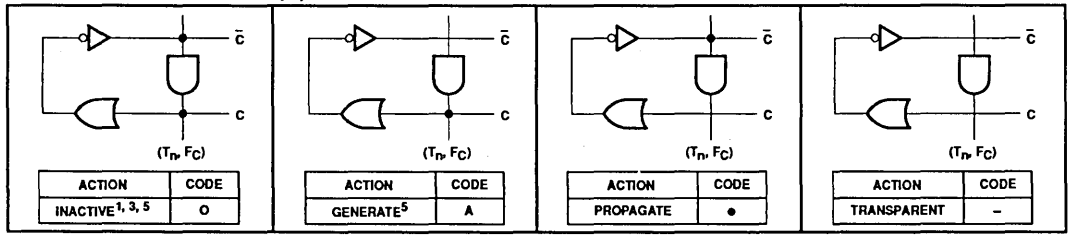
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
$\bar{I}, \bar{B}, \bar{Q}$	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



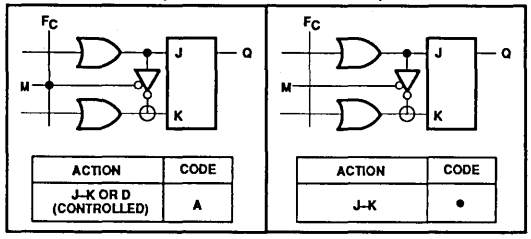
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)	A

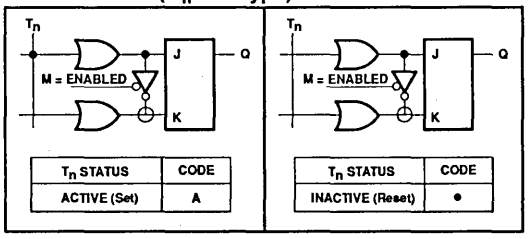
ACTION	CODE
J–K	•

Notes on following page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic sequencer (16 × 45 × 12) PLS157

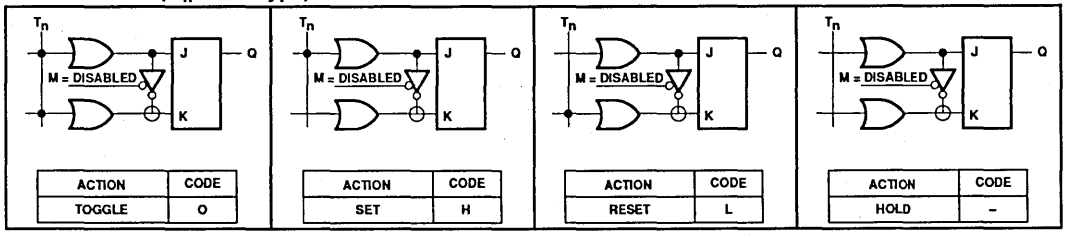
"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)	A

T_n STATUS	CODE
INACTIVE (Reset)	•

"OR" ARRAY – (Q_n = J-K Type)



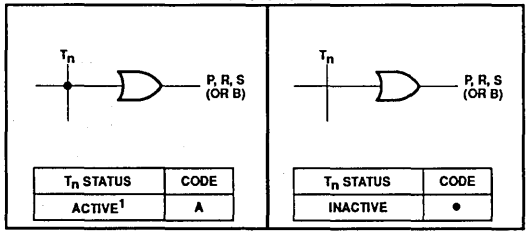
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

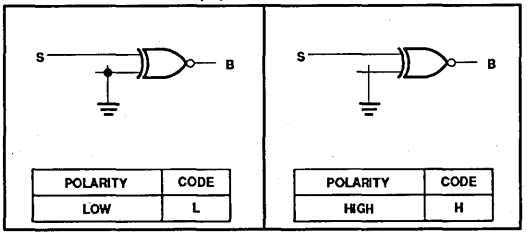
"OR" ARRAY – (S or B), (P), (R)



T_n STATUS	CODE
ACTIVE[1]	A

T_n STATUS	CODE
INACTIVE	•

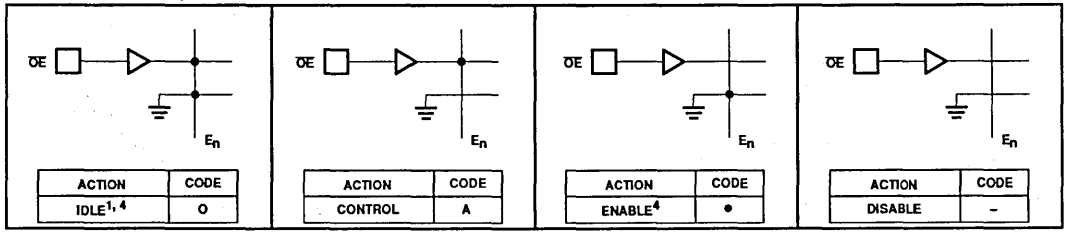
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

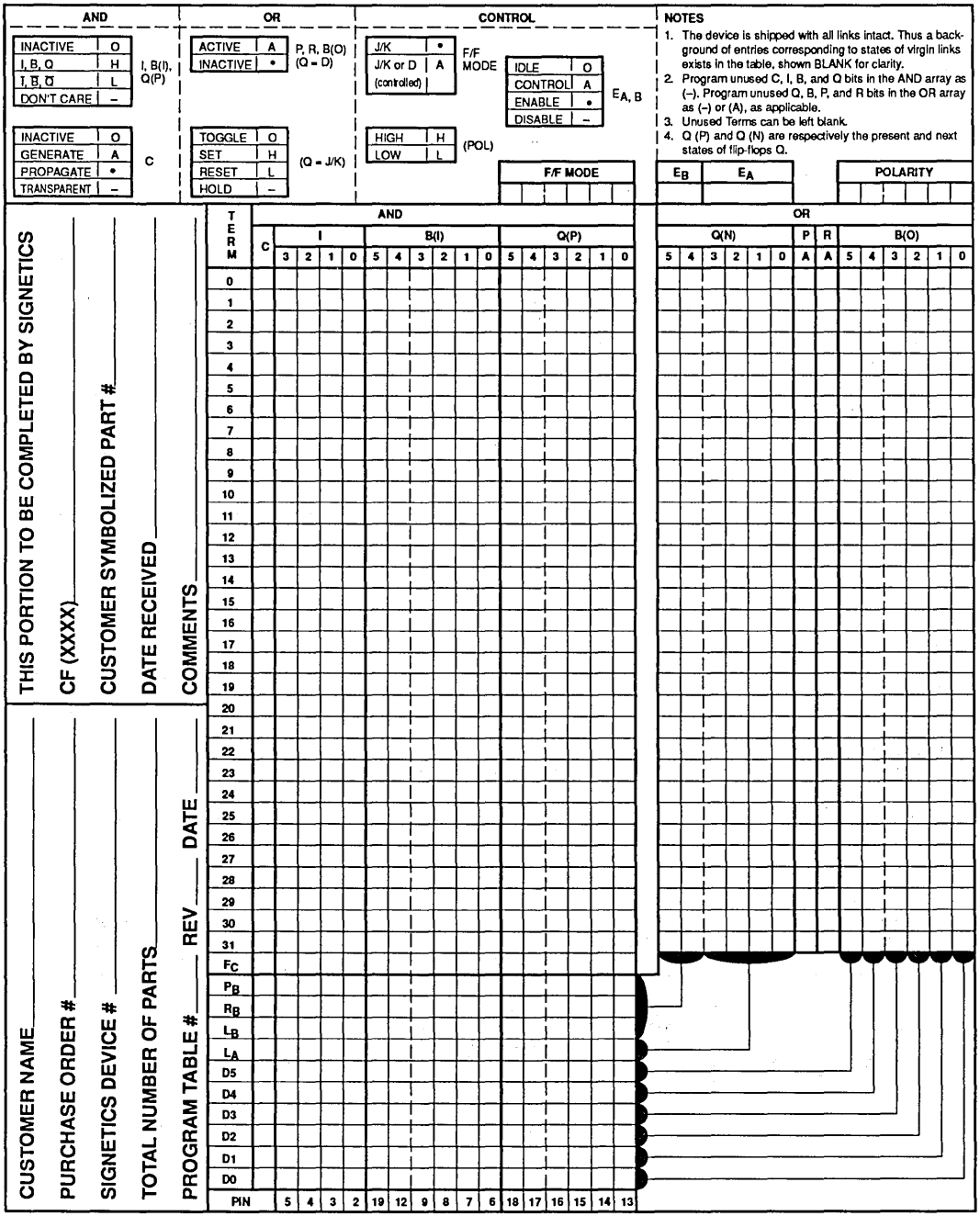
ACTION	CODE
ENABLE[4]	•

ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate $(T, F_C, L, P, R, D)_n$ will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates $(L, P, R, D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (16 × 45 × 12) PLS157

PROGRAM TABLE



Philips Components-Signetics

Document No.	853–1159
ECN No.	93255
Date of Issue	May 11, 1988
Status	Product Specification
Programmable Logic Devices	

PLS159A

Programmable logic sequencer ($16 \times 45 \times 12$)

DESCRIPTION

The PLS159A is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate F_C. It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates can drive bidirectional I/O lines (B), whose output polarity is individually programmable through a set of Ex-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the 32 AND gates can drive the J-K inputs of all flip-flops. There are 4 AND gates for the Asynchronous Preset/Reset functions.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

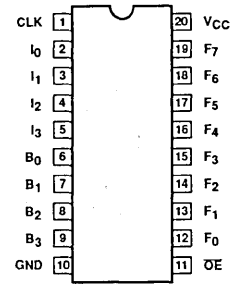
The PLS159A is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

FEATURES

- High-speed version of PLS159
- f_{MAX} = 18MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J-K, T, or D-type flip-flops
- Power-on reset feature on all flip-flops (F_n = 1)
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Input loading: −100μA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- 3-State outputs

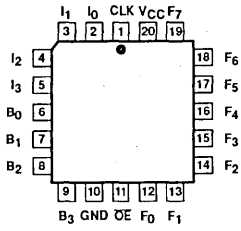
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



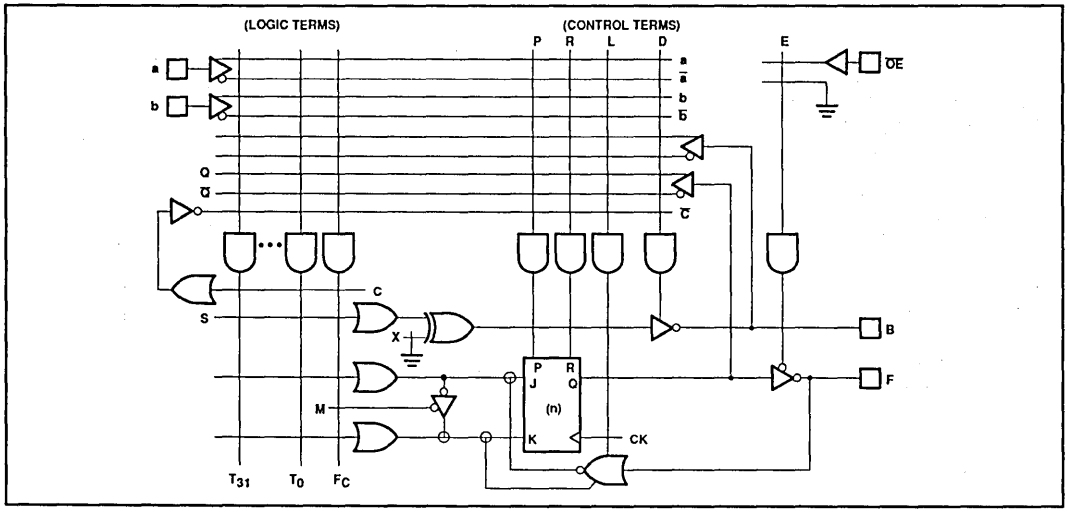
A = Plastic Leaded Chip Carrier

APPLICATIONS

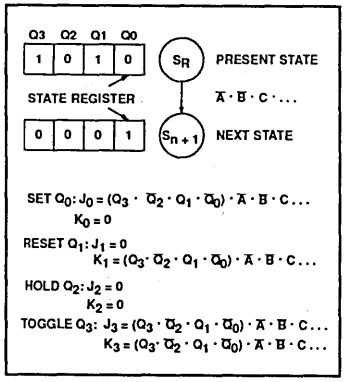
- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

Programmable logic sequencer (16 × 45 × 12) — PLS159A

FUNCTIONAL DIAGRAM



LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								HI-Z
L	X	X	L	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

NOTES:

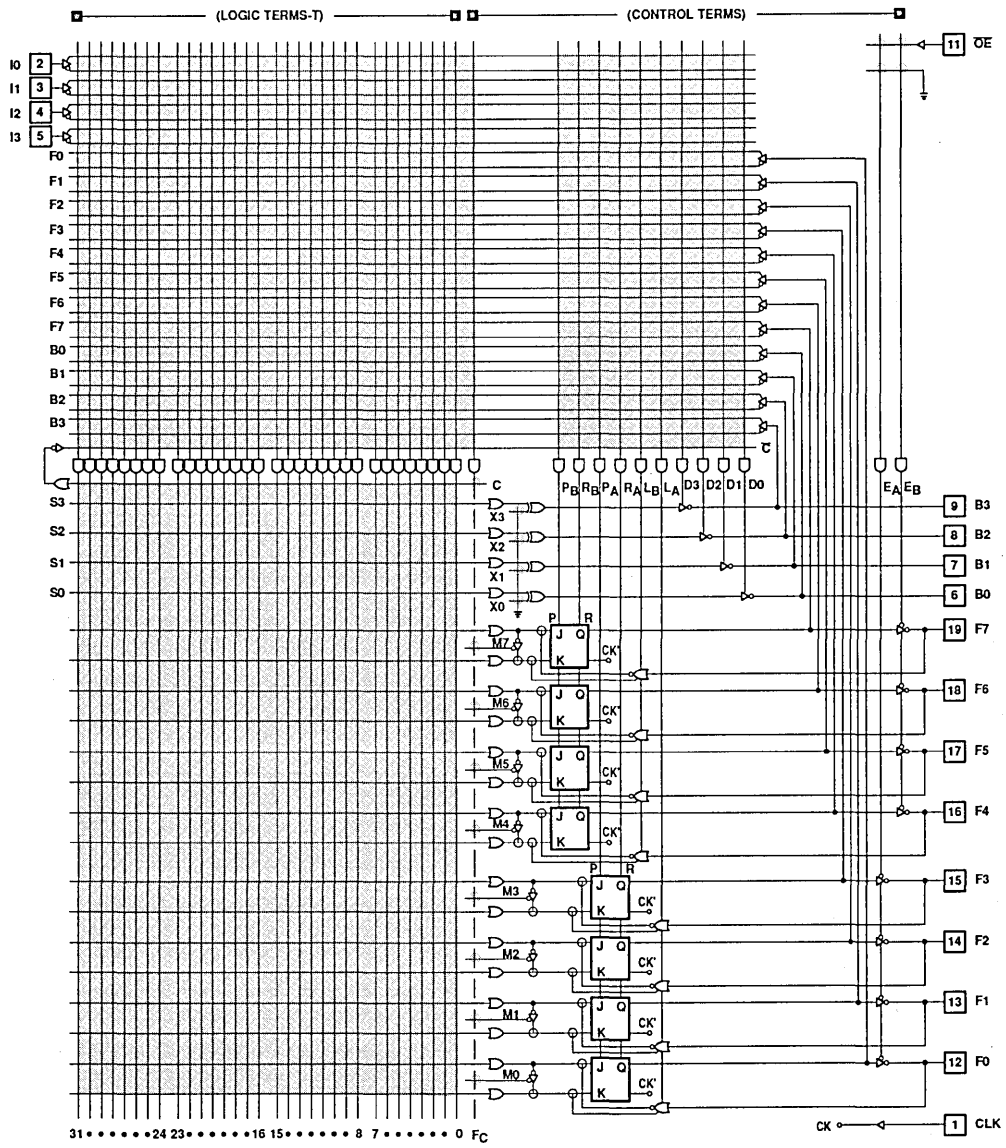
1. Positive Logic:
   $J\text{-}K = \overline{T}_0 + \overline{T}_1 + T_2 \ldots\ldots\ldots T_{31}$
   $T_n = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

CAUTION: PLS159A PROGRAMMING ALGORITHM

The programming voltage required to program the PLS159A is higher (17.5V) than that required to program the PLS159 (14.5V). Consequently, the PLS159 programming algorithm will not program the PLS159A. Please exercise caution when accessing programmer device codes to insure that the correct algorithm is used.

Programmable logic sequencer (16 × 45 × 12) PLS159A

LOGIC DIAGRAM



NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ▒ Programmable connection.

Programmable logic sequencer (16 × 45 × 12) PLS159A

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin Plastic DIP (300mil-wide)	PLS159AN
20-Pin Plastic Leaded Chip Carrier	PLS159AA

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low	I_{OL} = 10mA		0.35	0.5	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		–10	–100	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[4, 7]	V_{CC} = MAX, V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V		–1	–140	µA
I_{OS}	Short circuit[3, 5]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[6]	V_{CC} = MAX		150	190	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IH} applied to $\overline{OE}$.
5. Duration of short circuit should not exceed 1 second.
6. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
7. Leakage values are a combination of input and output leakage.

Programmable logic sequencer (16 × 45 × 12) PLS159A

AC ELECTRICAL CHARACTERISTICS

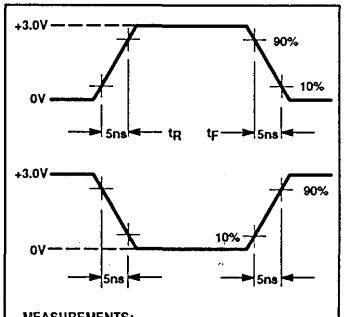
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	20	15		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	20	15		ns
t_{CKP}	Period	CK +	CK +	C_L = 30pF	55	45		ns
t_{PRH}	Preset/Reset pulse	(I,B) –	(I,B) +	C_L = 30pF	35	30		ns
Setup time[5]								
t_{IS1}	Input	(I,B) ±	CK +	C_L = 30pF	35	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	15	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I,B) ±	CK +	C_L = 30pF	55	45		ns
Hold time								
t_{IH1}	Input	(I,B) ±	CK +	C_L = 30pF	0	–5		ns
t_{IH2}	Input (through F_n)	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delay								
t_{CKO}	Clock	CK +	F ±	C_L = 30pF		15	20	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I,B) ±	B ±	C_L = 30pF		25	35	ns
t_{OE2}	Output enable[3]	(I,B) +	B ±	C_L = 30pF		20	30	ns
t_{OD2}	Output disable[3]	(I,B) –	B +	C_L = 5pF		20	30	ns
t_{PRO}	Preset/Reset	(I,B) +	F ±	C_L = 30pF		35	45	ns
t_{PPR}	Power–on/preset	V_{CC} +	F –	C_L = 30pF		0	10	ns

NOTES:

1. All typical values are at V_{CC} = 5V, T_A = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 75ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

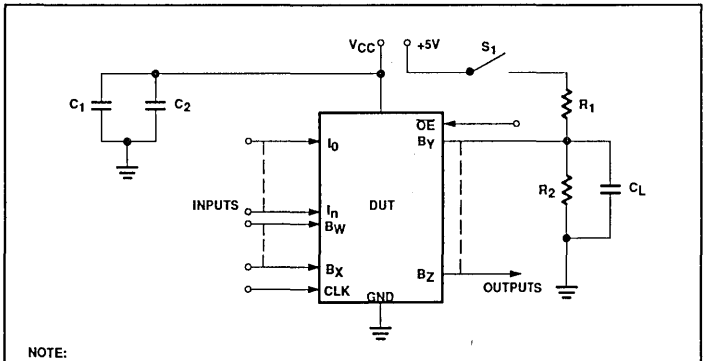
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

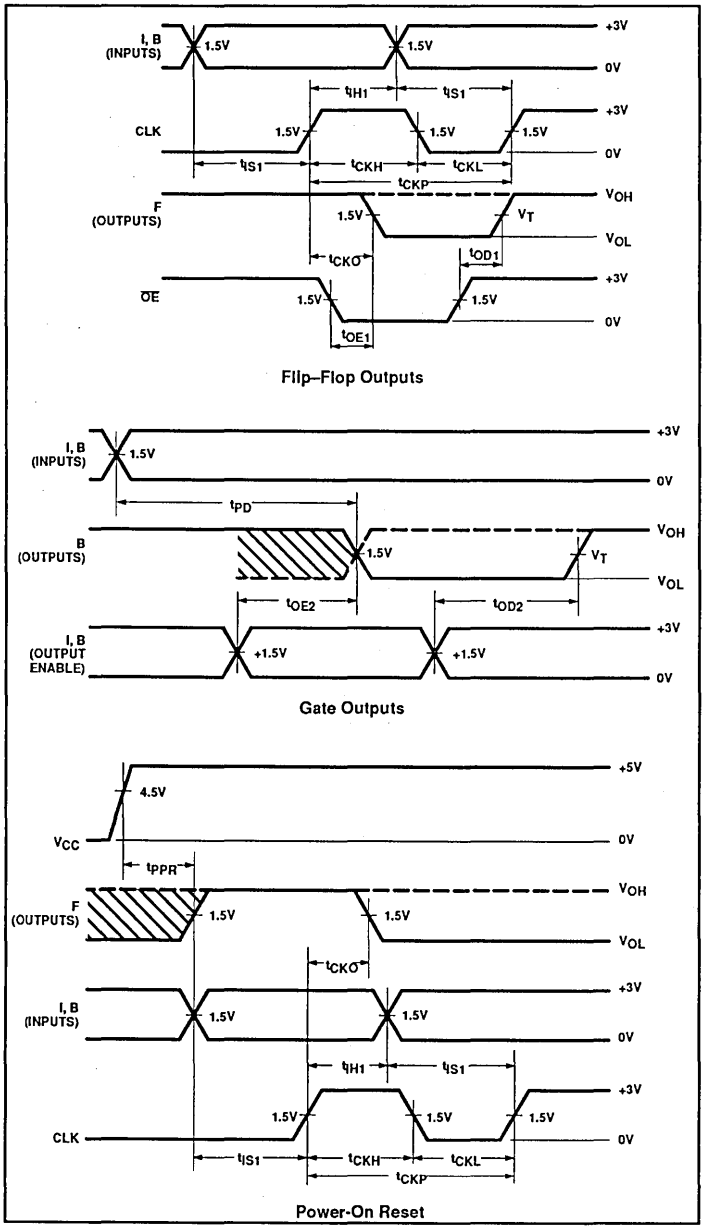
TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Programmable logic sequencer (16 × 45 × 12) PLS159A

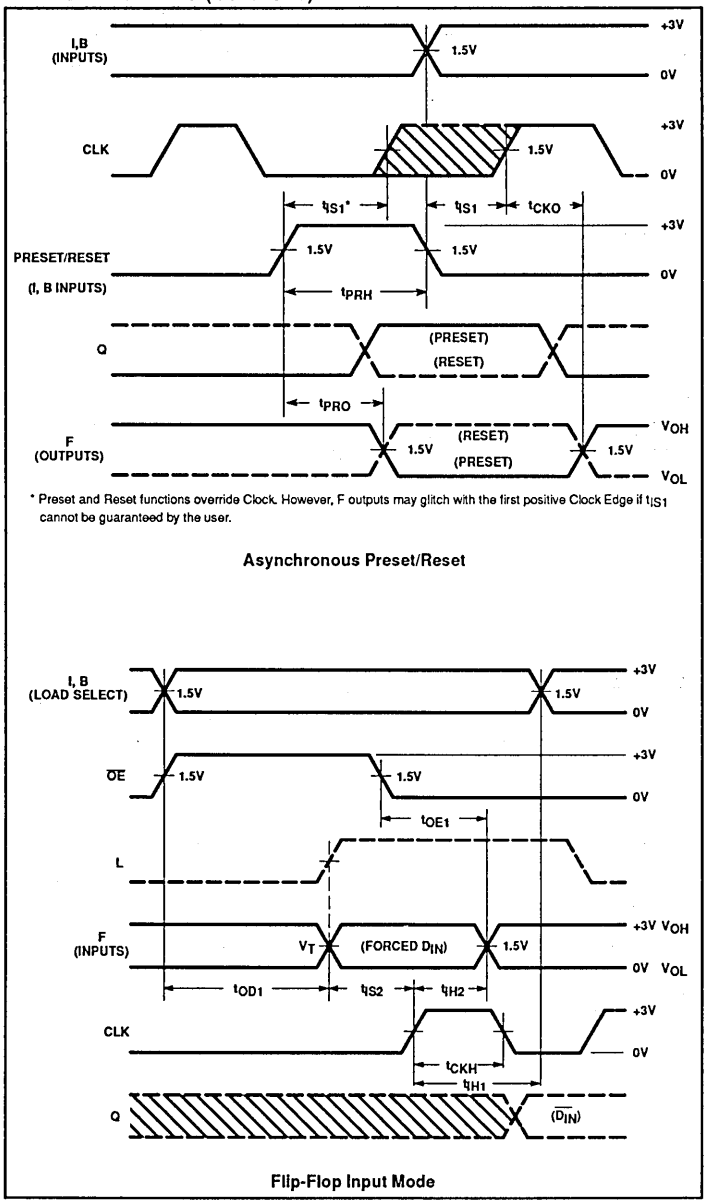
TIMING DIAGRAMS



TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

TIMING DIAGRAMS (Continued)



Programmable logic sequencer (16 × 45 × 12) — PLS159A

LOGIC PROGRAMMING

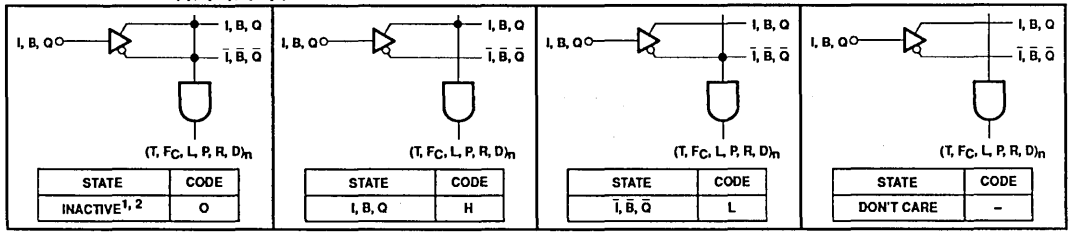
The PLS159A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE, Data I/O Corporation's ABEL™, and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS159A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

"AND" ARRAY – (I), (B), (Qp)



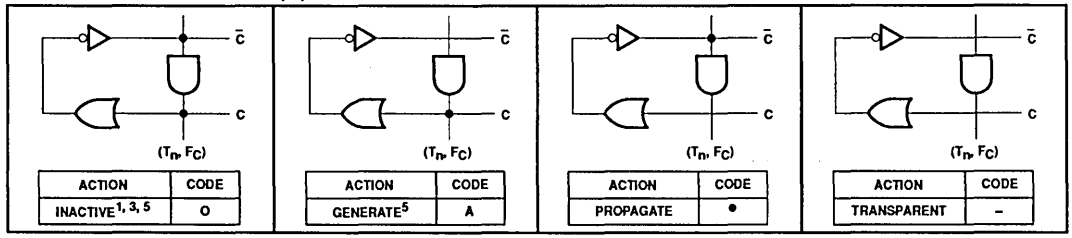
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
$\bar{I}, \bar{B}, \bar{Q}$	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



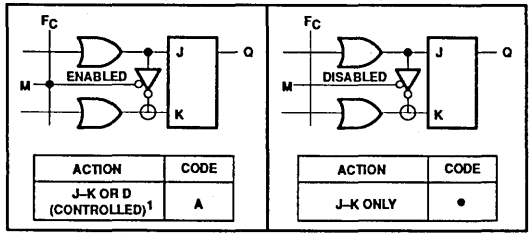
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

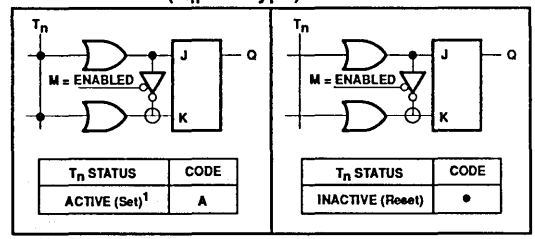
"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)[1]	A

ACTION	CODE
J–K ONLY	•

"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)[1]	A

T_n STATUS	CODE
INACTIVE (Reset)	•

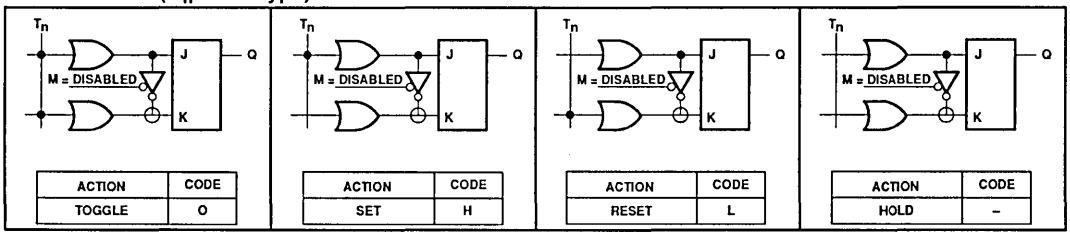
Notes on following page.

CAUTION:
THE PLS159A Programming Algorithm is different from the PLS159.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic sequencer (16 × 45 × 12) PLS159A

"OR" ARRAY – (Q_n = J-K Type)



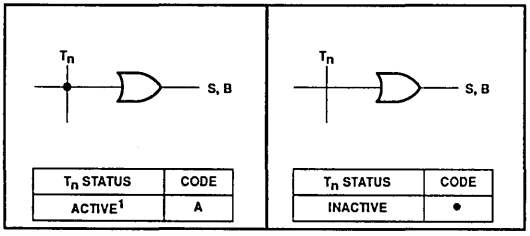
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

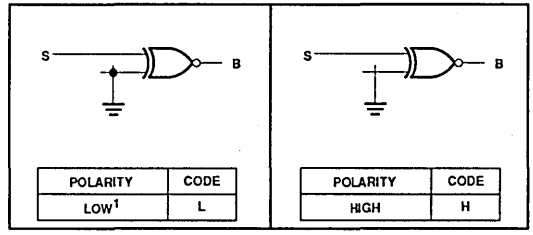
"OR" ARRAY – (S or B)



T_n STATUS	CODE
ACTIVE[1]	A

T_n STATUS	CODE
INACTIVE	•

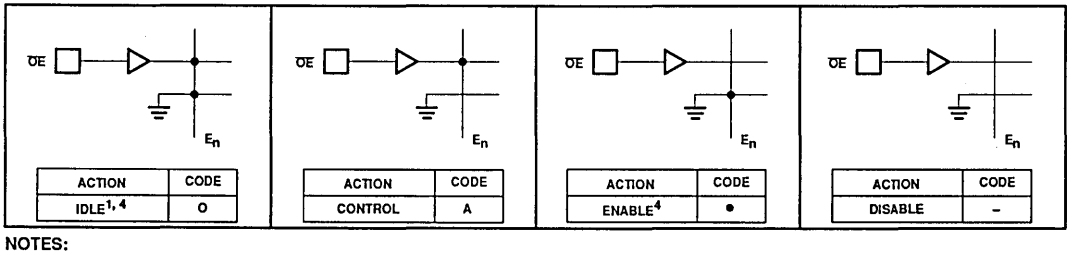
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW[1]	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1,4]	O

ACTION	CODE
CONTROL	A

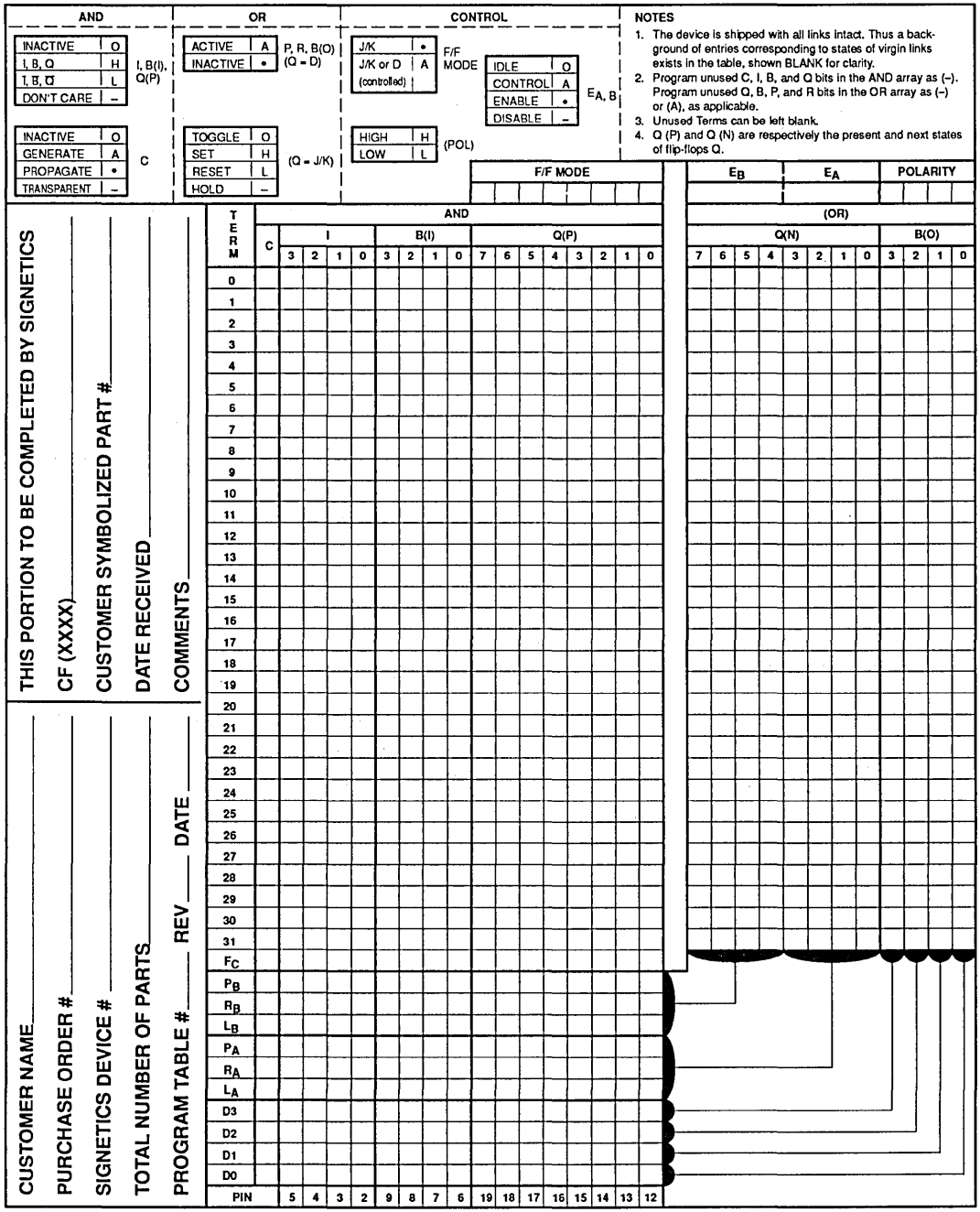
ACTION	CODE
ENABLE[4]	•

ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate $(T, F_C, L, P, R, D)_n$ will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates $(L, P, R, D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (16 × 45 × 12) PLS159A

FPLS PROGRAM TABLE



Document No.	853–0314
ECN No.	97885
Date of Issue	October 16, 1989
Status	Product Specification
Programmable Logic Devices	

PLS167/A

Programmable logic sequencers ($14 \times 48 \times 6$)

DESCRIPTION

The PLS167 and PLS167A are bipolar, Programmable Logic State machines of the Mealy type. The Programmable Logic Sequencers (PLS) contain logic AND/OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 8 Q_P, and 4 Q_F edge-triggered, clocked S/R flip-flops, with an asynchronous Preset Option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 14 external inputs, $I_{0\text{-}13}$, with 8 internal inputs, $P_{0\text{-}7}$, fed back from the State Register to form up to 48 transition terms (AND terms). In addition, P_0 and P_1 of the internal State Register are brought off-chip to allow extending the Output Register to 6 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-enable function, as an additional user programmable option.

Order codes are listed in the Ordering Information Table.

FEATURES

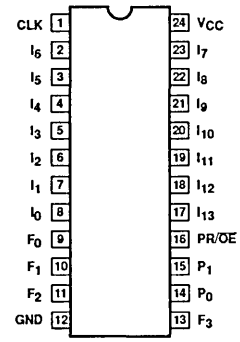
- PLS167
  - f_{MAX} = 13.9MHz
  - 20MHz clock rate
- PLS167A
  - f_{MAX} = 20MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 14 True/Complement buffered inputs
- 48 programmable AND gates
- 25 programmable OR gates
- 8-bit State Register
- 2-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- Power: 600mW (typ.)
- TTL compatible
- 3-State outputs
- Single +5V supply

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Security locking systems

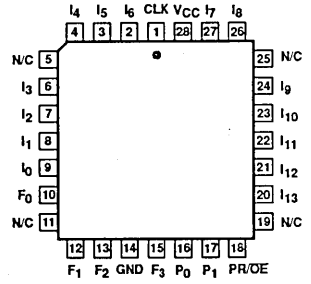
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

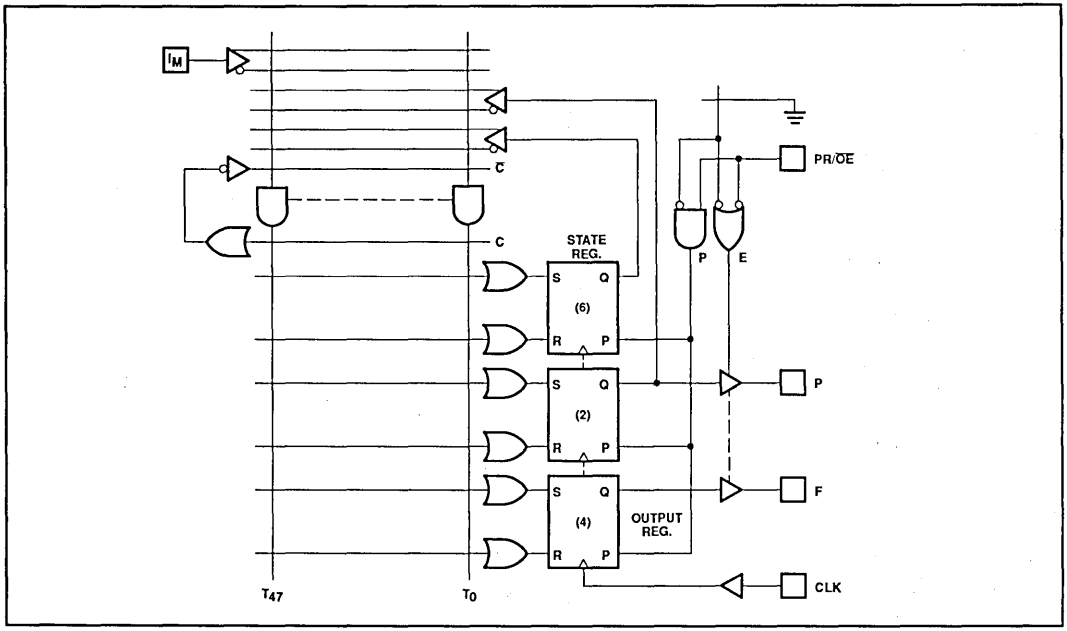
A Package



A = Plastic Leaded Chip Carrier

Programmable logic sequencers (14 × 48 × 6) — PLS167/A

FUNCTIONAL DIAGRAM

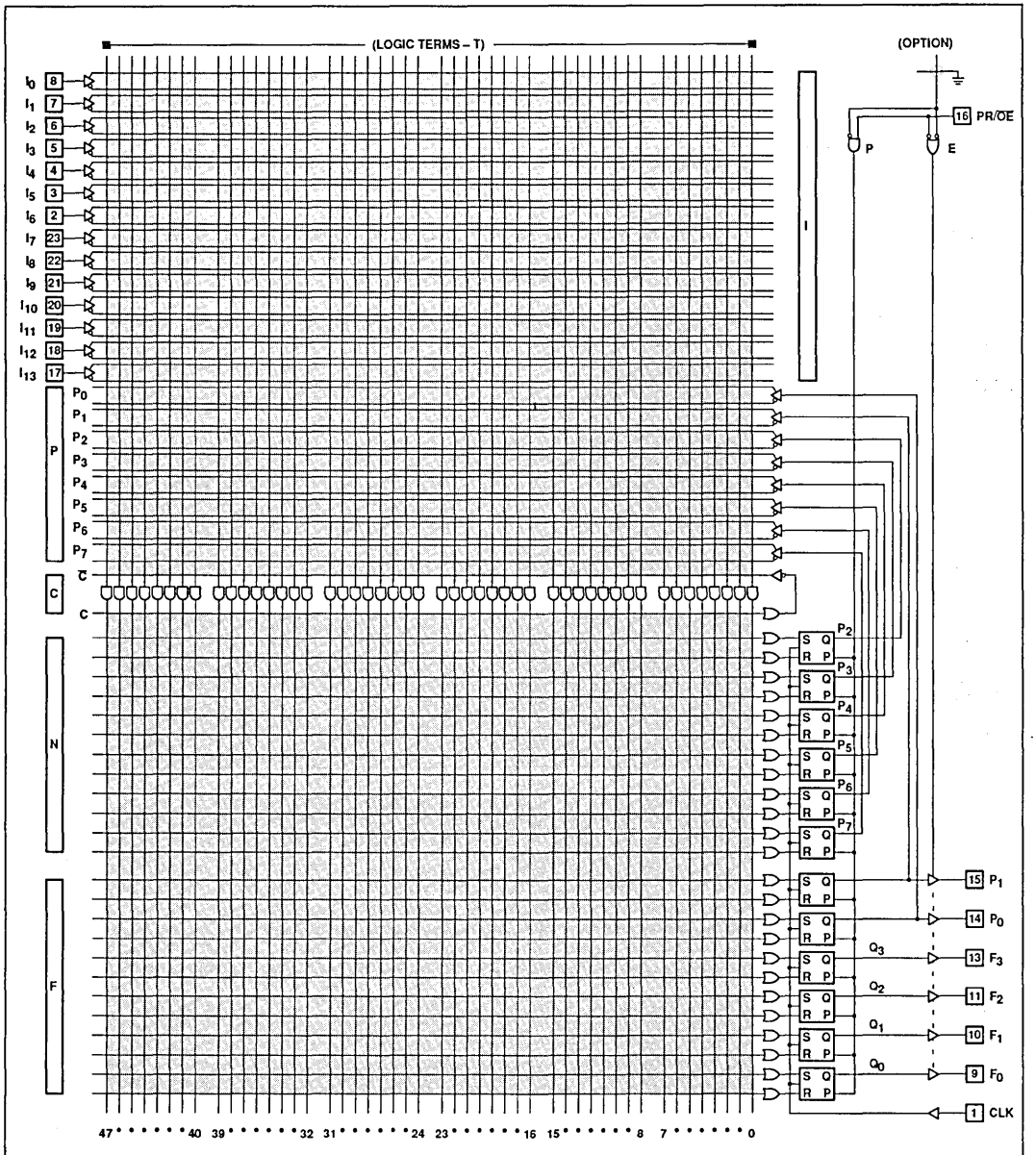


PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2 – 7 17 – 23	$I_1 - I_{13}$	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
8	I_0	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercised with standard TTL levels. When I_0 is held at +10V, device outputs F_{0-3} and P_{0-1} reflect the contents of State Register bits P_{2-7} (see Diagnostic Output Mode diagram). The contents of flip-flops P_{0-1} and F_{0-3} remain unaltered.	Active-High/Low
9 – 11 13	F_{0-3}	**Logic/Diagnostic Outputs:** Four device outputs which normally reflect the contents of Output Register bits Q_{0-3}, when enabled. When I_0 is held at +10V, $F_{0-3} = (P_{2-5})$.	Active-High
14 – 15	P_{0-1}	**Logic/Diagnostic Outputs:** Two register bits with shared function as least Significant State Register bits, or most significant Output Register bits. When I_0 is held at +10V, $P_{0-1} = (P_{6-7})$.	Active-High
16	PR/$\overline{OE}$	**Preset or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Preset:** Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and P_{0-7} and F_{0-3} are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an Output Enable function to all output buffers.	Active-Low (L)

Programmable logic sequencers (14 × 48 × 6) PLS167/A

LOGIC DIAGRAM



NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ◌ Programmable connection.

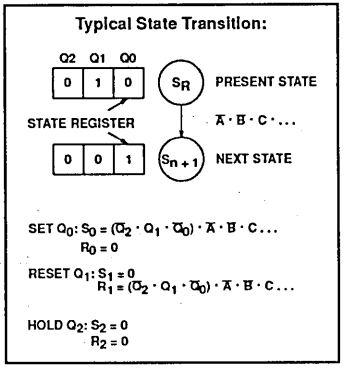
Programmable logic sequencers (14 × 48 × 6) — PLS167/A

TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I_0	CK	S	R	$Q_{P/F}$	F
	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q_n	$(Q_P)_n$
	L		X	X	X	X	Q_n	$(Q_F)_n$
		H	*	X	X	X	Q_n	Hi-Z
+5V		L	+10V	X	X	X	Q_n	$(Q_P)_n$
		L	X	X	X	X	Q_n	$(Q_F)_n$
		L	X	↑	L	L	Q_n	$(Q_F)_n$
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots T_{47}$
   $T_n = C(I_0\, I_1\, I_2 \ldots)\,(P_0\, P_1 \ldots P_7)$
2. Either Preset (Active-High) or $\overline{\text{Output Enable}}$ (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. R = S = High is an illegal input condition.
5. * = H or L or +10V.
6. X = Don't Care (≤5.5V)

LOGIC FUNCTION



VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. PR/$\overline{OE}$ option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array pre-programmed with a standard test pattern.
   NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Signetics qualified programming equipment.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic DIP (300mil-wide)	PLS167N, PLS167AN
28-Pin Plastic Leaded Chip Carrier	PLS167A, PLS167AA

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
V_{OH}	High[4]	V_{CC} = MIN I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
I_{IL}	Low (CK input)	V_{IN} = 0.45V		−50	−250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[5, 6]	V_{CC} = MAX V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		120	180	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5.0V V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IL} applied to $\overline{OE}$ and a logic high stored, or with V_{IH} applied to PR.
5. Measured with a programmed logic condition for which the output is at a low logic level, and V_{IL} applied to PR/$\overline{OE}$ Output sink current is supplied through a resistor to V_{CC}.
6. Measured with V_{IH} applied to PR/$\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.
8. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (14 × 48 × 6) — PLS167/A

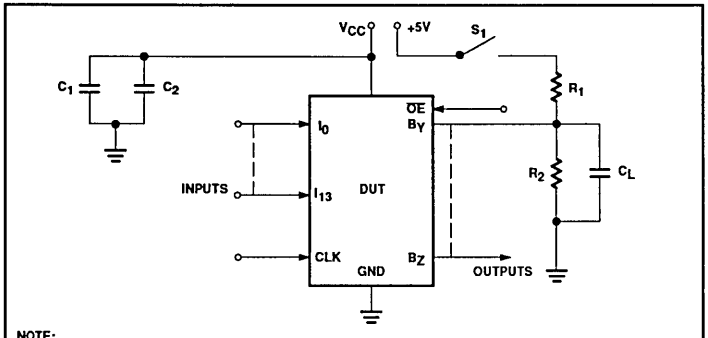
AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75°CV \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLS167			PLS167A			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width[3]										
t_{CKH}	Clock[2] High	CK +	CK –	25	15		20	15		ns
t_{CKL}	Clock Low	CK –	CK +	25	15		20	15		ns
t_{CKP}	Clock Period	CK +	CK +	50	30		40	30		ns
t_{PRH}	Preset pulse	PR +	PR –	25	15		25	15		ns
Setup time[3]										
$t_{IS1}A$	Input	Input ±	CK +	60			40			ns
$t_{IS1}B$	Input	Input ±	CK +	50			30			ns
$t_{IS1}C$	Input	Input ±	CK +	42			N/A			ns
$t_{IS2}A$	Input (through Complement Array)	Input ±	CK +	90			70			ns
$t_{IS2}B$	Input (through Complement Array)	Input	CK +	80			60			ns
$t_{IS2}C$	Input (through Complement Array)	Input	CK +	72			N/A			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		0	–10		ns
t_{PRS}	Preset	PR –	CK –	0	–10		0	–10		ns
Hold time										
t_{IH}	Input	CK +	Input ±	5	–10		0	–5		ns
Propagation delay										
t_{CKO}	Clock	CK +	Output ±		15	30		15	20	ns
t_{OE}	Output enable[4]	OE –	Output –		20	30		20	30	ns
t_{OD}	Output disable[4]	OE +	Output +		20	30		20	30	ns
t_{PR}	Preset	PR +	Output +		18	30		18	30	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10		0	10	ns
Frequency of operation[3]										
$f_{MAX}C$	Without Complement Array			13.9			20.0			MHz
$f_{MAX}C$	With Complement Array			9.8			12.5			MHz

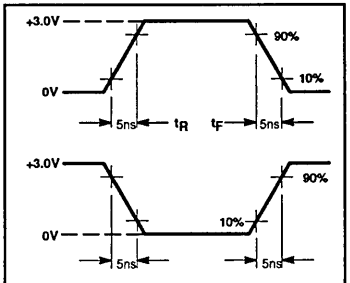
NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 30ns.
3. See "Speed vs. OR Loading" diagrams.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

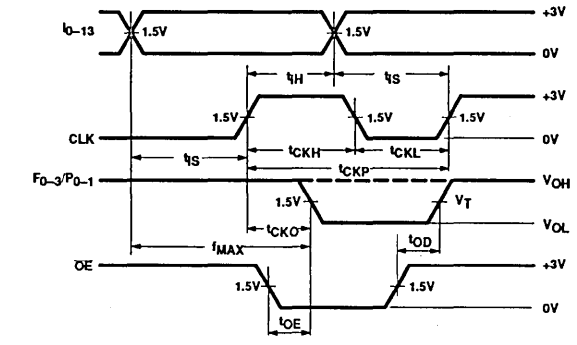
VOLTAGE WAVEFORMS



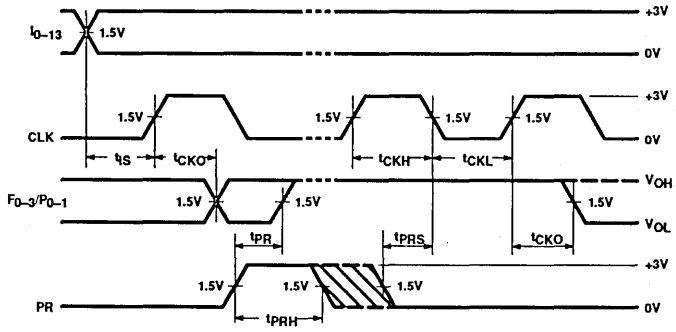
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

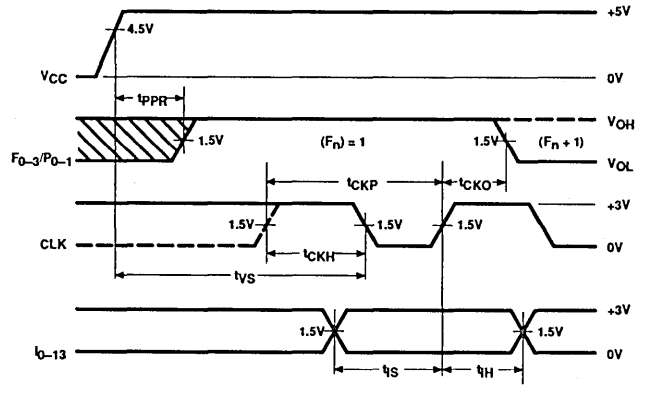
TIMING DIAGRAMS



Flip-Flop Outputs



Asynchronous Preset

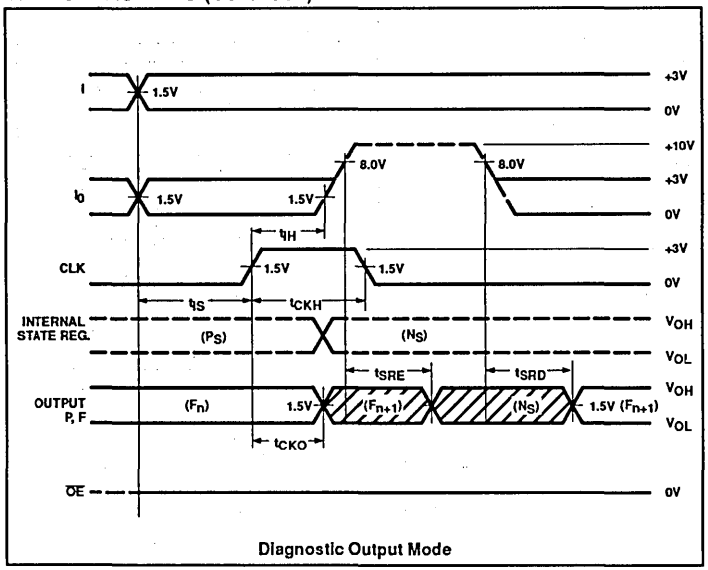


Power-On Preset

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of clock preceding first reliable clock pulse.
t_{PRS}	Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with PR/$\overline{OE}$ Low).
t_{OE}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{SRE}	Delay between input I_0 transition to Diagnostic mode and when the outputs reflect the contents of the State Register.
t_{SRD}	Delay between input I_0 transition to Logic mode and when the outputs reflect the contents of the Output Register.
t_{PR}	Delay between positive transition of Preset and when outputs become valid at "1".
t_{PPR}	Delay between V_{CC} (after power-on) and when outputs become preset at "1".
t_{PRH}	Width of preset input pulse.
f_{MAX}	Minimum guaranteed operating frequency.

TIMING DIAGRAMS (Continued)



Diagnostic Output Mode

SPEED VS. "OR" LOADING

The maximum frequency at which the PLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms T_n used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects t_{IS}, due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of t_{IS1} with the number of terms connected per OR.

The PLS167 AC electrical characteristics contain three limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 1). The first, t_{IS1A} is guaranteed for a device with 48 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 32 terms connected to any OR line. And t_{IS1C} is guranteed for a device with 24 terms conntected to any OR line.

The three other entries in the AC table, t_{IS2} A, B, and C are corresponding 48, 32, and 24 term limits when using the on-chip Complement Array.

The PLS167A AC electrical characteristics contain two limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 2). The first, t_{IS1A} is guaranteed for a device with 24 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 16 terms connected to any OR line.

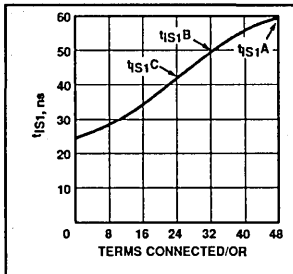


Figure 1. PLS167 t_{IS1} vs. Terms/OR Connected

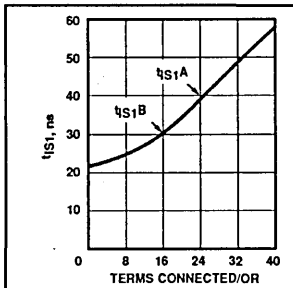


Figure 2. PLS167A t_{IS1} vs. Terms/OR Connected

The two other entries in the AC table, t_{IS2} A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of t_{IS} for a given application can be determined by identifying the OR line with the maximum number of T_n connections. This can be done by referring to the interconnect pattern in the PLS logic diagram, typically illustrated in Figure 3, or by counting the maximum number of "H" *or* "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 or 2 will yield the worst case t_{IS} and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

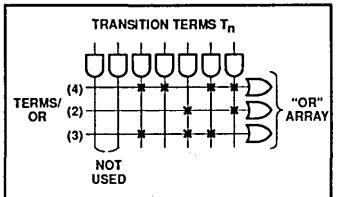


Figure 3. Typical OR Array Interconnect Pattern

Programmable logic sequencers (14 × 48 × 6) — PLS167/A

LOGIC PROGRAMMING

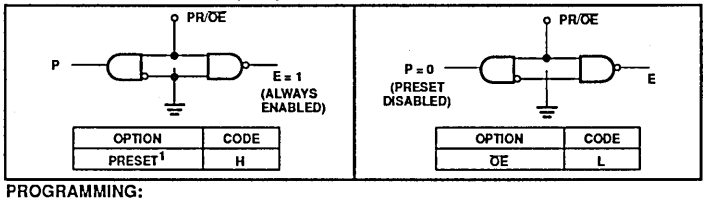
The PLS167/A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE design software package. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLS167/A architecture.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS167/A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PRESET/$\overline{OE}$ OPTION – (P/E)



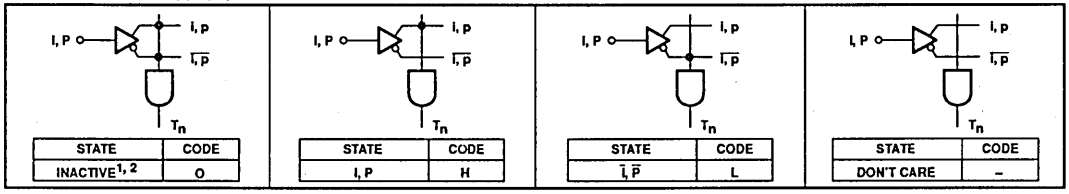
OPTION	CODE
PRESET[1]	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING:

The PS167/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY – (I), (P)



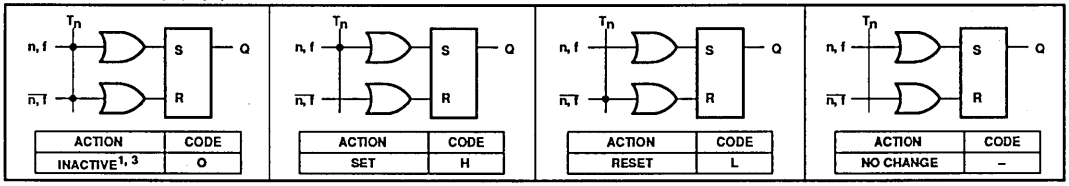
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I, P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



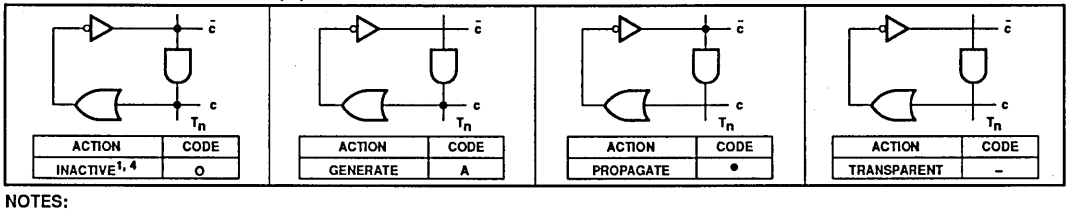
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I, P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable logic sequencers (14 × 48 × 6) PLS167/A

PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
SIGNETICS DEVICE # CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND			OR		
INACTIVE	0	C_n	INACTIVE	0	Ns, Fr
GENERATE	A		SET	H	
PROPAGATE	•		RESET	L	
TRANSPARENT	—		NO CHANGE	—	
INACTIVE	0	I_m, P_s	OPTION		
I, P	H		PRESET	H	P/E
$\overline{I}$, $\overline{P}$	L		$\overline{OE}$	L	
DON'T CARE	—				

TERM	C_n	AND: INPUT (I_m) 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) 7 6 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (Ns) 7 6 5 4 3 2 1 0	OUTPUT (Fr) 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		17 18 19 20 21 22 23 2 3 4 5 6 7 8				15 14 13 11 10 9
VARIABLE NAME						

NOTES:
1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

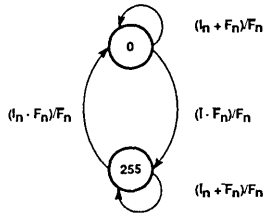
Programmable logic sequencers (14 × 48 × 6) PLS167/A

TEST ARRAY

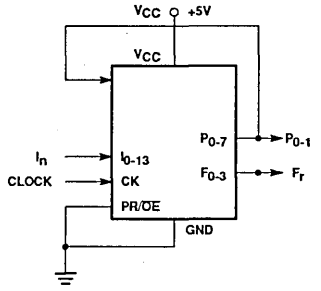
The PLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the PLS and applying the proper input sequence to $I_{0\text{-}13}$ as shown in the test circuit timing diagram.



State Diagram

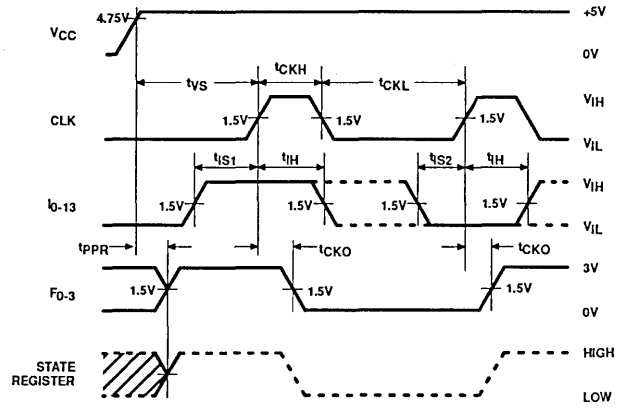


PLS Under Test

TERM	C	AND INPUT (Im) 13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 7	6	5	4	3	2	1	0	OPTION (P/E) H; OR NEXT STATE (Ns) 7	6	5	4	3	2	1	0	OUTPUT (Fr) 3	2	1	0
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any Signetic's qualified programming equipment.



Test Circuit Timing Diagram

TERM	C	AND INPUT (Im) 13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 7	6	5	4	3	2	1	0	OPTION (P/E) H; OR NEXT STATE (Ns) 7	6	5	4	3	2	1	0	OUTPUT (Fr) 5	4	3	2	1	0
48	–	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	–	–	–	–	–	–	–	–	–	–	–	–	–	–
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Test Array Deleted

Document No.	853–0322
ECN No.	97853
Date of Issue	October 11, 1989
Status	Product Specification
Programmable Logic Devices	

PLS168/A

Programmable logic sequencers ($12 \times 48 \times 8$)

DESCRIPTION

The PLS168 and the PLS168A are bipolar, Programmable Logic State machines of the Mealy type. They contain logic AND/OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 10 Q_P, and 4 Q_F edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset Option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 12 external inputs, $I_{0\text{-}11}$, with 10 internal inputs, $P_{0\text{-}9}$, fed back from the State Register to form up to 48 transition terms (AND terms). In addition, P_0 - P_3 of the internal State Register are brought off-chip to allow extending the Output Register to 8 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-enable function, as an additional user programmable option.

Order codes are listed in the Ordering Information Table.

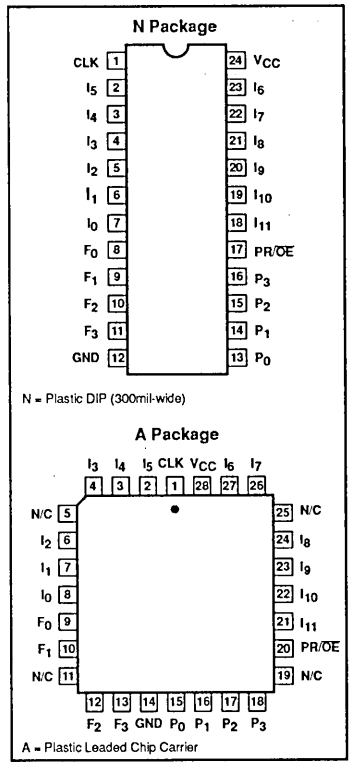
FEATURES

- PLS168
  - f_{MAX} = 13.9MHz
  - 20MHz clock rate
- PLS168A
  - f_{MAX} = 20MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 12 True/Complement buffered inputs
- 48 programmable AND gates
- 29 programmable OR gates
- 10-bit State Register
- 4-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- Power: 600mW (typ.)
- TTL compatible
- 3-State outputs
- Single +5V supply

APPLICATIONS

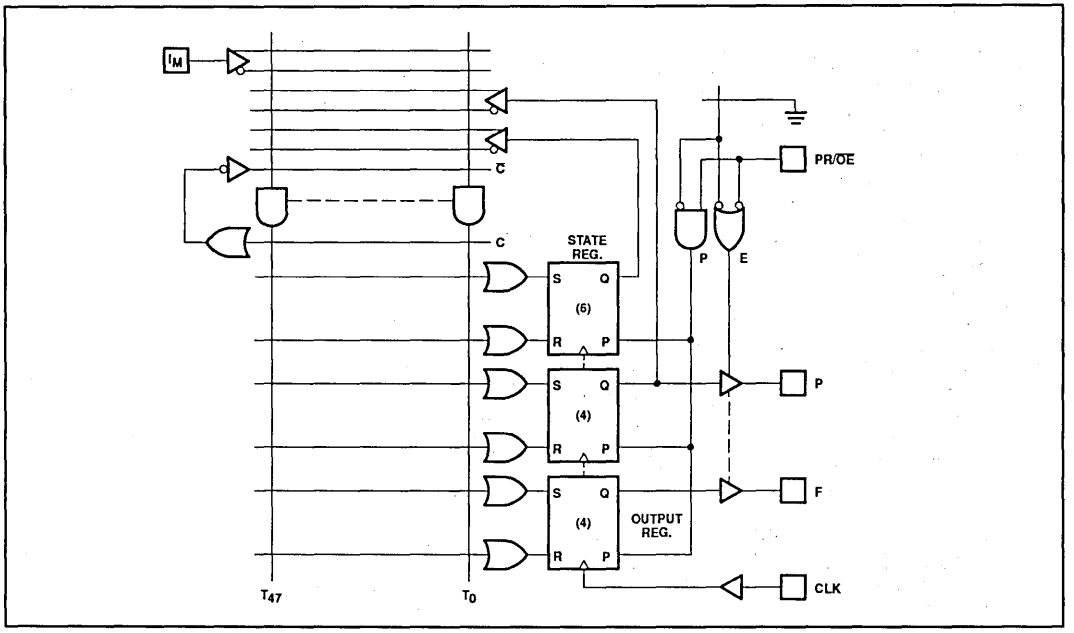
- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Security locking systems
- Counters
- Shift registers

PIN CONFIGURATIONS



Programmable logic sequencers (12 × 48 × 8) — PLS168/A

FUNCTIONAL DIAGRAM

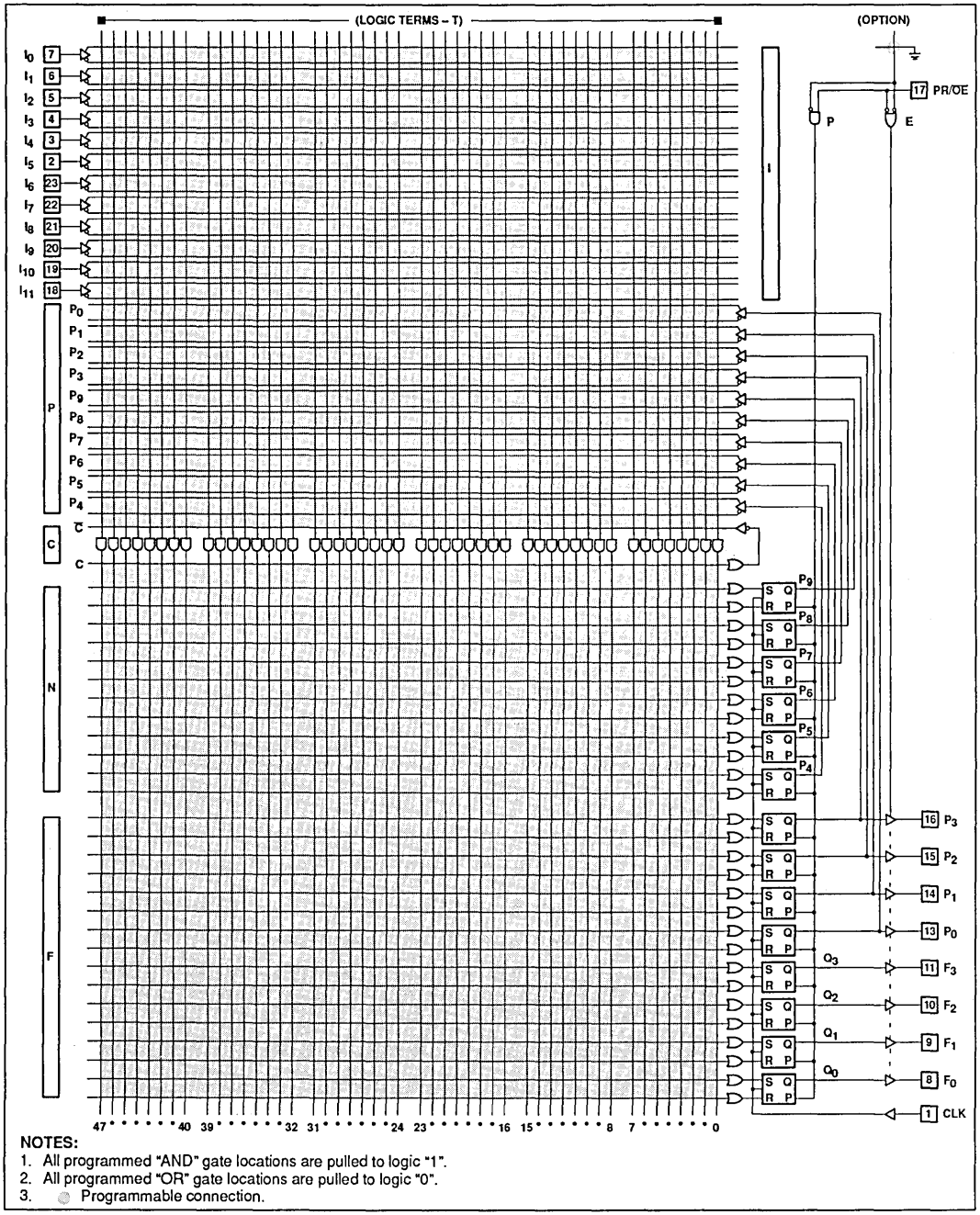


PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2 – 6 18 – 23	$I_1 - I_{11}$	**Logic Inputs:** The 11 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
7	I_0	**Logic/Diagnostic Input:** A 12th external logic input to the AND array, as above, when exercised with standard TTL levels. When I_0 is held at +10V, device outputs $F_2 - F_3$ and $P_0 - P_3$ reflect the contents of State Register bits P_{4-9} (see Diagnostic Output Mode diagram). The contents of flip-flops P_{0-1} and F_{0-3} remain unaltered.	Active-High/Low
13 – 16	P_{0-3}	**Logic/Diagnostic Outputs:** Four device outputs which normally reflect the contents of State Register bits P_{0-3}. When I_0 is held at +10V these pins reflect ($P_6 - P_9$).	Active-High
10 – 11	$F_2 - F_3$	**Logic/Diagnostic Outputs:** Two register bits ($F_2 - F_3$) which reflect Output register bits ($Q_2 - Q_3$). When I_0 is held at +10V, these pins reflect ($P_4 - P_5$).	Active-High
17	PR/$\overline{OE}$	**Preset or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Preset:** Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and P_{0-9} and F_{0-3} are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.	Active-High (H)
		• $\overline{\text{Output Enable}}$: Provides an Output Enable function to all output buffers.	Active-Low (L)
8, 9	$F_0 - F_1$	**Logic Output:** Two device outputs which reflect Output Registers $Q_0 - Q_1$. When I_0 is haeld at +10V, $F_0 - F_1$ = Logic "1".	

Programmable logic sequencers (12 × 48 × 8) PLS168/A

LOGIC DIAGRAM



NOTES:
1. All programmed "AND" gate locations are pulled to logic "1".
2. All programmed "OR" gate locations are pulled to logic "0".
3. ◌ Programmable connection.

Programmable logic sequencers (12 × 48 × 8) — PLS168/A

TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I_0	CLK	S	R	$Q_{P/F}$	F
	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q_n	$(Q_P)_n$
	L		X	X	X	X	Q_n	$(Q_F)_n$
		H	*	X	X	X	Q_n	Hi-Z
+5V		L	+10V	X	X	X	Q_n	$(Q_P)_n$
		L	X	X	X	X	Q_n	$(Q_F)_n$
		L	X	↑	L	L	Q_n	$(Q_F)_n$
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots + T_{47}$
   $T_n = C(I_0\, I_1\, I_2 \ldots)(P_0\, P_1 \ldots P_9)$
2. Either Preset (Active-High) or $\overline{\text{Output Enable}}$ (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. R = S = High is an illegal input condition.
5. * = H or L or +10V.
6. X = Don't Care (≤5.5V)

ORDERING INFORMATION

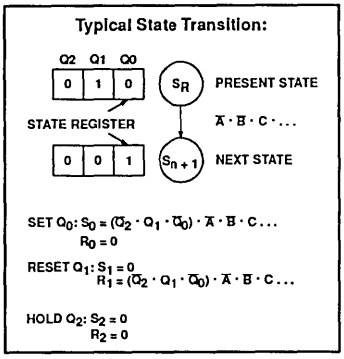
DESCRIPTION	ORDER CODE
24-Pin Plastic DIP (300mil-wide)	PLS168N, PLS168AN
28-Pin Plastic Leaded Chip Carrier	PLS168A, PLS168AA

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION



VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. PR/$\overline{OE}$ option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be docked via a Test Array pre-programmed with a standard test pattern.
   NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Signetics qualified programming equipment.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High[4]	I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
I_{IL}	Low (CLK input)	V_{IN} = 0.45V		−50	−250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[6]	V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		120	180	mA
Capacitance[6]						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IL} applied to $\overline{OE}$ and a logic high stored, or with V_{IH} applied to PR.
5. Measured with a programmed logic condition for which the output is at a low logic level, and V_{IL} applied to PR/$\overline{OE}$ Output sink current is supplied through a resistor to V_{CC}.
6. Measured with V_{IH} applied to PR/$\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.
8. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (12 × 48 × 8) PLS168/A

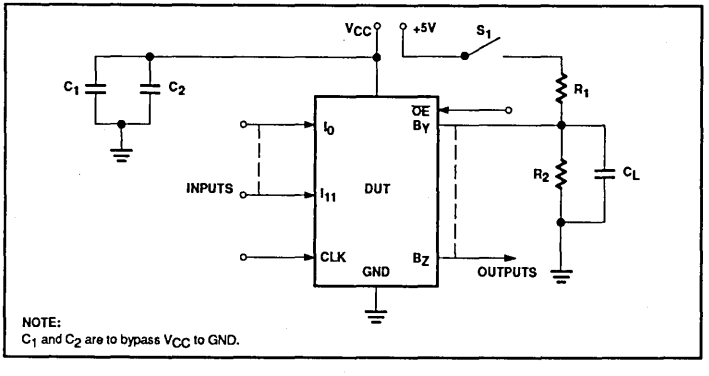
AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75°CV \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLS168			PLS168A			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width[3]										
t_{CKH}	Clock[2] High	CK +	CK –	25	15		20	15		ns
t_{CKL}	Clock Low	CK –	CK +	25	15		20	15		ns
t_{CKP}	Clock Period	CK +	CK +	50	30		40	30		ns
t_{PRH}	Preset pulse	PR +	PR –	25	15		25	15		ns
Setup time[3]										
$t_{IS1}A$	Input	Input ±	CK +	60			40			ns
$t_{IS1}B$	Input	Input ±	CK +	50			30			ns
$t_{IS1}C$	Input	Input ±	CK +	42			N/A			ns
$t_{IS2}A$	Input (through Complement Array)	Input ±	CK +	90			70			ns
$t_{IS2}B$	Input (through Complement Array)	Input	CK +	80			60			ns
$t_{IS2}C$	Input (through Complement Array)	Input	CK +	72			N/A			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		0	–10		ns
t_{PRS}	Preset	PR –	CK –	0	–10		0	–10		ns
Hold time										
t_{IH}	Input	CK +	Input ±	5	–10		5	–10		ns
Propagation delay										
t_{CKO}	Clock	CK +	Output ±		15	30		15	20	ns
t_{OE}	Output enable[4]	OE –	Output –		20	30		20	30	ns
t_{OD}	Output disable[4]	OE +	Output +		20	30		20	30	ns
t_{PR}	Preset	PR +	Output +		18	30		18	30	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10		0	10	ns
Frequency of operation[3]										
$f_{MAX}C$	Without Complement Array			13.9			20.0			MHz
$f_{MAX}C$	With Complement Array			9.8			12.5			MHz

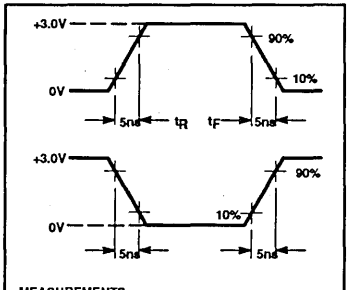
NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 30ns.
3. See "Speed vs. OR Loading" diagrams.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} – 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS

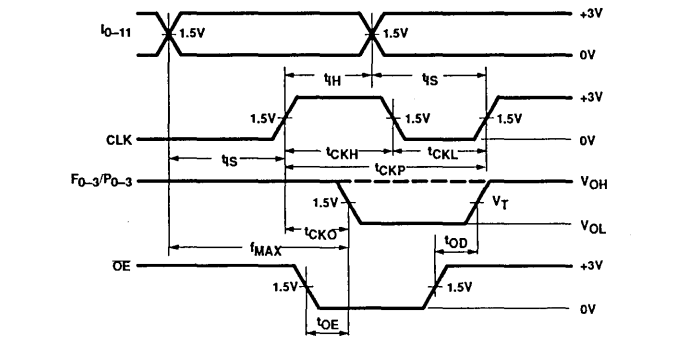


MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

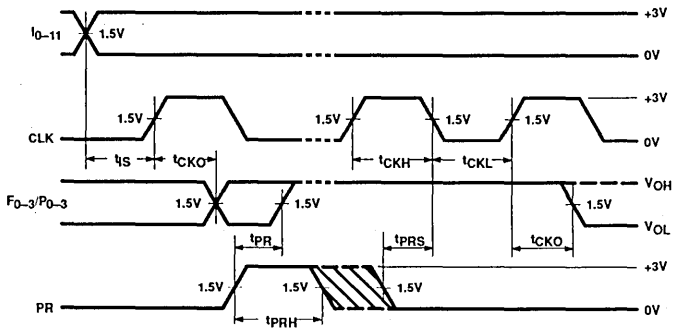
Input Pulses

Programmable logic sequencers (12 × 48 × 8) — PLS168/A

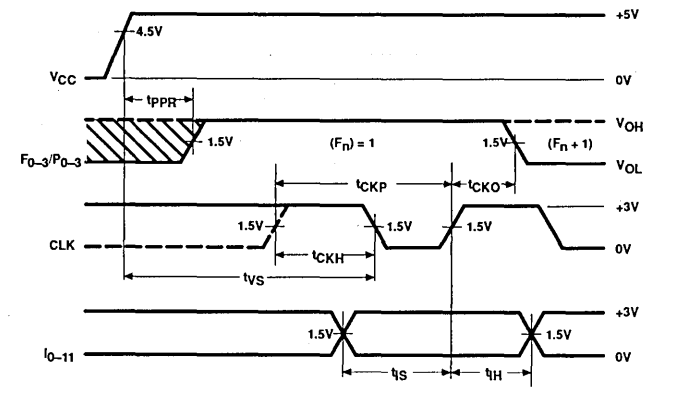
TIMING DIAGRAMS



Flip-Flop Outputs



Asynchronous Preset



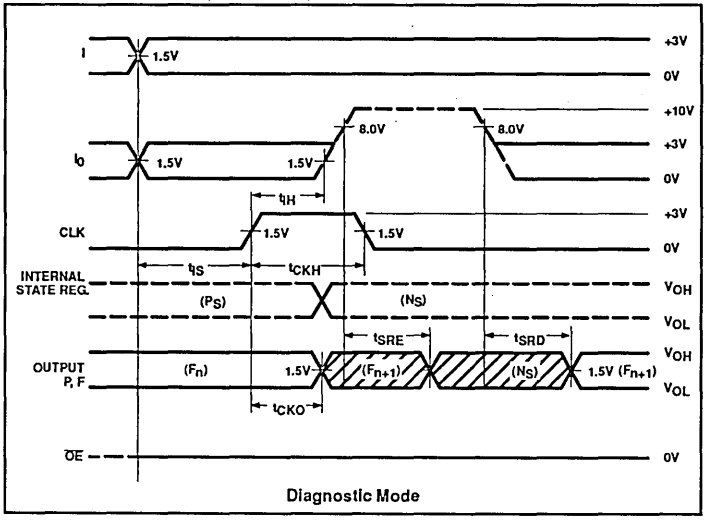
Power-On Preset

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of clock preceding first reliable clock pulse.
t_{PRS}	Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with PR/$\overline{OE}$ Low).
t_{OE}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{SRE}	Delay between input I_0 transition to Diagnostic mode and when the outputs reflect the contents of the State Register.
t_{SRD}	Delay between input I_0 transition to Logic mode and when the outputs reflect the contents of the Output Register.
t_{PR}	Delay between positive transition of Preset and when outputs become valid at "1".
t_{PPR}	Delay between V_{CC} (after power-on) and when outputs become preset at "1".
t_{PRH}	Width of preset input pulse.
f_{MAX}	Minimum guaranteed operating frequency.

Programmable logic sequencers (12 × 48 × 8) PLS168/A

TIMING DIAGRAMS (Continued)



Diagnostic Mode

SPEED VS. "OR" LOADING

The maximum frequency at which the PLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms T_n used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects t_{IS}, due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of t_{IS1} with the number of terms connected per OR.

The PLS168 AC electrical characteristics contain three limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 1). The first, t_{IS1A} is guaranteed for a device with 48 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 32 terms connected to any OR line. And t_{IS1C} is guranteed for a device with 24 terms conntected to any OR line.

The three other entries in the AC table, t_{IS2} A, B, and C are corresponding 48, 32, and 24 term limits when using the on-chip Complement Array.

The PLS168A AC electrical characteristics contain two limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 2). The first, t_{IS1A} is guaranteed for a device with 24 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 16 terms connected to any OR line.

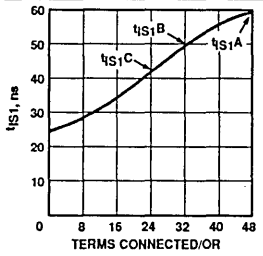


Figure 1. PLS168 t_{IS1} vs. Terms/OR Connected

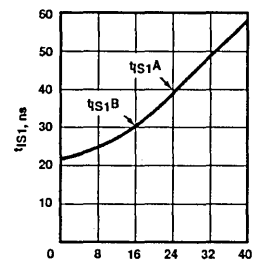


Figure 2. PLS168A t_{IS1} vs. Terms/OR Connected

The two other entries in the AC table, t_{IS2} A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of t_{IS} for a given application can be determined by identifying the OR line with the maximum number of T_n connections. This can be done by referring to the interconnect pattern in the PLS logic diagram, typically illustrated in Figure 3, or by counting the maximum number of "H" *or* "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 or 2 will yield the worst case t_{IS} and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

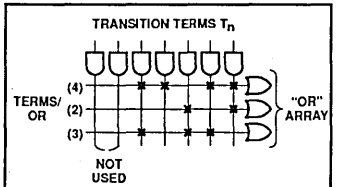


Figure 3. Typical OR Array Interconnect Pattern

Programmable logic sequencers (12 × 48 × 8) — PLS168/A

LOGIC PROGRAMMING

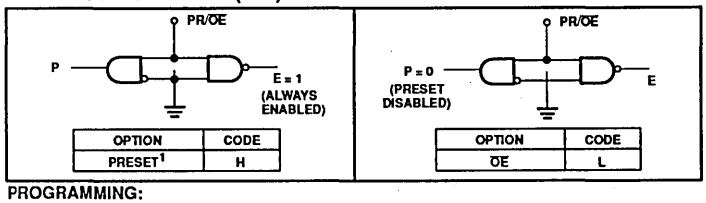
The PLS168/A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE design software package. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLS168/A architecture.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS168/A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PRESET/$\overline{OE}$ OPTION – (P/E)



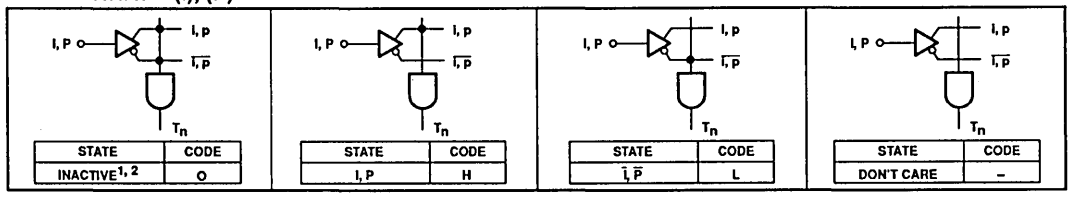
OPTION	CODE
PRESET[1]	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING:

The PLS168/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY – (I), (P)



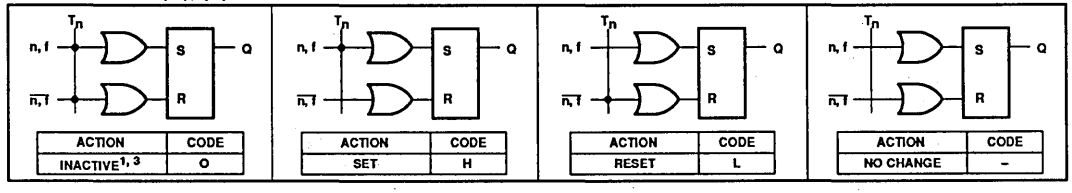
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I}$, $\overline{P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



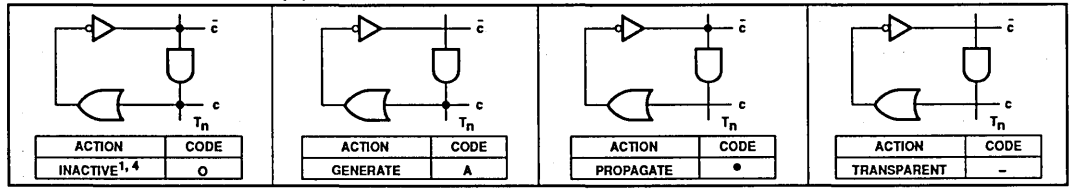
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I, P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable logic sequencers (12 × 48 × 8) PLS168/A

PROGRAM TABLE

CUSTOMER NAME ____________
PURCHASE ORDER # ____________
SIGNETICS DEVICE # CF (XXXX) ____________
CUSTOMER SYMBOLIZED PART # ____________
TOTAL NUMBER OF PARTS ____________
PROGRAM TABLE ____________
REV ____________
DATE ____________

PROGRAM TABLE ENTRIES

AND

INACTIVE	0	C_n
GENERATE	A	
PROPAGATE	•	
TRANSPARENT	—	

INACTIVE	0	I_m, P_s
I, P	H	
$\bar{I}$, $\bar{P}$	L	
DON'T CARE	—	

OR

INACTIVE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	—	

OPTION

PRESET	H	P/E
$\overline{OE}$	L	

		AND																							OPTION (P/E) OR													
TERM	C_n	INPUT (I_m)												PRESENT STATE (P_s)										REMARKS	NEXT STATE (N_s)										OUTPUT (F_r)			
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		9	8	7	6	5	4	3	2	1	0	3	2	1	0
0																																						
1																																						
2																																						
3																																						
4																																						
5																																						
6																																						
7																																						
8																																						
9																																						
10																																						
11																																						
12																																						
13																																						
14																																						
15																																						
16																																						
17																																						
18																																						
19																																						
20																																						
21																																						
22																																						
23																																						
24																																						
25																																						
26																																						
27																																						
28																																						
29																																						
30																																						
31																																						
32																																						
33																																						
34																																						
35																																						
36																																						
37																																						
38																																						
39																																						
40																																						
41																																						
42																																						
43																																						
44																																						
45																																						
46																																						
47																																						
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																		16	15	14	13	11	10	9	8
VARIABLE NAME																																						

NOTES:
1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

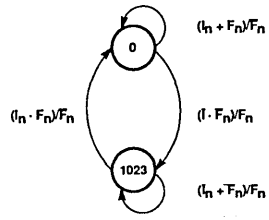
Programmable logic sequencers (12 × 48 × 8) PLS168/A

TEST ARRAY

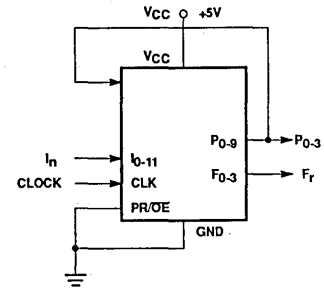
The PLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to $I_{0\text{-}13}$ as shown in the test circuit timing diagram.



State Diagram

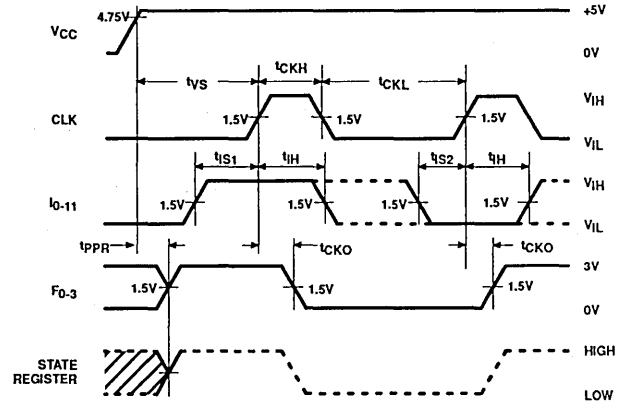


FPLS Under Test

TERM	C_n	AND INPUT (Im) 11 10 9 8 7 6 5 4 3 2 1 0	AND PRESENT STATE (Ps) 9 8 7 6 5 4 3 2 1 0	OPTION (P/E) H — OR NEXT STATE (Ns) 9 8 7 6 5 4 3 2 1 0	OUTPUT (Fr) 3 2 1 0
48	A	H H H H H H H H H H H H	H H H H H H H H H H	L L L L L L L L L L	L L L L
49	•	L L L L L L L L L L L L	L L L L L L L L L L	H H H H H H H H H H	H H H H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any Signetic's qualified programming equipment.



Test Circuit Timing Diagram

TERM	C_n	AND INPUT (Im) 11 10 9 8 7 6 5 4 3 2 1 0	AND PRESENT STATE (Ps) 9 8 7 6 5 4 3 2 1 0	OPTION (P/E) H — OR NEXT STATE (Ns) 9 8 7 6 5 4 3 2 1 0	OUTPUT (Fr) 3 2 1 0
48	–	H H H H H H H H H H H H	H H H H H H H H H H	– – – – – – – – – –	– – – –
49	•	L L L L L L L L L L L L	L L L L L L L L L L	– – – – – – – – – –	– – – –

Test Array Deleted

Philips Components–Signetics

Document No.	853–0862
ECN No.	93255
Date of Issue	May 11, 1988
Status	Product Specification
Programmable Logic Devices	

PLS179

Programmable logic sequencer (20 × 45 × 12)

DESCRIPTION

The PLS179 is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "foldback" inverting buffer and control gate, F_C. It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). There are 8 dedicated inputs. These yield variable I/O gate and register configurations via control gates (D, L) ranging from 20 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 8 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and the Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates can drive bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the 32 AND gates can drive the J-K inputs of all flip-flops. Four AND gates have been dedicated for the Asynchronous Preset/Reset functions.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

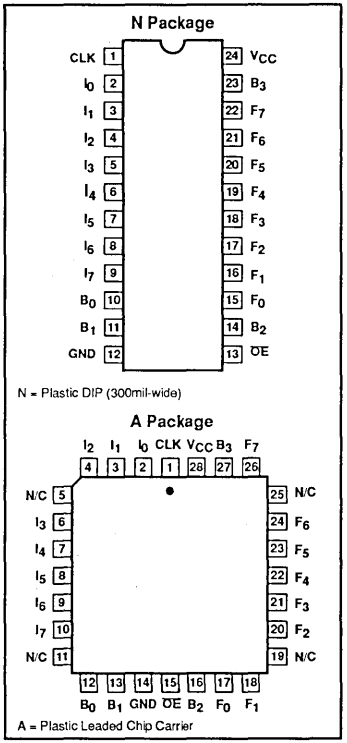
The PLS179 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are listed in the Ordering Information Table.

FEATURES

- f_{MAX} = 18.2MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 8 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J/K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Power-on reset on flip-flop (F_n = "1")
- Input loading: – 100μA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- 3-State outputs

PIN CONFIGURATIONS

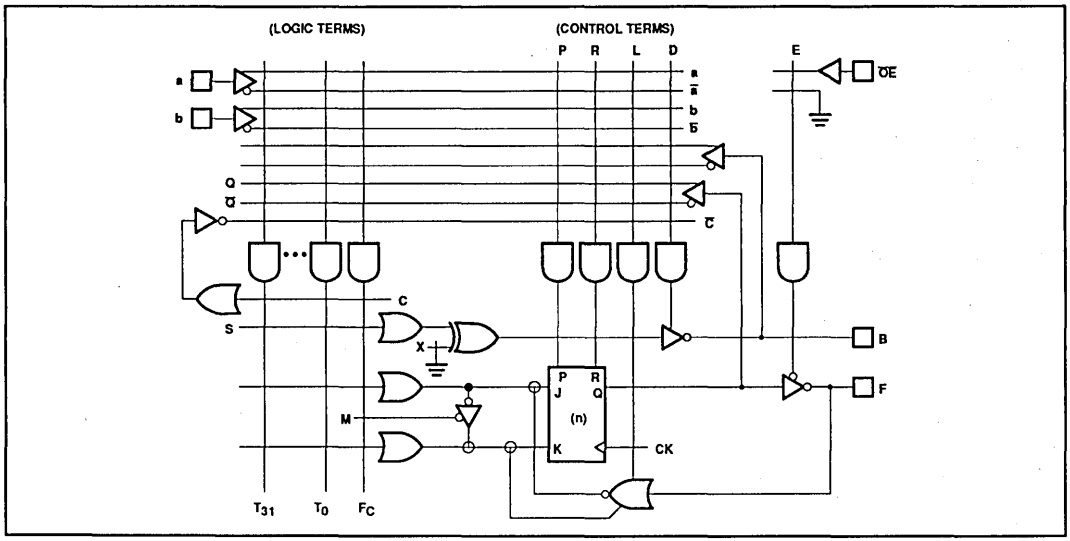


APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

Programmable logic sequencer (20 × 45 × 12) PLS179

FUNCTIONAL DIAGRAM

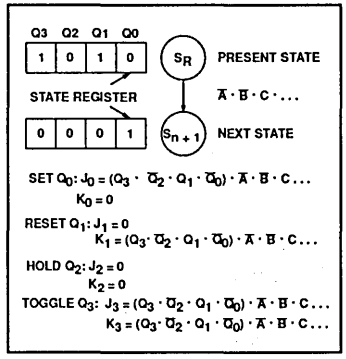


FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								H/Hi-Z
L	X	X	X	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

NOTES:
1. Positive Logic:
   $J\text{-}K = \overline{T_0} + \overline{T_1} + T_2 \ldots\ldots\ldots\ldots\ldots \overline{T_{31}}$
   $\overline{T_n} = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

LOGIC FUNCTION



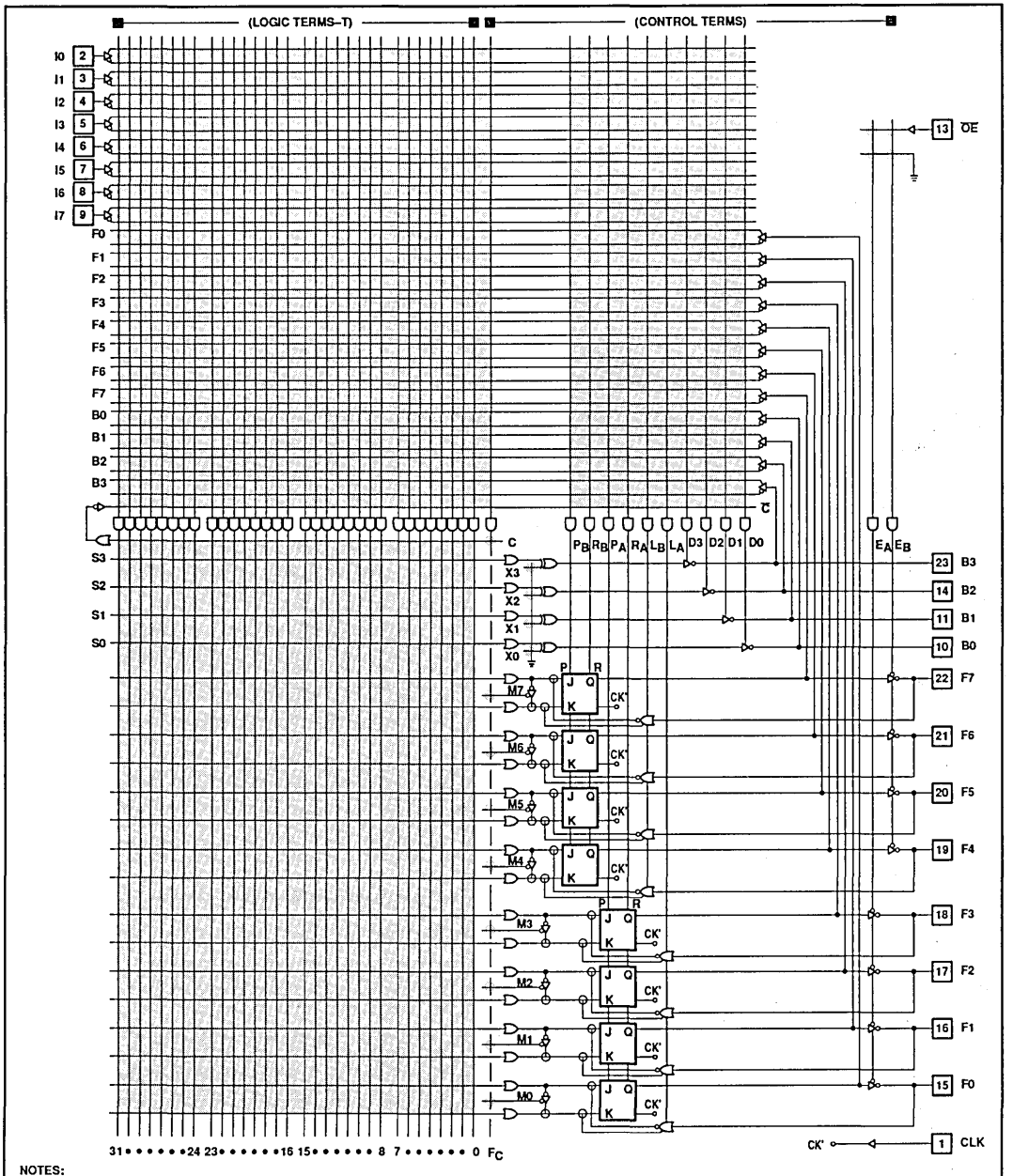
NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

Programmable logic sequencer (20 × 45 × 12) PLS179

LOGIC DIAGRAM



NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ▒ Programmable connection.

Programmable logic sequencer (20 × 45 × 12) PLS179

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic DIP (300mil-wide)	PLS179N
28-Pin Plastic Leaded Chip Carrier	PLS179A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIn			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 10mA		0.35	0.5	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = 5.5V		<1	40	µA
I_{IL}	Low	V_{IN} = 0.45V		–10	–100	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[4, 7]	V_{CC} = MAX, V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V			–140	µA
I_{OS}	Short circuit[3, 5]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[6]	V_{CC} = MAX		150	210	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IH} applied to $\overline{OE}$.
5. Duration of short circuit should not exceed 1 second.
6. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
7. Leakage values are a combination of input and output leakage.

Programmable logic sequencer (20 × 45 × 12) PLS179

AC ELECTRICAL CHARACTERISTICS

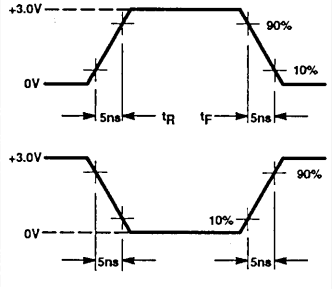
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75°CV \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN[5]	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width[3]								
t_{CKH}	Clock[2] High	CK +	CK –	$C_L = 30pF$	20	15		ns
t_{CKL}	Clock Low	CK –	CK +	$C_L = 30pF$	20	15		ns
t_{CKP}	Clock period	CK +	CK +	$C_L = 30pF$	40	30		ns
t_{PRH}	Preset/Reset pulse	(I, B) –	(I, B) +	$C_L = 30pF$	35	30		ns
Setup time								
t_{IS1}	Input	(I, B) ±	CK +	$C_L = 30pF$	35	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	$C_L = 30pF$	15	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I, B) ±	CK +	$C_L = 30pF$	55	45		ns
Hold time								
t_{IH1}	Input	(I, B) ±	CK +	$C_L = 30pF$	0	–5		ns
t_{IH2}	Input (through F_n)	F ±	CK +	$C_L = 30pF$	15	10		ns
Propagation delay								
t_{CKO}	Clock	CK ±	F ±	$C_L = 30pF$		15	20	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	$C_L = 30pF$		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	$C_L = 5pF$		20	30	ns
t_{PD}	Output	(I, B) ±	B ±	$C_L = 30pF$		25	35	ns
t_{OE2}	Output enable[3]	(I, B) +	B ±	$C_L = 30pF$		20	30	ns
t_{OD2}	Output disable[3]	(I, B) –	B +	$C_L = 5pF$		20	30	ns
t_{PRO}	Preset/Reset	(I, B) +	F ±	$C_L = 30pF$		35	45	ns
t_{PPR}	Power-on preset	V_{CC} +	F –	$C_L = 30pF$		0	10	ns

NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} – 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array $t_{CKP} = 75ns$ (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

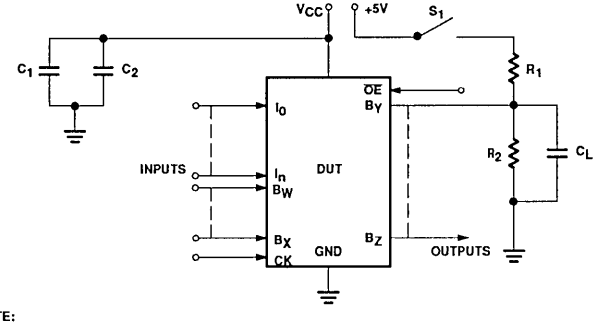
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

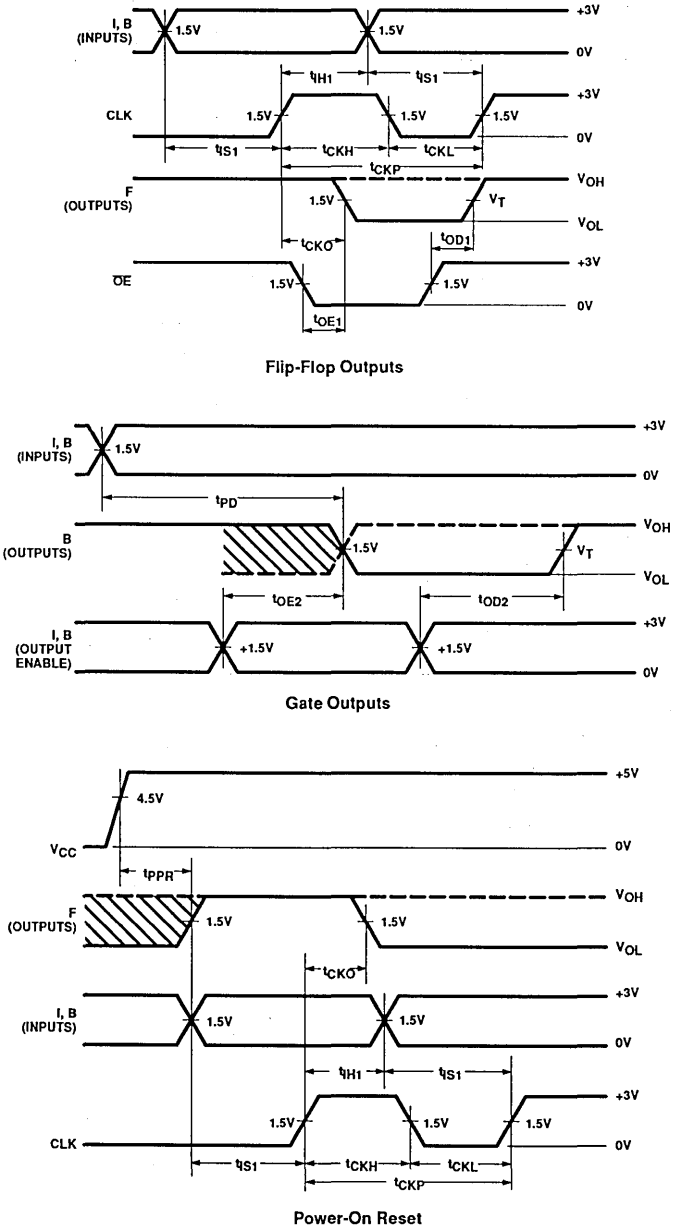
TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Programmable logic sequencer (20 × 45 × 12) PLS179

TIMING DIAGRAMS

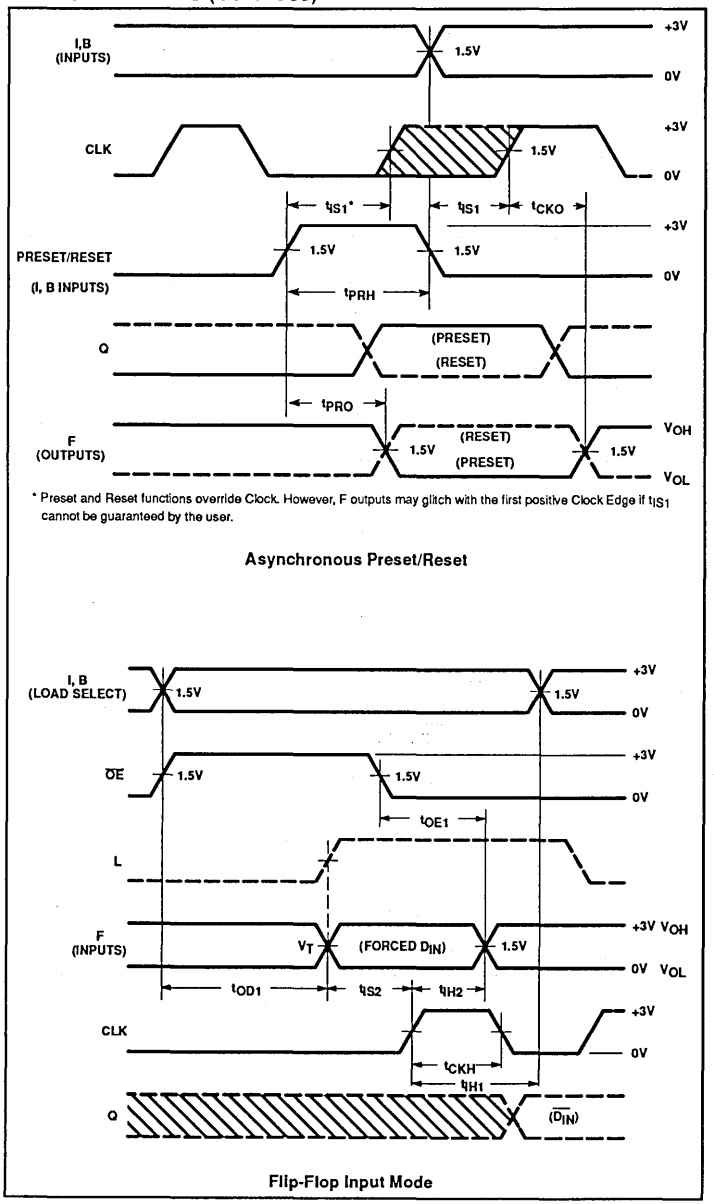


TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational Outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational Outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

Programmable logic sequencer (20 × 45 × 12) PLS179

TIMING DIAGRAMS (Continued)



Programmable logic sequencer (20 × 45 × 12) PLS179

LOGIC PROGRAMMING

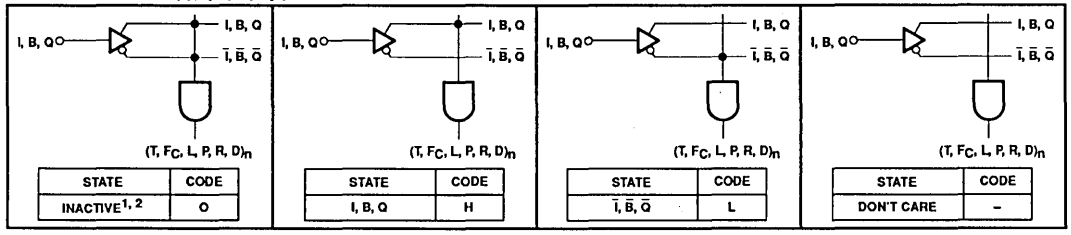
The PLS179 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE, Data I/O Corporation's ABEL™, and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS179 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

"AND" ARRAY – (I), (B), (Qp)



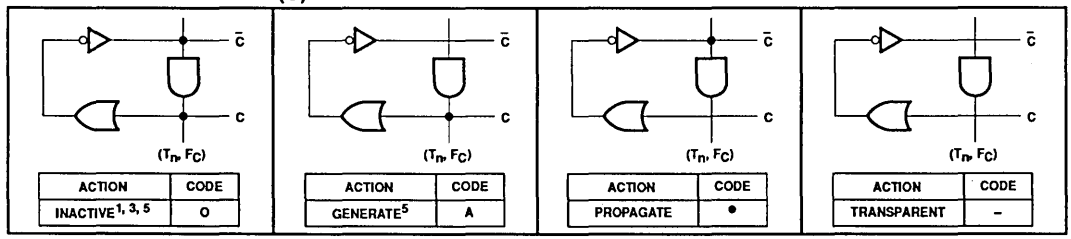
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
Ī, B̄, Q̄	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



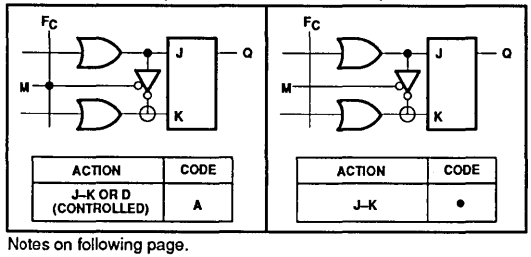
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

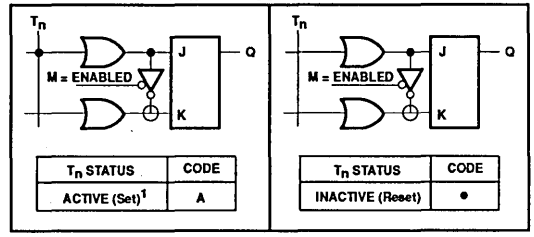
"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)	A

ACTION	CODE
J–K	•

"OR" ARRAY – (Q_n = D-Type)



Tn STATUS	CODE
ACTIVE (Set)[1]	A

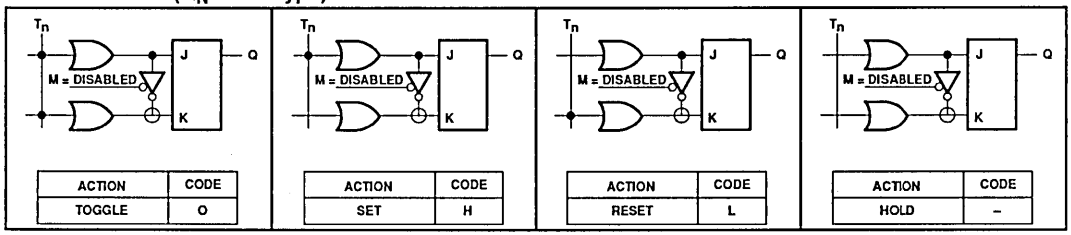
Tn STATUS	CODE
INACTIVE (Reset)	•

Notes on following page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic sequencer (20 × 45 × 12) PLS179

"AND" ARRAY – (Q_N = J-K Type)



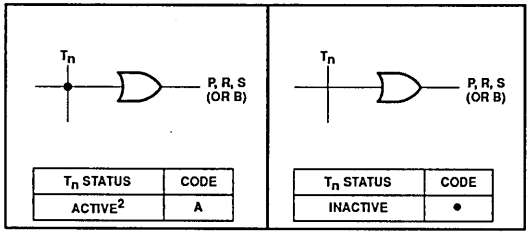
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

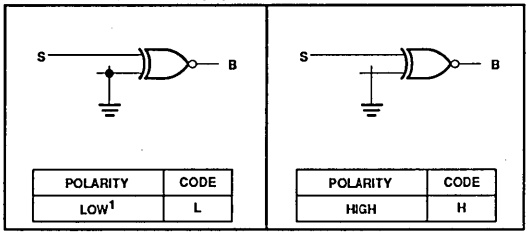
"OR" ARRAY – (S or B)



T_n STATUS	CODE
ACTIVE[2]	A

T_n STATUS	CODE
INACTIVE	•

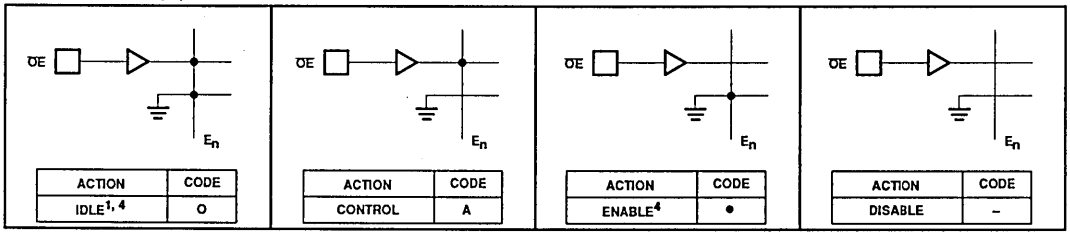
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW[1]	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

ACTION	CODE
ENABLE[4]	•

ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate $(T, F_C, L, P, R, D)_n$ will be unconditionally inhibited if any one of the I, B, or Q link pairs are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates $(L, P, R, D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (20 × 45 × 12) PLS179

PROGRAM TABLE

AND

INACTIVE	O	I, B(I), Q(P)
I, B, Q	H	
$\overline{I}$, $\overline{B}$, $\overline{Q}$	L	
DON'T CARE	–	

INACTIVE	O	C
GENERATE	A	
PROPAGATE	•	
TRANSPARENT	–	

OR

ACTIVE	A	P, R, B(O) (Q = D)
INACTIVE	•	

TOGGLE	O	(Q = J/K)
SET	H	
RESET	L	
HOLD	–	

CONTROL

J/K	•	F/F MODE
J/K or D (controlled)	A	

IDLE	0	$E_{A, B}$
CONTROL	A	
ENABLE	•	
DISABLE	–	

HIGH	H	(POL)
LOW	L	

NOTES

1. The device is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity.
2. Program unused C, I, B, and Q bits in the AND array as (–). Program unused Q, B, P, and R bits in the OR array as (–) or (A), as applicable.
3. Unused Terms can be left blank.
4. Q (P) and Q (N) are respectively the present and next states of flip-flops Q.

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____

DATE RECEIVED _____ COMMENTS _____

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____ REV _____ DATE _____

TERM	AND C	AND I 7	6	5	4	3	2	1	0	AND B(I) 3	2	1	0	AND Q(P) 7	6	5	4	3	2	1	0	F/F MODE	E_B	E_A	POLARITY	(OR) Q(N) 7	6	5	4	3	2	1	0	(OR) B(O) 3	2	1	0
0																																				
1																																				
2																																				
3																																				
4																																				
5																																				
6																																				
7																																				
8																																				
9																																				
10																																				
11																																				
12																																				
13																																				
14																																				
15																																				
16																																				
17																																				
18																																				
19																																				
20																																				
21																																				
22																																				
23																																				
24																																				
25																																				
26																																				
27																																				
28																																				
29																																				
30																																				
31																																				
F_C																																				
P_B																																				
R_B																																				
L_B																																				
P_A																																				
R_A																																				
L_A																																				
D3																																				
D2																																				
D1																																				
D0																																				
PIN		9	8	7	6	5	4	3	2	23	14	11	10	22	21	20	19	18	17	16	15														

Document No.	853–1414
ECN No.	99206
Date of Issue	March 26, 1990
Status	Product Specification
Programmable Logic Devices	

PLC42VA12

CMOS programmable multi-function PLD ($42 \times 105 \times 12$)

DESCRIPTION

The new PLC42VA12 CMOS PLD from Signetics exhibits a unique combination of the two architectural concepts that revolutionized the PLD marketplace.

The Signetics unique Output Macro Cell (OMC) embodies all the advantages and none of the disadvantages associated with the "V" type Output Macro Cell devices. This new design, combined with added functionality of two programmable arrays, represents a significant advancement in the configurability and efficiency of multi-function PLDs.

The most significant improvement in the Output Macro Cell structure is the implementation of the register bypass function. Any of the 10 J-K/D registers can be individually bypassed, thus creating a combinatorial I/O path from the AND array to the output pin. Unlike other "V" type devices, the register in the PLC42VA12 Macro Cell remains fully functional as a buried register. Both the combinatorial I/O and buried register have separate input paths (from the AND array). In most V-type architectures, the register is lost as a resource when the cell is configured as a combinatorial I/O. This feature provides the capability to operate the buried register independently from the combinatorial I/O.

The PLC42VA12 is an EPROM-based CMOS device. Designs can be generated using Signetics AMAZE, SNAP and SLICE PLD design software packages or one of several other commercially available JEDEC standard PLD design software packages.

FEATURES

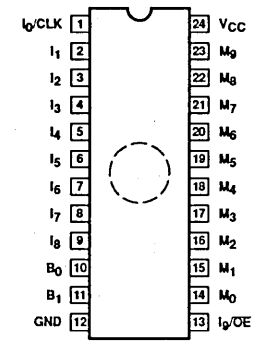
- High-speed EPROM-based CMOS Multi-Function PLD
  - Super set of 22V10, 32VX10 and 20RA10 PAL® ICs
- Two fully programmable arrays eliminate "P-term Depletion"
  - Up to 64 P-terms per OR function
- Improved Output Macro Cell Structure
  - Individually programmable as:
    - Registered Output with feedback
    - Registered Input
    - Combinatorial I/O with Buried Register
    - Dedicated I/O with feedback
    - Dedicated Input (combinatorial)
  - Bypassed Registers are 100% functional with separate input and feedback paths
  - Individual Output Enable control functions
    - From pin or AND array
- Eleven clock sources
- Register Preload and Diagnostic Test Mode Features
- Security fuse

APPLICATIONS

- Mealy or Moore State Machines
  - Synchronous
  - Asynchronous
- Multiple, independent State Machines
- 10-bit ripple cascade
- Sequence recognition
- Bus Protocol generation
- Industrial control
- A/D Scanning

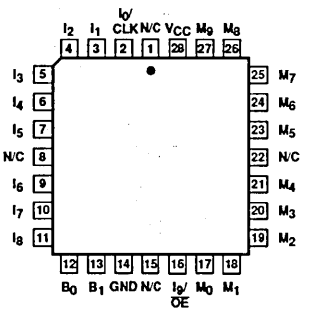
PIN CONFIGURATIONS

FA and N Packages



N = Plastic DIP (300mil-wide)
FA = Ceramic DIP with Quartz Window (300mil-wide)

A Package



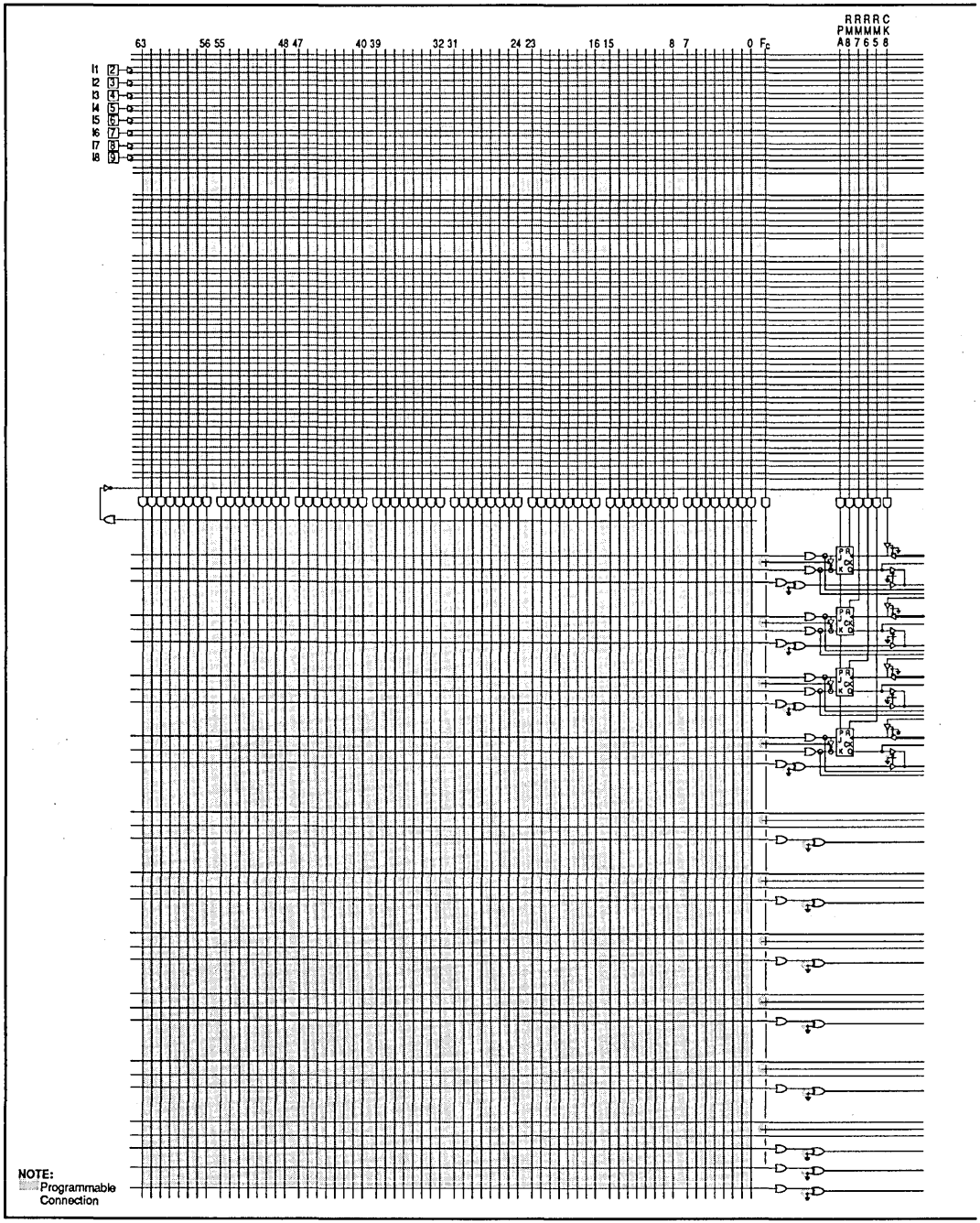
A = Plastic Leaded Chip Carrier (450mil-square)

PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

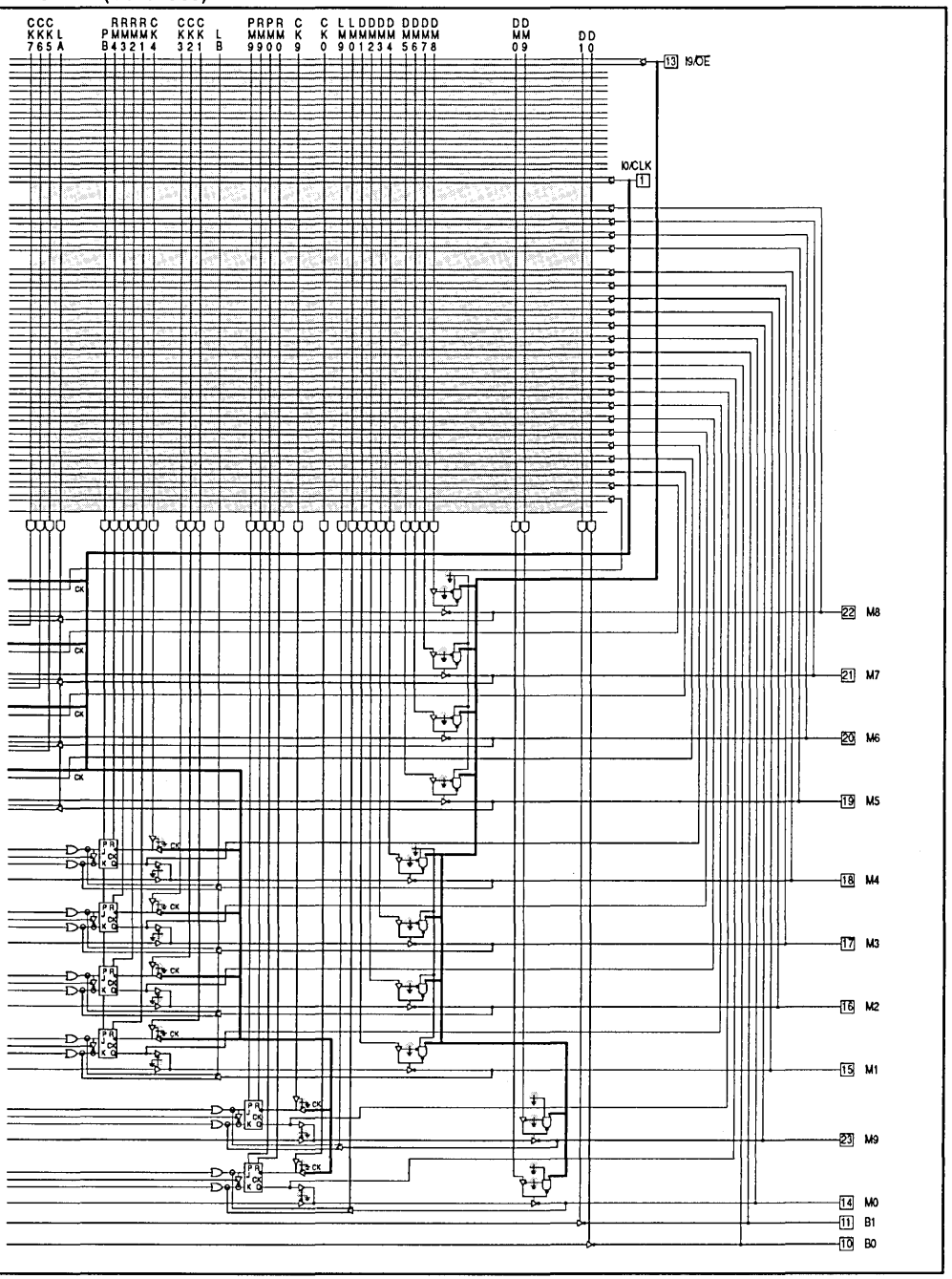
LOGIC DIAGRAM



CMOS programmable multi-function PLD ($42 \times 105 \times 12$)

PLC42VA12

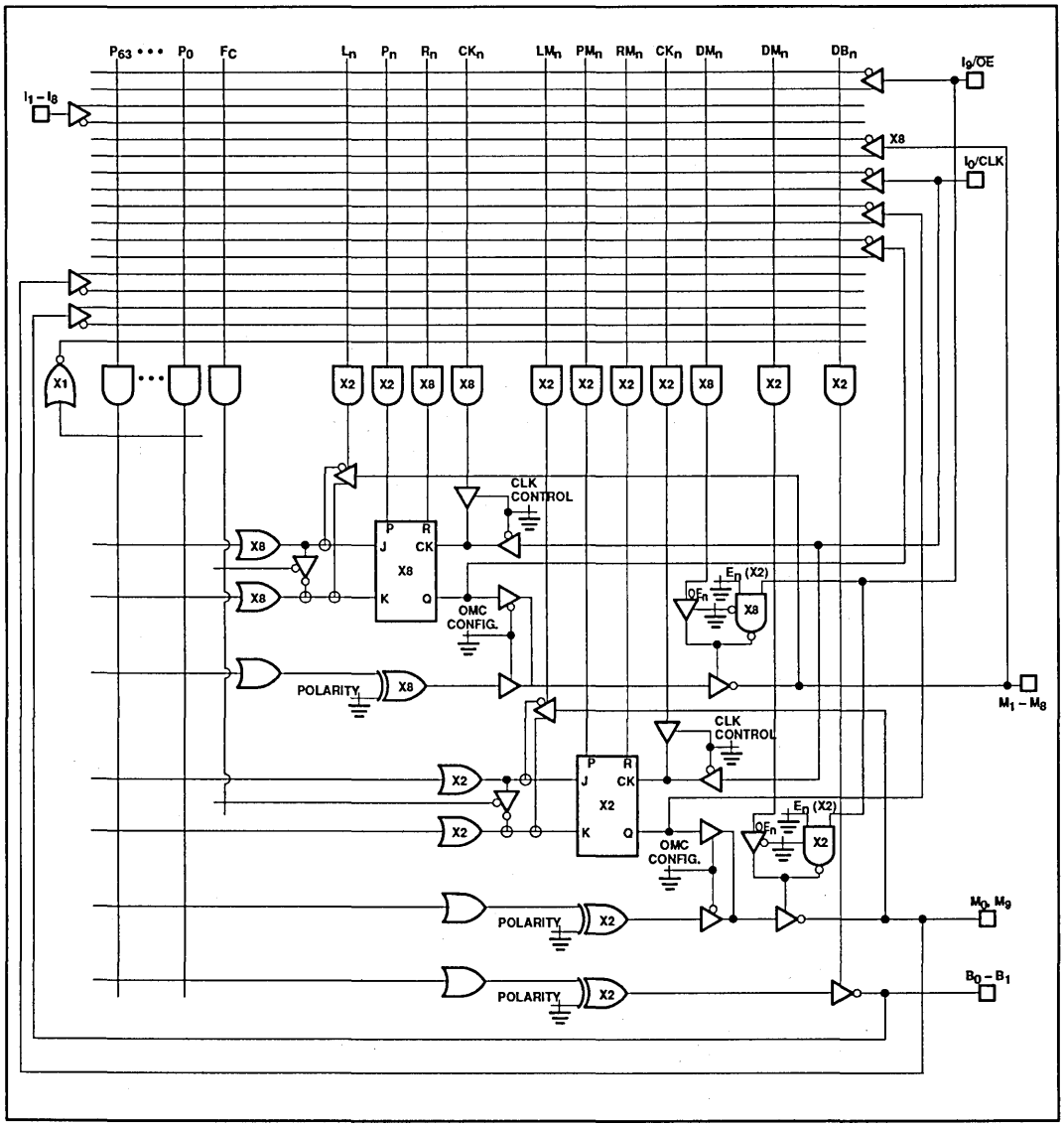
LOGIC DIAGRAM (Continued)



CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

FUNCTIONAL DIAGRAM



CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Ceramic Dual In-Line with window, Reprogrammable (300mil-wide)	PLC42VA12FA
24-Pin Plastic Dual In-Line, One Time Programmable (300mil-wide)	PLC42VA12N
28-Pin Plastic Leaded Chip Carrier, One Time Programmable (450mil-wide)	PLC42VA12A

ABSOLUTE MAXIMUM RATINGS[1]

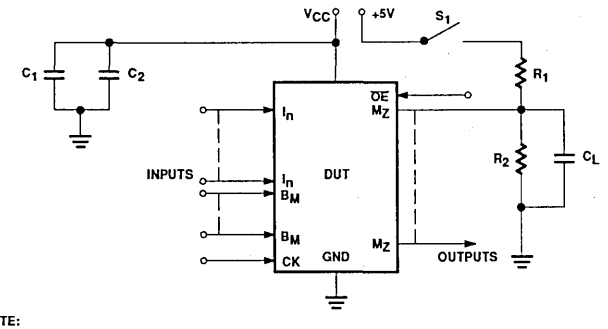
SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	–0.5 to +7	V_{DC}
V_{IN}	Input voltage	–0.5 to V_{CC} +0.5	V_{DC}
V_{OUT}	Output voltage	–0.5 to V_{CC} +0.5	V_{DC}
I_{IN}	Input currents	–10 to +10	mA
I_{OUT}	Output currents	+24	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

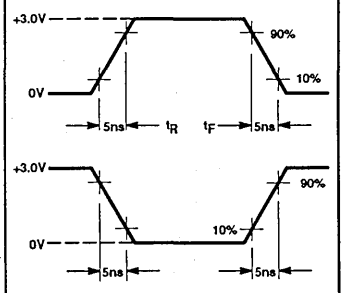
AC TEST CONDITIONS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Test Load Circuit

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN; I_{OL} = 16mA		0.3	0.5	V
V_{OH}	High	V_{CC} = MIN; I_{OH} = –3.2mA	2.4	4.3		V
Input current						
I_{IL}	Low	V_{IN} = GND		–1	–10	µA
I_{IH}	High	V_{IN} = V_{CC}		+1	10	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC} V_{OUT} = GND		1 –1	10 –10	µA µA
I_{OS}	Short-circuit[3,6]	V_{OUT} = GND			–130	mA
I_{CC}	V_{CC} supply current (Active)[4]	I_{OUT} = 0mA, f = 15MHz[5], V_{CC} = MAX		90	120	mA
Capacitance						
C_I	Input	V_{CC} = 5V; V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short-circuit should not exceed one second. Test one at a time.
4. Tested with V_{IL} = 0.45V, V_{IH} = 2.4V.
5. Refer to Figure 1, ΔI_{CC} vs Frequency (worst case). (Referenced from 15MHz)
   The I_{CC} increases by 1.5mA per MHz for the frequency range of 16MHz up to 25MHz.
   The I_{CC} remains at a worst case of 135mA for the frequency range of 26MHz up to 37MHz.
   The I_{CC} decreases by 1.0mA per MHz for the frequency range of 14MHz down to 1MHz (106mA @ 1MHz).
   The worst case I_{CC} is calculated as follows:
   - All dedicated inputs are switching.
   - All OMCs are configured as JK flip-flops in the toggle mode. . .all are toggling.
   - All 12 outputs are disabled.
   - The number of product terms connected does not impact the I_{CC}.
   - I_{CC} levels are identical for both TTL and CMOS input levels.
6. Refer to Figure 2 for Δt_{PD} vs output capacitance loading.

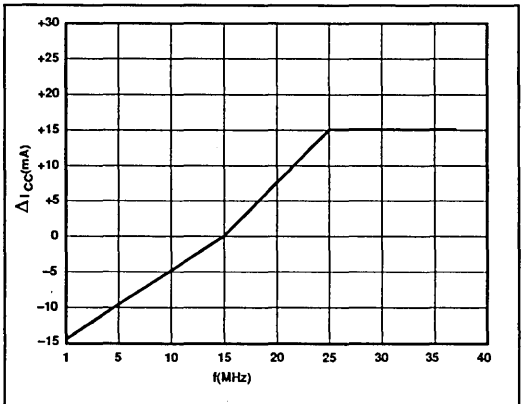


Figure 1. ΔI_{CC} vs Frequency (Worst Case) (Referenced from 15MHz)

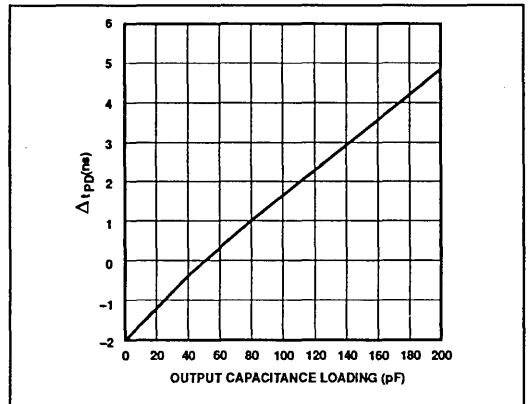


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$; $R_1 = 238\Omega$, $R_2 = 170\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST[2] CONDITION (C_L (pF))	PLC42VA12 MIN	PLC42VA12 TYP[1]	PLC42VA12 MAX	UNIT
Set-up Time								
t_{IS1}	Input; dedicated clock	(I, B, M) +/–	CK+	50	23	16		ns
t_{IS2}	Input; P-term clock	(I, B, M) +/–	(I, B, M) +/–	50	20	13		ns
t_{IS3}[3]	Preload; dedicated clock	(M) +/–	CK+	50	10	3.5		ns
t_{IS4}[3]	Preload; P-term clock	(M) +/–	(I, B, M) +/–	50	2	–1.0		ns
t_{IS5}[3]	Input through complement array; dedicated clock	(I, B, M) +/–	CK+	50	50	34		ns
t_{IS6}[3]	Input through complement array; P-term clock	(I, B, M) +/–	(I, B, M) +/–	50	40	30		ns
Propagation Delay								
t_{PD1}	Propagation Delay	(I, B, M) +/–	(I, B, M) +/–	50		20	35	ns
t_{PD2}	Propagation Delay with complement array (2 passes)	(I, B,) +/–	(I, B, M) +/–	50		36	55	ns
t_{CKO1}	Clock to Output; Dedicated clock	CK+	(M) +/–	50		13	17	ns
t_{CKO2}	Clock to output; P-term clock	(I, B, M) +/–	(M) +/–	50		18	27	ns
t_{RP1}	Registered operating period; Dedicated clock ($t_{IS1} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50		29	40	ns
t_{RP2}	Registered operating period; P-term clock ($t_{IS2} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50		31	47	ns
t_{RP3}[3]	Register preload operating period; Dedicated clock ($t_{IS3} + t_{CKO1}$)	(M) +/–	(M) +/–	50		16.5	27	ns
t_{RP4}[3]	Register preload operating period; P-term clock ($t_{IS4} + t_{CKO2}$)	(M) +/–	(M) +/–	50		17	29	ns
t_{RP5}[3]	Registered operating period with complement array; dedicated clock ($t_{IS5} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50		47	67	ns
t_{RP6}[3]	Registered operating period with complement array; P-term clock ($t_{IS6} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50		48	67	ns
t_{OE1}	Output Enable; from /OE pin[4]	/OE –	(M) +/–	50		10	20	ns
t_{OE2}	Output Enable; from P-term[4]	(I, B, M) +/–	(B, M) +/–	50		12.5	25	ns
t_{OD1}	Output Disable; from /OE pin[4]	/OE +	Outputs disabled	5		10	20	ns
t_{OD2}	Output Disable; from P-term[4]	(I, B, M) +/–	Outputs disabled	5		14.5	25	ns
t_{PRO}[3]	Preset to Output	(I, B, M) +/–	(M) +/–	50		25	35	ns
t_{PPR}[3]	Power-on Reset (Mn = 1)	V_{CC} +	(M) +/–	50			15	ns
Hold Time								
t_{IH1}	Input (Dedicated clock)	CK+	(I, B, M) +/–	50	0	–13		ns
t_{IH2}	Input (P-term clock)	(I, B, M) +/–	(I, B, M) +/–	50	5	–7.5		ns
t_{IH3}[3]	Input; from Mn (Dedicated clock)	CK+	(M) +/–	50	5	–1.5		ns
t_{IH4}[3]	Input; from Mn (P-term clock)	(I, B, M) +/–	(M) +/–	50	10	3.5		ns
Pulse Width								
t_{CKH1}	Clock High; Dedicated clock	CK+	CK–	50	10	5		ns
t_{CKL1}	Clock Low; Dedicated clock	CK–	CK+	50	10	5		ns
t_{CKH2}	Clock High; P-term clock	CK+	CK–	50	15	7		ns
t_{CKL2}	Clock Low; P-term clock	CK–	CK+	50	15	7		ns
t_{PRH}[3]	Width of preset/reset input pulse	(I, B, M) +/–	(I, B, M) +/–	50	30	7		ns

Notes on page 306.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

AC ELECTRICAL CHARACTERISTICS (Continued)

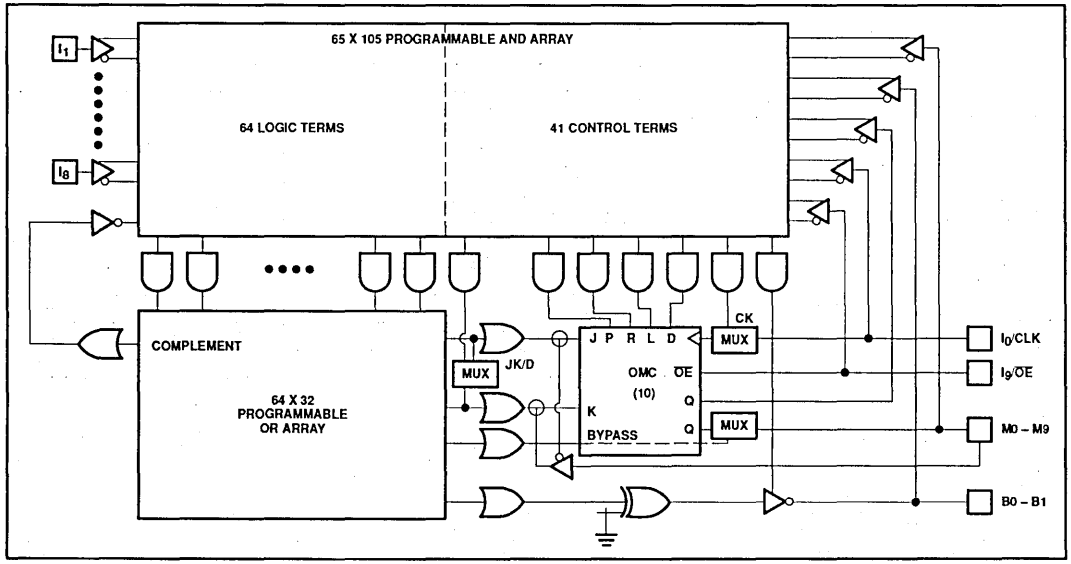
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$; $R_1 = 238\Omega$, $R_2 = 170\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST[2] CONDITION (C_L (pF))	PLC42VA12 MIN	PLC42VA12 TYP[1]	PLC42VA12 MAX	UNIT
Frequency of Operation								
f_{CK1}	Dedicated clock frequency	C+	C+	50	50	100		MHz
f_{CK2}	P-term clock frequency	C+	C+	50	33	71.4		MHz
f_{MAX1}	Registered operating frequency; Dedicated clock ($t_{IS1} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50	25	34.5		MHz
f_{MAX2}	Registered operating frequency; P-term clock ($t_{IS2} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50	21.3	32.3		MHz
f_{MAX3}[3]	Register preload operating frequency; Dedicated clock ($t_{IS3} + t_{CKO1}$)	(M) +/–	(M) +/–	50	37	60.6		MHz
f_{MAX4}[3]	Register preload operating frequency; P-term clock ($t_{IS4} + t_{CKO2}$)	(M) +/–	(M) +/–	50	34.5	58.8		MHz
f_{MAX5}[3]	Registered operating frequency with complement array; Dedicated clock ($t_{IS5} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50	14.9	21.3		MHz
f_{MAX6}[3]	Registered operating frequency with complement array; P-term clock ($t_{IS6} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50	14.9	20.8		MHz

NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$. These limits are not tested/guaranteed.
2. Refer also to AC Test Conditions (Test Load Circuit).
3. These limits are not tested, but are characterized periodically and are guaranteed by design.
4. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

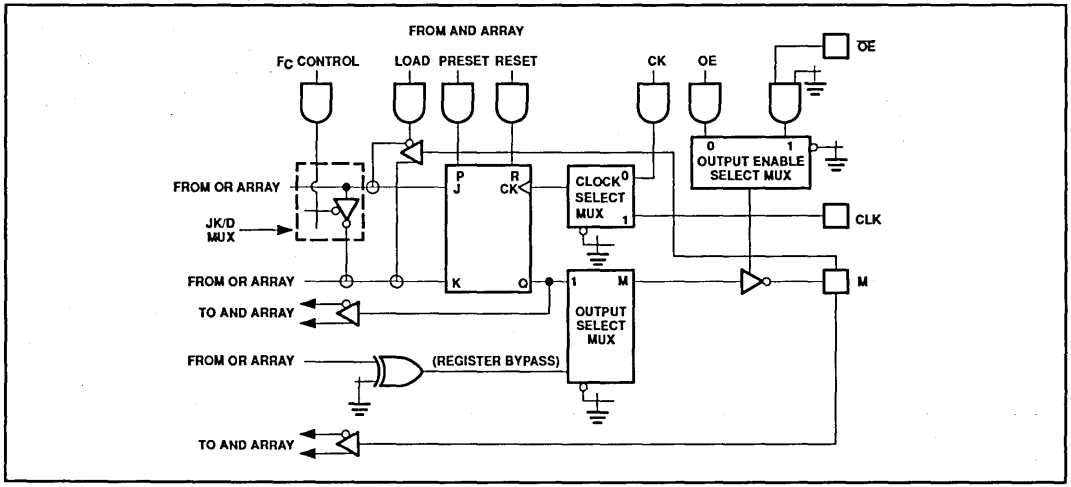
BLOCK DIAGRAM



CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

OUTPUT MACRO CELL (OMC)



Output Macro Cell Configuration

Signetics unique Output Macro Cell design represents a significant advancement in the configurability of multi-function Programmable Logic Devices.

The PLC42VA12 has 10 programmable Output Macro Cells. Each can be individually programmed in any of 5 basic configurations:

- Dedicated I/O (combinatorial) with feedback to AND array
- Dedicated Input
- Combinatorial I/O with feedback and Buried Register with feedback (register bypass)
- Registered Input
- Registered Output with feedback

Each of the registered options can be further customized as J-K type or D-type, with either an internally derived clock (from the AND array) or clocked from an external source. With these additional programmable options, it is possible to program each Output Macro Cell in any one of 14 different configurations.

These 14 configurations, combined with the fully programmable OR array, make the PLC42VA12 the most versatile and silicon efficient of all the Output Macro Cell-type PLDs.

The most significant Output Macro Cell (OMC) feature is the implementation of the register bypass function. Any of the 10 J-K/D registers can be individually bypassed, thus creating a combinatorial I/O path from the AND array to the output pin. Unlike other Output Macro Cell-type devices, the register in the OMC is fully functional as a buried register. Furthermore, both the combinatorial I/O and the buried register have separate input paths (from the AND array) and separate feedback paths (to the AND array). This feature provides the capability to operate the buried register independently from the combinatorial I/O.

The PLC42VA12 is ideally suited for both synchronous and asynchronous logic functions. Eleven clock sources – 10 driven from the AND array and one from an external source – make it possible to design synchronous state machine functions, event-driven state machine functions and combinatorial (asynchronous) functions all on the same chip.

Sophisticated control functions support individual OE control and Reset functions from the AND array. OE control is also available from the $I_9/\overline{OE}$ pin. Register Preset and Load functions are controlled from the AND array, in 2 banks of 4 for OMCs M_1 – M_8. Output Macro Cells M_0 and M_9 have individual Preset and Load Control terms.

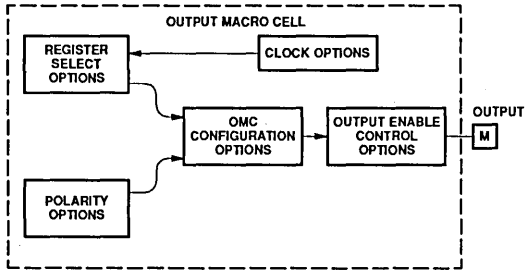
Output Polarity for the combinatorial I/O paths is configurable via 12 programmable EX-OR gates. The output of each register can be configured as inverting (active Low) or non-inverting (active High) via manipulation of the logic equations.

The output of each buried register can also be configured as inverting or non-inverting via the input buffer which feeds back to the AND array.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

OUTPUT MACRO CELL PROGRAMMABLE OPTIONS



OMC Programmable Options

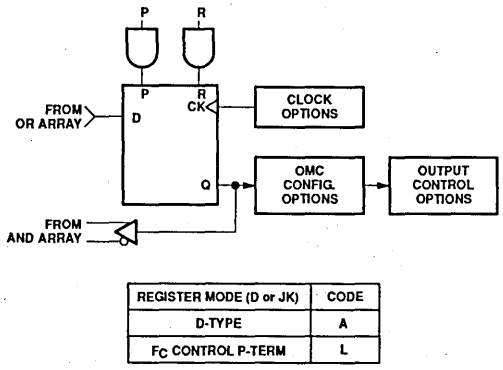
For purposes of programming, the Output Macro Cell should be considered to be partitioned into five separate blocks. As shown in the drawing titled "Output Macro Cell Programmable Options", the programmable blocks are: Register Select Options, Polarity Options, Clock Options, OMC Configuration Options and Output Enable Control Options.

There is one programmable location associated with each block except the Output Enable Control block which has two programmable fuse locations per OMC.

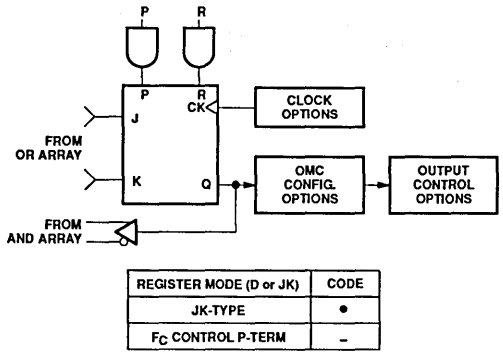
The following drawings detail the options associated with each programmable block. The associated programming codes are also included. The table titled "Output Macro Cell Configurations" (page 15) lists all the possible combinations of the five programmable options.

ARCHITECTURAL OPTIONS

REGISTER SELECT OPTIONS



REGISTER MODE (D or JK)	CODE
D-TYPE	A
F_C CONTROL P-TERM	L



REGISTER MODE (D or JK)	CODE
JK-TYPE	•
F_C CONTROL P-TERM	–

Register Select Options

Each OMC Register can be configured either as a dedicated D-type or a J-K flip-flop. The Flip-Flop Control term, Fc, provides the option to control each Register dynamically—switching from D-type to J-K type, based on the Fc control signal.

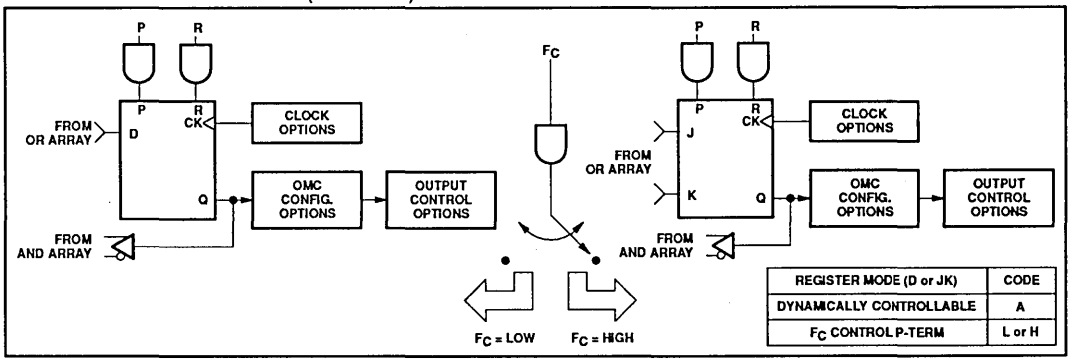
Register Preset and Reset are controlled from the AND array. Each OMC has an individual Reset Control term (RMn). The Register Preset function is controlled in two banks of 4 for OMCs $M_1 - M_3$ and $M_4 - M_8$ (via the control terms PA and PB). OMCs M_0 and M_9 have individual control terms (PM_0 and PM_9 respectively).

Notes on page 313.

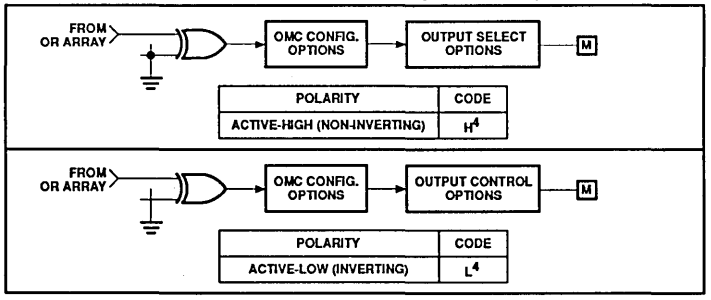
CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

REGISTER SELECT OPTIONS (Continued)



POLARITY OPTIONS (for Combinatorial I/O Configurations Only[1])

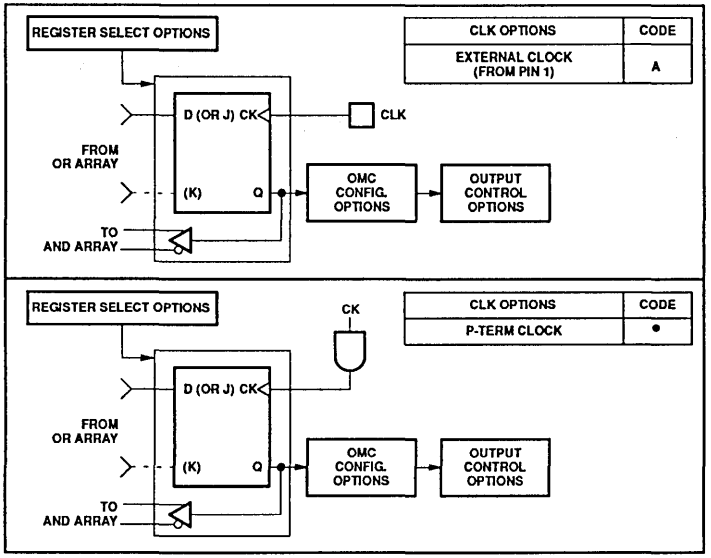


Polarity Options

When an OMC is configured as a Combinatorial I/O with Buried Register, the polarity of the combinatorial path can be programmed as Active-High or Active-Low. A configurable EX-OR gate provides polarity control.

If an OMC is configured as a Registered Output, /Q is propagated to the output pin. Note that either Q or /Q can be fedback to the AND array by manipulating the feedback logic equations. (TRUE or COMPLEMENT).

CLOCK OPTIONS



Clock Options

In the unprogrammed state, all Output Macro Cell clock sources are connected to the External Clock pin (I_0/CLK pin 1). Each OMC can be individually programmed such that its P-term Clock (CK_n) is enabled, thus disabling it from the External Clock and providing event-driven clocking capability.

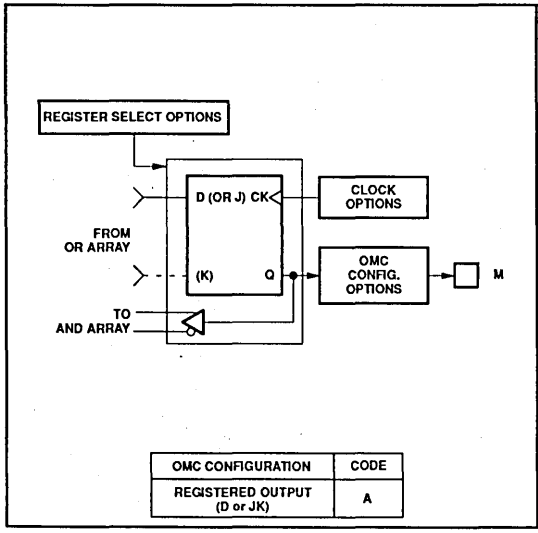
This feature supports multiple state machines, clocked at several different rates, all on one chip, or the ability to collect large amounts of random logic, including 10 separately clocked flip-flops.

Notes on page 313.

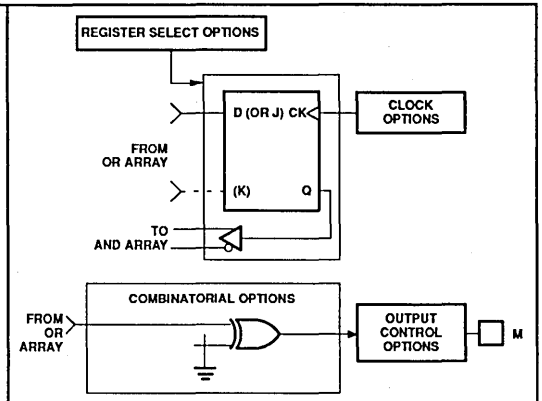
CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

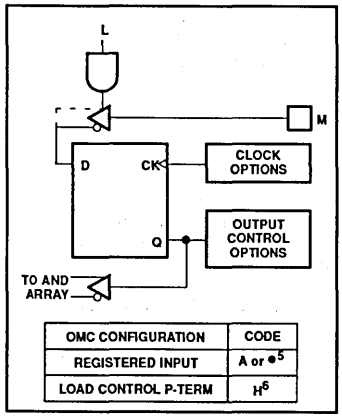
OUTPUT MACRO CELL CONFIGURATION OPTIONS



OMC CONFIGURATION	CODE
REGISTERED OUTPUT (D or JK)	A



OMC CONFIGURATION	CODE
COMBINATORIAL OUTPUT WITH BURIED REGISTER (D or JK)	•



OMC CONFIGURATION	CODE
REGISTERED INPUT	A or •[5]
LOAD CONTROL P-TERM	H[6]

OMC Configuration Options

Each OMC can be configured as a Registered Output with feedback, a Registered Input or a Combinatorial I/O with Buried Register. Dedicated Input and dedicated I/O configurations are also possible.

When the Combinatorial I/O option is selected, (the Register Bypass option), the Buried Register remains 100% functional, with its own inputs from the AND array and a separate feedback path. This unique feature is ideal for designing any type of state machine; synchronous Mealy-types that require both Buried and Output Registers, or asynchronous Mealy-types that require buried registers and combinatorial output functions. Both synchronous and asynchronous Moore-type state machines can also be easily accommodated with the flexible OMC structure.

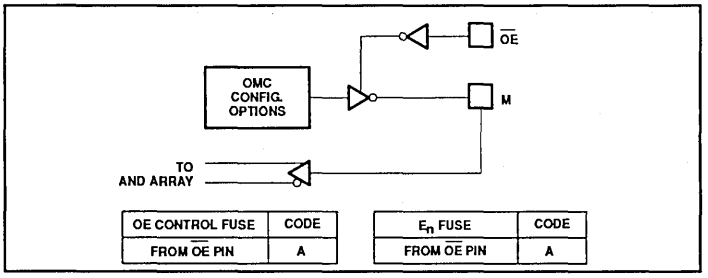
Note that an OMC can be configured as either a Combinatorial I/O (with Buried Register) or a Registered Output with feedback and it can still be used as a Registered Input. By disabling the outputs via any OE control function, the M pin can be used as an input. When the Load Control P-term is asserted HIGH, the register is preloaded from the M pin(s). When the L_C P-term is Active-Low and the output is enabled, the OMC will again function as configured (either a combinatorial I/O or a registered output with feedback). This feature is suited for synchronizing input signals prior to commencing a state sequence.

Notes on page 313.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

OUTPUT CONTROL OPTIONS



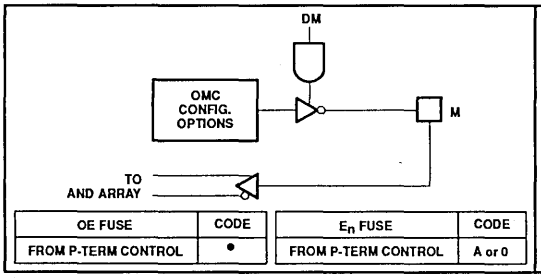
OE CONTROL FUSE	CODE
FROM $\overline{OE}$ PIN	A

E_n FUSE	CODE
FROM $\overline{OE}$ PIN	A

Output Enable Control Options

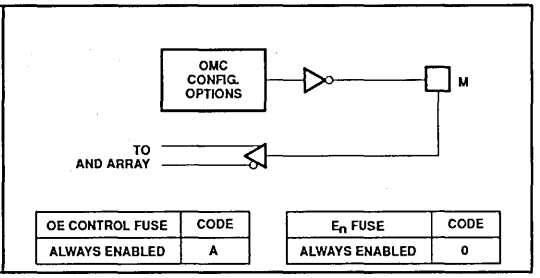
Similar to the Clock Options, the Output Enable Control for each OMC can be connected either to an external source ($I_9/\overline{OE}$, pin 13) or controlled from the AND array (P-terms DM_n). Each Output can also be permanently enabled.

Output Enable control for the two bi-directional I/O (B pins 10 and 11) is from the AND array only (P-terms DB0 and DB1 respectively).



OE FUSE	CODE
FROM P-TERM CONTROL	•

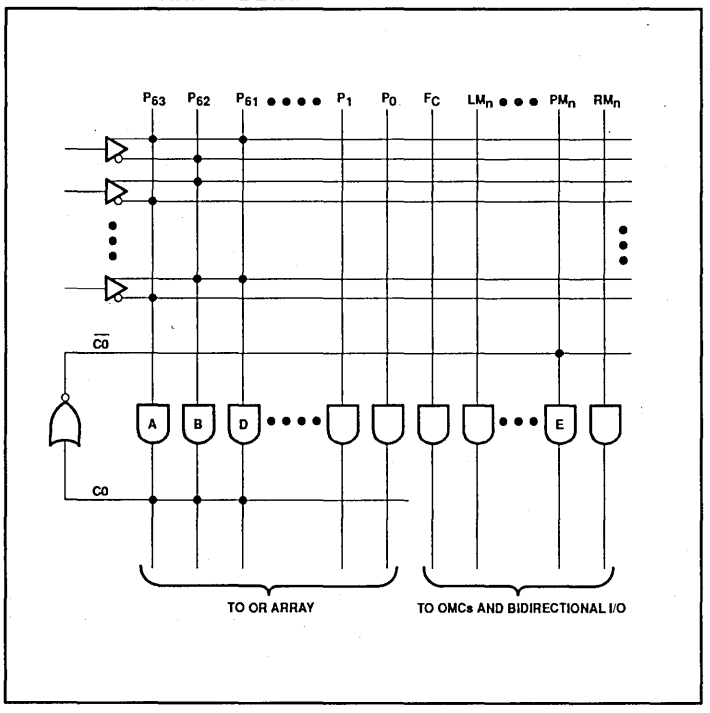
E_n FUSE	CODE
FROM P-TERM CONTROL	A or 0



OE CONTROL FUSE	CODE
ALWAYS ENABLED	A

E_n FUSE	CODE
ALWAYS ENABLED	0

COMPLEMENT ARRAY DETAIL



Complement Array Detail

The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, such an approach can be prohibitive, both in terms of time and wasted resources.

Notes on page 313.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

LOGIC PROGRAMMING

The PLC42VA12 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLC42VA12 architecture.

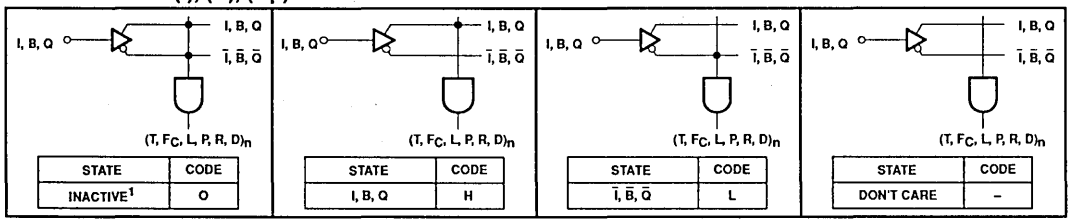
All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC42VA12 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below. Symbols for OMC configuration have been previously defined in the Architectural Options section.

LOGIC IMPLEMENTATION

"AND" ARRAY – (I), (B), (Qp)



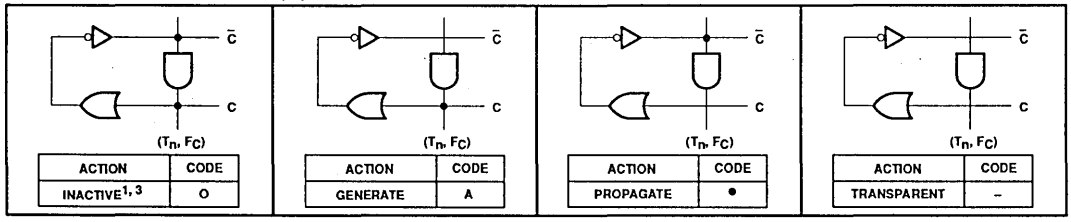
STATE	CODE
INACTIVE[1]	O

STATE	CODE
I, B, Q	H

STATE	CODE
Ī, B̄, Q̄	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



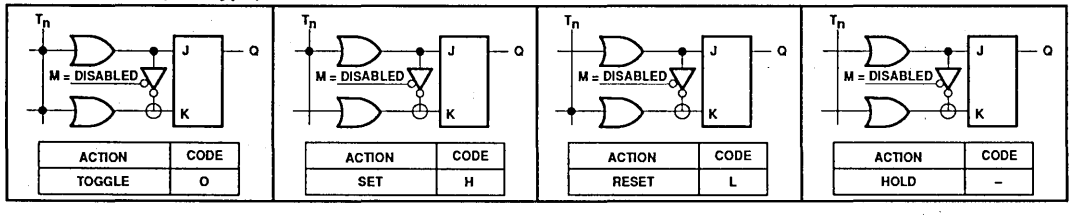
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"OR" ARRAY – (J-K Type)



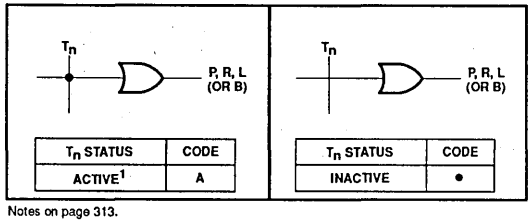
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

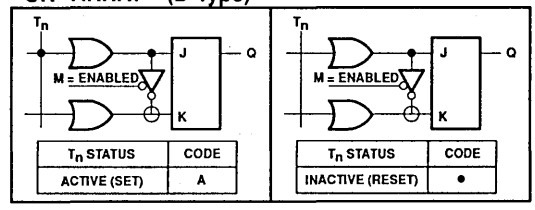
"OR" ARRAY



Tn STATUS	CODE
ACTIVE[1]	A

Tn STATUS	CODE
INACTIVE	•

"OR" ARRAY – (D-Type)



Tn STATUS	CODE
ACTIVE (SET)	A

Tn STATUS	CODE
INACTIVE (RESET)	•

Notes on page 313.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

LOGIC IMPLEMENTATION (Continued)

OUTPUT MACRO CELL CONFIGURATIONS

OUTPUT MACRO CELL CONFIGURATION	PROGRAMMING CODES			
	REGISTER SELECT FUSE	OMC CONFIGURATION FUSE	POLARITY FUSE	CLOCK FUSE
Combinatorial I/O with Buried D-type register				
External clock source	A	•	H or L	A
P-term clock source	A	•	H or L	•
Combinatorial I/O with Buried J-K type register				
External clock source	•	•	H or L	A
P-term clock source	•	•	H or L	•
Registered Output (D-type) with feedback				
External clock source	A	A	N/A	A
P-term clock source	A	A	N/A	•
Registered Output (J-K type) with feedback				
External clock source	•	A	N/A	A
P-term clock source	•	A	N/A	•
Registered Input (Clocked Preload) with feedback				
External clock source	A	A or •[5]	Optional[5]	A
P-term clock source	A	A or •[5]	Optional[5]	•

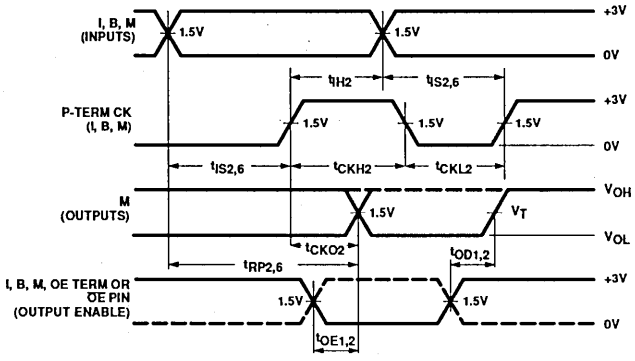
OUTPUT ENABLE CONTROL[8] CONFIGURATION	OUTPUT CONTROL FUSES		CONTROL SIGNAL
	OE CONTROL FUSE	En FUSES	
OMC controlled by /OE pin	A	A	
Output Enabled			Low
Output Disabled			High
OMC controlled by P-term	•	A or 0	
Output Enabled			High
Output Disabled			Low
Output always Enabled	A	0	Not Applicable

NOTES:
1. This is the initial (unprogrammed) state of the device.
2. Any gate will be unconditionally inhibited if both the TRUE and COMPLEMENT fuses are left intact.
3. To prevent oscillations, this state is not allowed for Complement Array fuse pairs that are coupled to active product terms.
4. The OMC Configuration fuse must be programmed as Combinatorial I/O in order to make use of the Polarity Option.
5. Regardless of the programmed state of the OMC Configuration fuse, an OMC can be used as a Registered Input. Note that the Load Control P-term must be asserted Active-High.
6. Output must be disabled.
7. Program code definitions:
   - A = Active (unprogrammed fuse)
   - 0, • = Inactive (programmed fuse)
   - – = Don't Care (both TRUE and COMPLEMENT fuses unprogrammed)
   - H = Active-High connection
   - L = Active-Low connection
8. OE control for B_0 and B_1 (Pins 10 and 11) is from the AND array only.

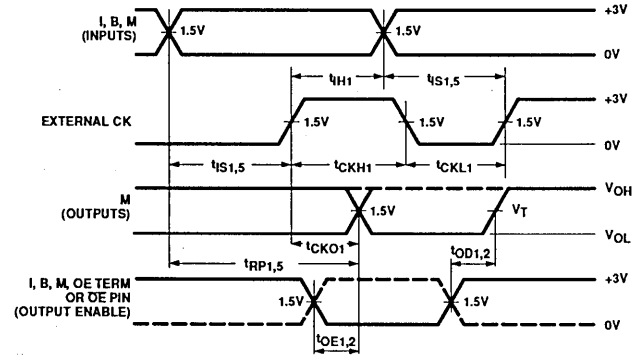
CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

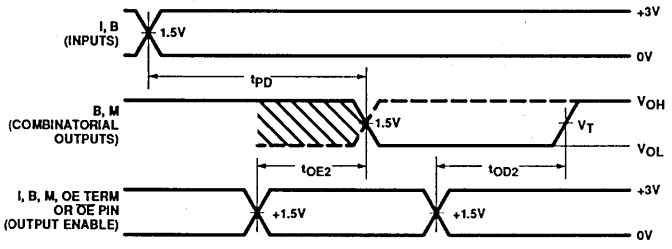
TIMING DIAGRAMS



Flip-Flop Outputs with P-term Clock



Flip-Flop Outputs with External Clock



Gated Outputs

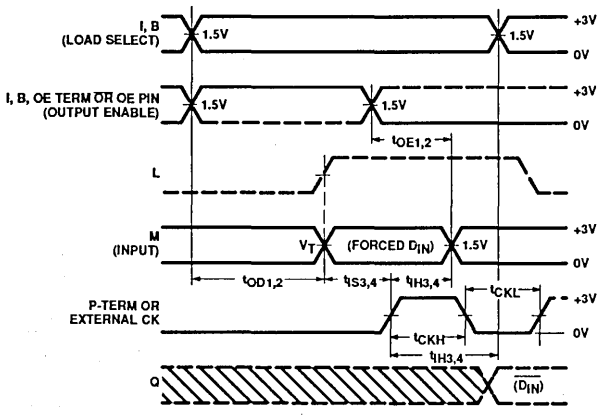
TIMING DEFINITIONS

SYMBOL	PARAMETER
f_{CK1}	Clock Frequency; External Clock
f_{CK2}	Clock Frequency; P-term Clock
t_{CKH1}	Width of Input Clock Pulse; External Clock
t_{CKH2}	Width of Input Clock Pulse; P-term Clock
t_{CKL1}	Interval between Clock pulses; External Clock
t_{CKL2}	Interval between Clock Pulses; P-term Clock
t_{CKO1}	Delay between the Positive Transition of External Clock and when M Outputs become valid.
t_{CKO2}	Delay between the Positive Transition of P-term Clock and when M Outputs become valid.
t_{RP1}	Delay between beginning of Valid Input and when the M outputs become Valid when using External Clock.
t_{RP2}	Delay between beginning of Valid Input and when the M outputs become Valid when using P-term Clock.
t_{RP3}	Delay between beginning of Valid Input and when the M outputs become Valid when using Preload Inputs (from M pins) and External Clock.
t_{RP4}	Delay between beginning of Valid Input and when the M outputs become valid when using Preload inputs (from M pins) and P-term Clock.
t_{RP5}	Delay between beginning of Valid Input and when the M outputs become Valid when using Complement Array and External clock.
t_{RP6}	Delay between beginning of Valid Input and when the M outputs become Valid when using Complement Array and P-term Clock.
f_{MAX1}	Minimum guaranteed Operating Frequency; Dedicated Clock
f_{MAX2}	Minimum guaranteed Operating Frequency; P-term Clock
f_{MAX3}	Minimum guaranteed Operating Frequency using Preload; Dedicated Clock (M pin to M pin)
f_{MAX4}	Minimum guaranteed Operating Frequency using Preload; P-term Clock (M pin to M pin)
f_{MAX5}	Minimum guaranteed Operating Frequency using Complement Array; Dedicated Clock
f_{MAX6}	Minimum Operating Frequency using Complement Array; P-term Clock
t_{IH1}	Required delay between positive transition of External Clock and end of valid input data.

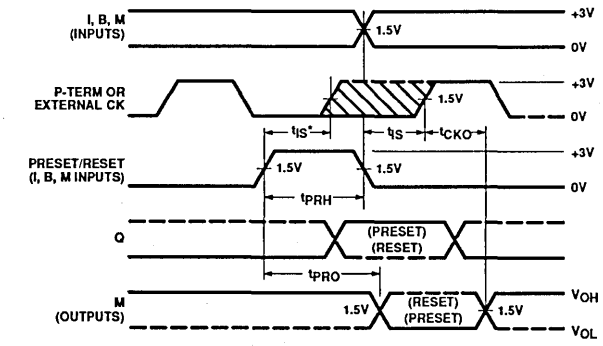
CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

TIMING DIAGRAMS (Continued)

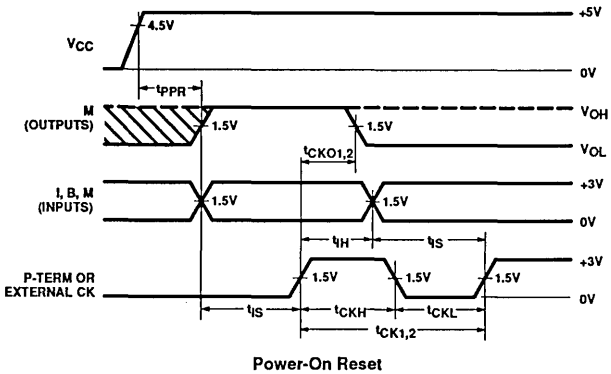


Flip-Flop Input Mode (Preload)



*Preset and Reset functions override Clock. However, M outputs may glitch with the first positive Clock Edge if t_{IS} cannot be guaranteed by the user.

Asynchronous Preset/Reset



Power-On Reset

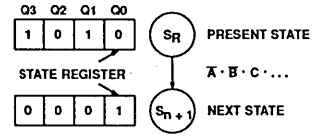
TIMING DEFINITIONS (Continued)

SYMBOL	PARAMETER
t_{IH2}	Required delay between positive transition of P-term Clock and end of valid input data.
t_{IH3}	Required delay between positive transition of External Clock and end of valid input data when using Preload Inputs (from M pins).
t_{IH4}	Required delay between positive transition of P-term Clock and end of valid input data when using Preload Inputs (from M pins).
t_{IS1}	Required delay between beginning of valid input and positive transition of External Clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of P-term Clock input.
t_{IS3}	Required delay between beginning of valid Preload input (from M pins) and positive transition of External Clock.
t_{IS4}	Required delay between beginning of valid Preload input (from M pins) and positive transition of P-term Clock input.
t_{IS5}	Required delay between beginning of valid input through Complement Array and positive transition of External Clock.
t_{IS6}	Required delay between beginning of valid input through Complement Array and positive transition of P-term Clock input.
t_{OE1}	Delay between beginning of Output Enable signal (Low) from /OE pin and when Outputs become valid.
t_{OE2}	Delay between beginning of Output Enable signal (High or Low) from OE P-term and when Outputs become valid.
t_{OD1}	Delay between beginning of Output Enable signal (HIGH) from /OE pin and when Outputs become disabled.
t_{OD2}	Delay between beginning of Output Enable signal (High or Low) from OE P-term and when Outputs become disabled.
t_{PD}	Delay between beginning of valid input and when the Outputs become valid (Combinatorial Path).
t_{PRH}	Width of Preset/Reset Pulse.
t_{PRO}	Delay between beginning of valid Preset/Reset Input and when the registered Outputs become Preset ("1") or Reset ("0").
t_{PPR}	Delay between V_{CC} (after power-up) and when flip-flops become Reset to "0". Note: Signal at Output (M pin) will be inverted.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

LOGIC FUNCTION



SET Q_0: $J_0 = (\overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$
$K_0 = 0$

RESET Q_1: $J_1 = 0$
$K_1 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$

HOLD Q_2: $J_2 = 0$
$K_2 = 0$

RESET Q_3: $J_3 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$
$K_3 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$

NOTE:
Similar logic functions are applicable for D mode flip-flops.

FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L_n	CK_n	P_n	R_n	J	K	Q	M
H								Hi-Z
L	X	X	X	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

NOTES:
1. Positive Logic:
   $J\text{-}K = T_0 + T_1 + T_2 + \ldots + T_{31}$
   $T_n = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \ldots)$
2. ↑ denotes transition for Low to High level.
3. X = Don't care
4. * = Forced at M_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J/K flip-flop (Diagnostic mode).

PLC42VA12 UNPROGRAMMED STATE

A factory shipped unprogrammed device is configured such that all cells are in a conductive state.

The following are:

ACTIVE:
- OR array logic terms
- Output Macro Cells M1 – M8;
  - D-type registered outputs
- External clock path
- Inputs: B_0, B_1, M_0, M_9

INACTIVE:
- AND array logic and control terms (except flip-flop mode control term, F_C)
- Bidirectional I/O (B_0, B_1);
  - Inputs are active. Outputs are 3-Stated via the OE P-terms, D_0 and D_1.
- Output Macro Cells M_0 and M_9;
  - Bidirectional I/O, 3-Stated via the OE P-terms, DM_0 and DM_9. The inputs are active.
- P-term clocks
- Complement Array
- J-K Flip-Flop mode

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC42VA12 devices are such that erasure begins to occur upon exposure to light with wavelength shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PLC42VA12 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC42VA12 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

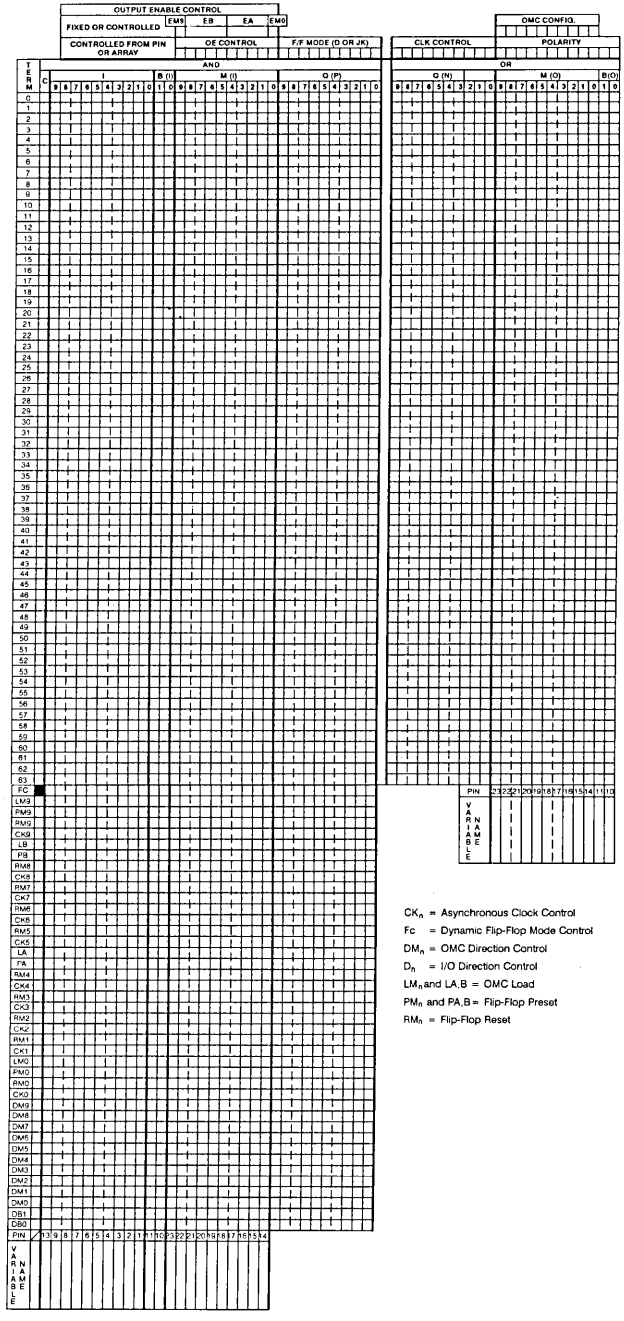
The recommended erasure procedure for the PLC42VA12 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000μW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12000μW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retentions exceeds 20 years.

CMOS programmable multi-function PLD ($42 \times 105 \times 12$)

PLC42VA12

PROGRAM TABLE

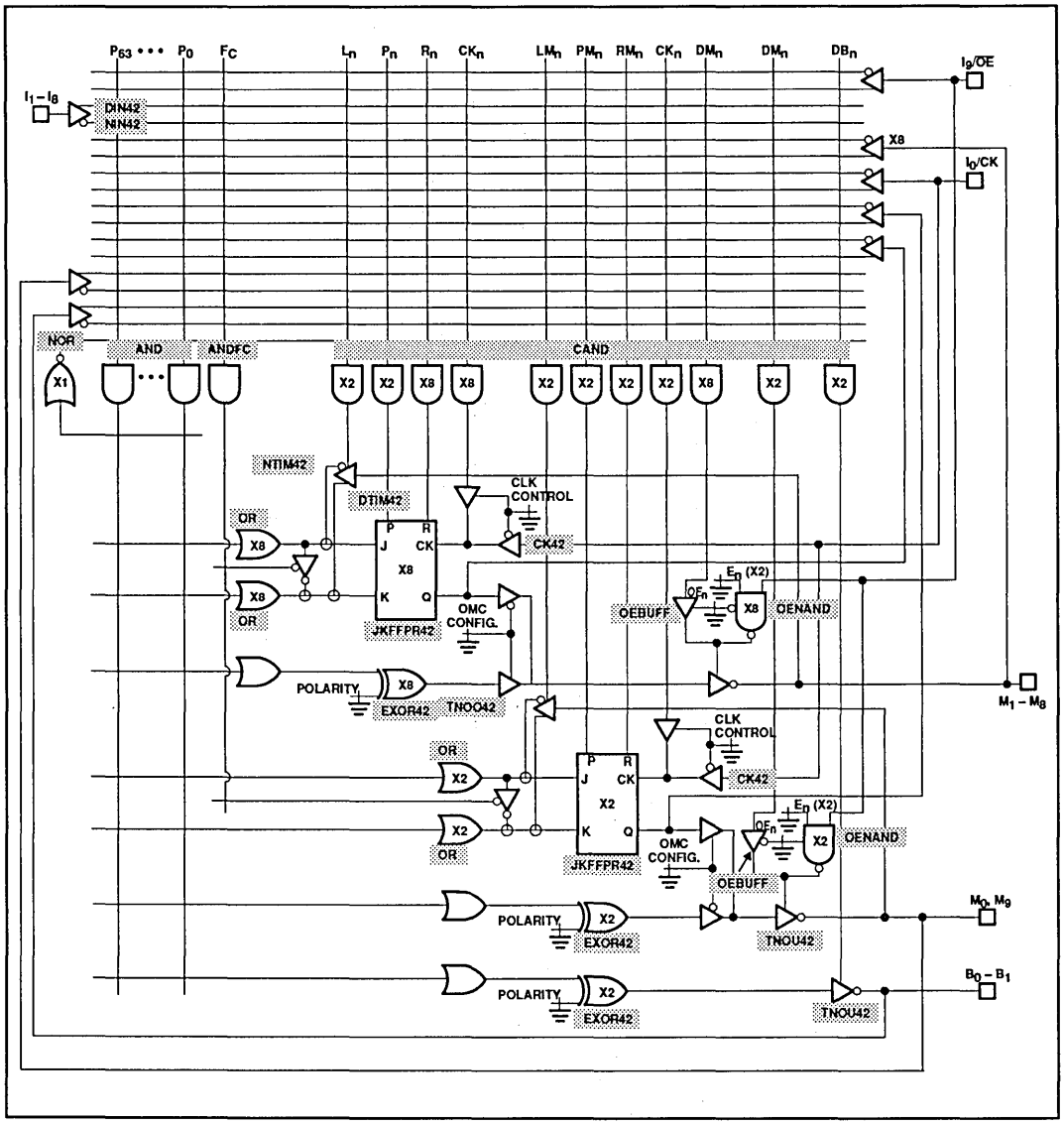


CK_n = Asynchronous Clock Control
Fc = Dynamic Flip-Flop Mode Control
DM_n = OMC Direction Control
D_n = I/O Direction Control
LM_n and LA,B = OMC Load
PM_n and PA,B = Flip-Flop Preset
RM_n = Flip-Flop Reset

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

SNAP RESOURCE SUMMARY DESIGNATIONS



Philips Components–Signetics

Document No.	853–1382
ECN No.	00079
Date of Issue	July 30, 1990
Status	Product Specification
Programmable Logic Devices	

PLC415–16

CMOS programmable logic sequencer (17 × 68 × 8)

DESCRIPTION

The PLC415–16 PLD is a CMOS Programmable Logic Sequencer of the Mealy type. The PLC415–16 is a pin-for-pin compatible, functional superset of the PLS105 and PLUS405 Bipolar Programmable Logic Sequencer devices.

The PLC415 is ideally suited for high density, power sensitive controller functions. The Power Down feature provides true CMOS standby power levels of less than 100µA. The EPROM-based process technology supports operating frequencies of 16 to 20MHz. The PLC415–16 has been designed to accept both CMOS and TTL input levels to facilitate logic integration in almost any system environment.

The PLC415 architecture has been tailored for state machine functions. Both arrays are programmable, thus providing full interconnectability. Any one or all of the 64 AND transition terms can be connected to any (or all) of the 8 buried state and 8 output registers.

Two clock sources enable the design of 2 state machines on one chip. Separate INIT functions and Output Enable functions for each are controllable either from the array or from an external pin. The J-K flip-flops provide the added flexibility of the toggle function which is indeterminate on S-R flip-flops. The programmable Initialization feature supports asynchronous initialization of the state machine to any user defined pattern.

The unique Complement Array feature supports complex ELSE transition statements with a single product term. The PLC415–16 has 2 Complement Arrays which allows the user to design two independent complement functions. This is particularly useful if two state machines have been implemented on one chip.

FEATURES

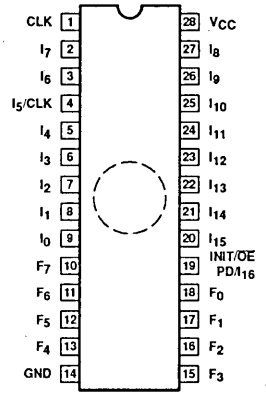
- Pin-for-Pin compatible, functional superset of PLS105/A and PLUS405 Logic Sequencers
- Zero standby power of less than 100µA (worst case)
  - Power dissipation at f_{MAX} = 80mA (worst case)
- CMOS and TTL compatible
- Programmable asynchronous Initialization and OE functions
  - Controllable from AND Array or external source
- 17 input variables
- 8 output functions
- 68 Product Terms
  - 64 transition terms
  - 4 control terms
- 8-bit State Register
- 8-bit Output Register
- 2 Transition Complement Arrays
- Multiple clocks
- Diagnostic test modes features for access to state and output registers
- Power-on preset of all registers to "1"
- J-K flip-flops
  - Automatic Hold states
- Security Fuse
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Arbitration functions
- Sequential circuits
- Security locking systems
- Counters
- Shift Registers

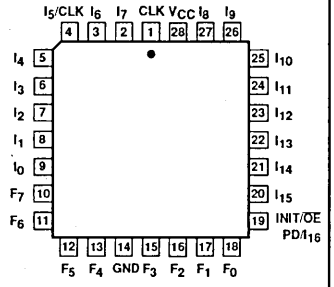
PIN CONFIGURATIONS

N, FA Packages



NOTE:
N = Plastic DIP (600mil-wide)
FA = Ceramic DIP with Quartz window (600mil-wide)

A Package



NOTE:
A = Plastic Leaded Chip Carrier

CMOS programmable logic sequencer ($17 \times 68 \times 8$)

PLC415–16

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK1	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers. Pin 1 only clocks P_{0-3} and F_{0-3} if Pin 4 is also being used as a clock.	Active-High (H)
2, 3, 5–9, 26–27 20–22	$I_0 - I_4, I_7, I_6$ $I_8 - I_9$ $I_{13} - I_{15}$	**Logic Inputs:** The 12 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/Low (H/L)
4	I_5/CLK2	**Logic Input/Clock:** A user programmable function:	
		• **Logic Input:** A 13th external logic input to the AND array, as above.	Active-High/Low (H/L)
		• **Clock:** A 2nd clock for the State Registers P_{4-7} and Output Registers F_{4-7}, as above. Note that input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using Pin 4 as a Clock.	Active-High (H)
23	I_{12}	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I_{12} is held at +11V, device outputs $F_0 - F_7$ reflect the contents of State Register bits $P_0 - P_7$. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
24	I_{11}	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I_{11} is held at +11V, device outputs $F_0 - F_7$ become direct inputs for State Register bits $P_0 - P_7$; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0 - F_7$ into the State Register bits $P_0 - P_7$. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
25	I_{10}	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I_{10} is held at +11V, device outputs $F_0 - F_7$ become direct inputs for Output Register bits $Q_0 - Q_7$; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0 - F_7$ into the Output Register bits $Q_0 - Q_7$. The contents of each State Register remains unaltered.	Active-High/Low (H/L)
10–13 15–18	$F_0 - F_7$	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register Bits $Q_0 - Q_7$, when enabled. When I_{12} is held at +11V, $F_0 - F_7 = (P_0 - P_7)$. When I_{11} is held at +11V, $F_0 - F_7$ become inputs to State Register bits $P_0 - P_7$. When I_{10} is held at +11V, $F_0 - F_7$ become inputs to Output Register bits $Q_0 - Q_7$.	Active-High (H)
19	INIT/$\overline{OE}$ I_{16}/PD	**External Initialization, External /OE, PD or I_{16}:** A user programmable function: Only one of the four options below may be selected. Note that both Initialization and /OE options are alternately available via the AND array. (P-terms INA, INB, OEA, and OEB.)	
		• **External Initialization:** Provides an asynchronous Preset to logic "1" or Reset to logic "0" of any or all State and Output Registers, determined individually on a register-by-register basis. INIT overrides the clock, and when held High, clocking is inhibited. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the INIT pulse goes Low. See timing diagrams for t_{NVCK} and t_{VCK}. Note that if the External Initialization option is selected, I_{16} is disabled automatically via the design software and the Power Down and External OE options are not available. Internal OE is available via P-Terms OEA and/or OEB. This option can be selected for one or both banks of registers.	Active-High (H)
		• **External Output Enable:** Provides an Output Enable/Disable function for Output Registers. Note that if the External OE option is selected, I_{16} is disabled automatically via the design software and the Power Down and External INIT options are not available. Internal INIT is available via P-terms INA and/or INB. This option can be selected for one or both banks of registers.	Active-Low (L)
		• **Power Down:** When invoked, provides a Power Down (zero power) mode. The contents of all Registers is retained, despite the toggling of the Inputs or the clocks. To obtain the lowest possible power level, all Inputs should be static and at CMOS input levels. Note that if the PD options is selected, I_{16} is disabled automatically via the design software and the External INIT and External OE options are not available. Internal INIT is available via P-terms INA and/or INB and Internal OE is available via P-terms OEA and/or OEB.	Active-High (H)
		• **Logic Input:** The 17th external logic input to the AND array as above. Note that when the I_{16} option is selected, the Power Down, External /OE and External INIT are not available. Internal OE and Internal INIT are available from P-Terms OEA/OEB and INA/INB, respectively.	Active-High/Low (H/L)

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

TRUTH TABLE [1, 2, 3, 4, 5]

V_{CC}	OPTION										
	INIT	$\overline{OE}$	I_{10}	I_{11}	I_{12}	CK	J	K	Q_P	Q_F	F
	H		X	X	X	X	X	X	H/L	H/L	Q_F
	X		+11V	X	X	↑	X	X	Q_P	L	L
	X		+11V	X	X	↑	X	X	Q_P	H	H
	X		X	+11V	X	↑	X	X	L	Q_F	L
	X		X	+11V	X	↑	X	X	H	Q_F	H
	X		X	X	+11V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
+5V		H	X	X	X	X	X	X	Q_P	Q_F	Hi-Z
		X	+11V	X	X	↑	X	X	Q_P	L	L
		X	+11V	X	X	↑	X	X	Q_P	H	H
		X	X	+11V	X	↑	X	X	L	Q_F	L
		X	X	+11V	X	↑	X	X	H	Q_F	H
		L	X	X	+11V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	L	L	X	X	X	X	X	X	H	H	H

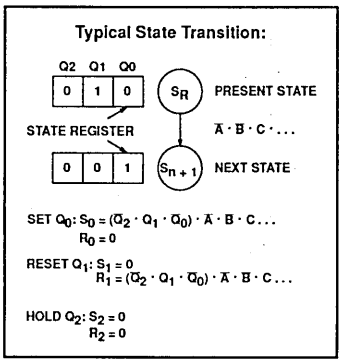
NOTES:
1. Positive Logic:
   S/R (or J/K) = $T_0 + T_1 + T_2 + \ldots T_{63}$
   $T_n = (C_0, C_1)(I_0, I_1, I_2, \ldots)(P_0, P_1 \ldots P_7)$
2. ↑ denotes transition from Low-to-High level.
3. X = Don't Care (≤5.5V)
4. H/L implies that either a High or a Low can occur, depending upon user-programmed Initialization selection (each State and Output Register individually programmable).
5. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

A factory-shipped virgin device contains all fusible links intact, such that:

1. INIT/$\overline{OE}$/PD/I_{16} is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user-programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are inactive (0).
3. All J/K flip-flop inputs are disabled (0).
4. The Complement Arrays are inactive.
5. Clock 1 is connected to all State and Output Registers.

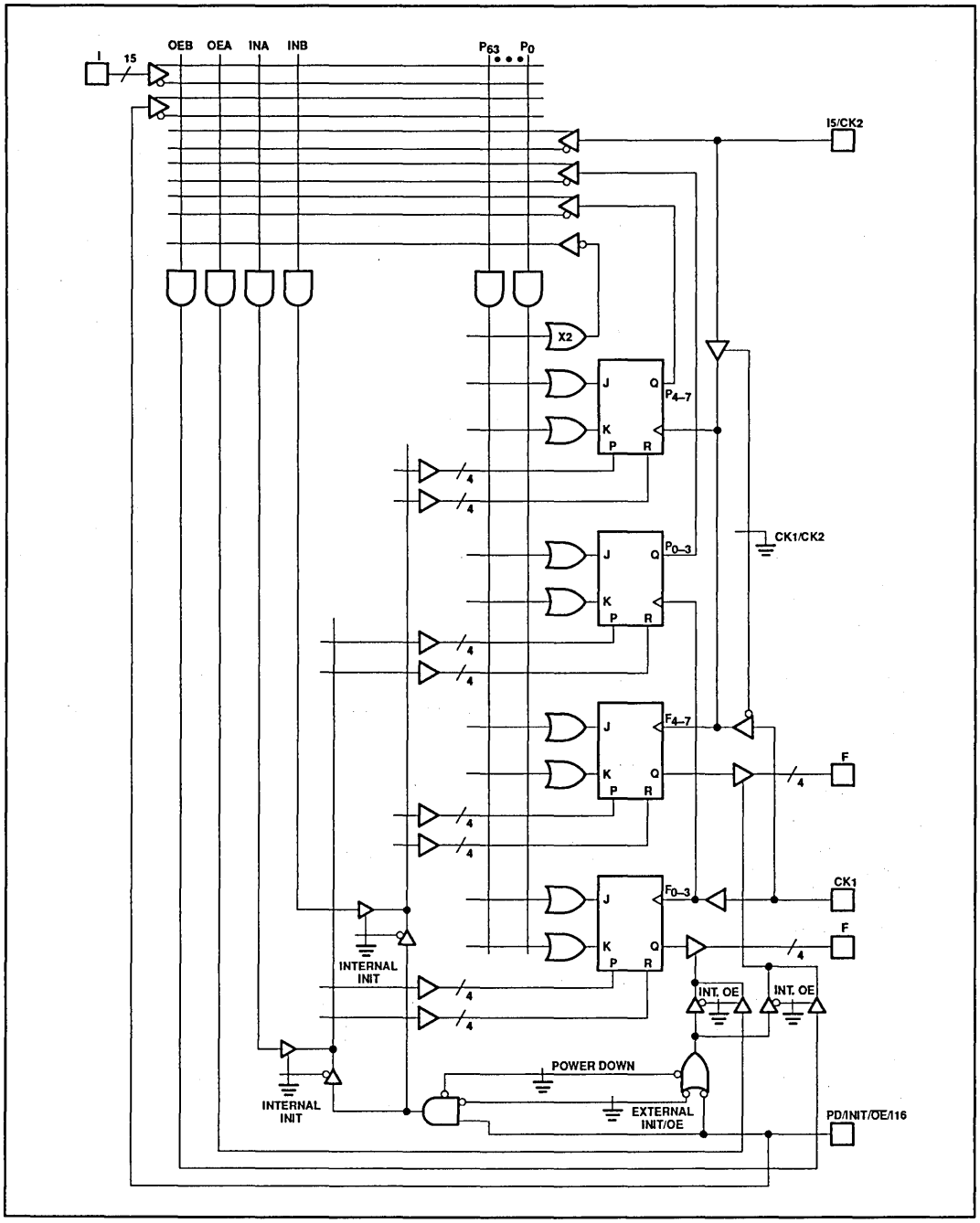
LOGIC FUNCTION



CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

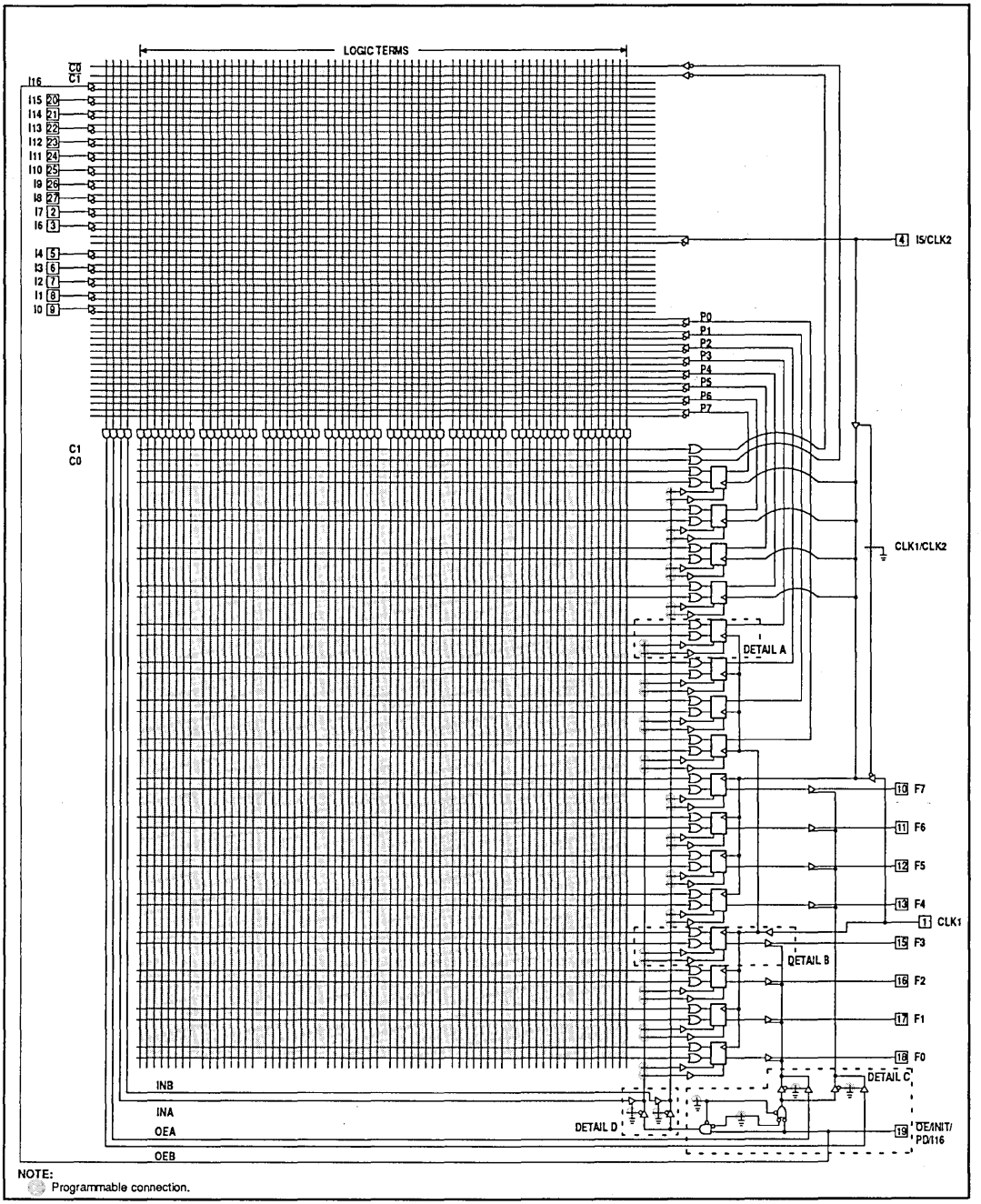
FUNCTIONAL DIAGRAM



CMOS programmable logic sequencer ($17 \times 68 \times 8$)

PLC415–16

LOGIC DIAGRAM

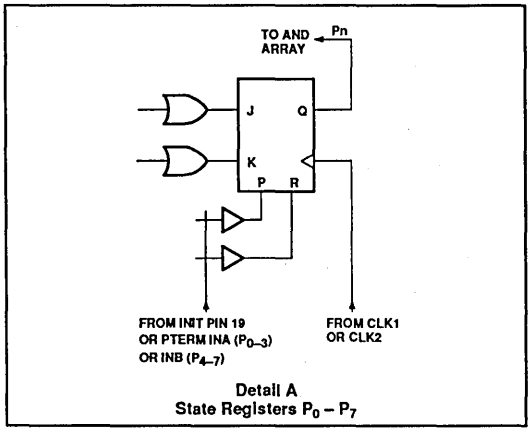


NOTE:
Programmable connection.

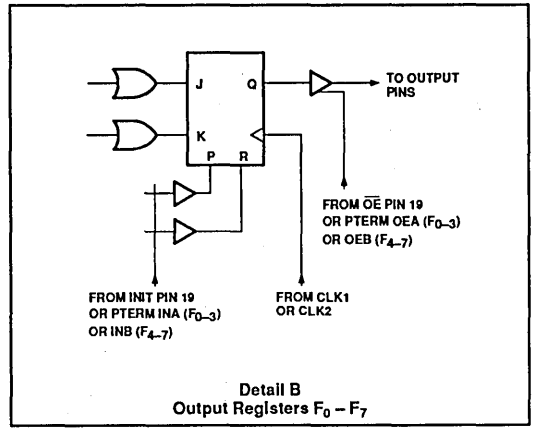
CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

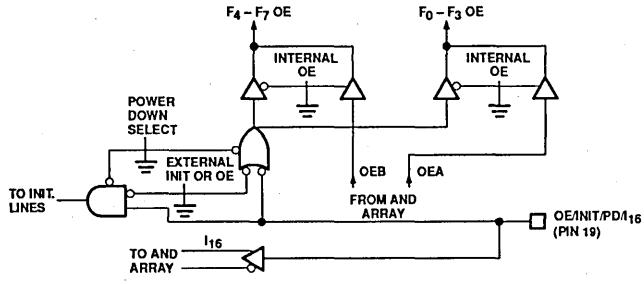
DETAILS FOR PLC415–16 LOGIC DIAGRAM



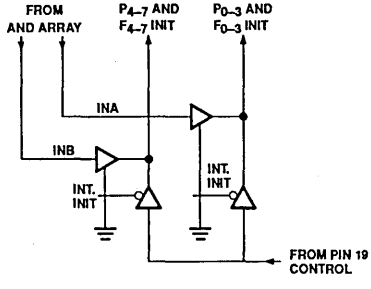
Detail A
State Registers $P_0 - P_7$



Detail B
Output Registers $F_0 - F_7$



Detail C
Pin 19 Options: $\overline{OE}$, Initialization, Power Down and Input 16

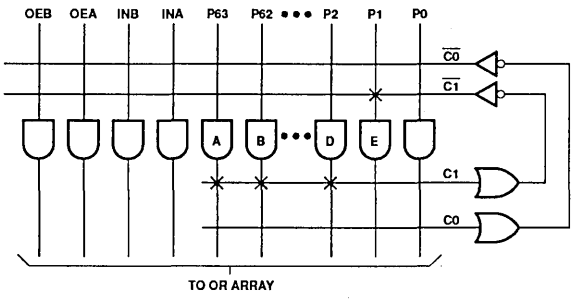


Detail D
Internal and External Initialization

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

DETAILS FOR PLC415–16 LOGIC DIAGRAM (Continued)



Complement Array Detail

The Complement Array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A • /B • /C) and ($\overline{A + B + C}$) are equivalent, you will begin to see the value of this single term NOR array.

The Complement Array is a single OR gate with inputs from the AND array. The output of the Complement Array is inverted and fed back to the AND array (NOR). The output of the array will be Low if any one or more of the AND terms connected to it are active (High). If, however, all the connected terms are inactive (Low), which is a classic unknown state, the output of the Complement Array will be High.

Consider the Product Terms A, B and D that represent defined states. They are also connected to the input of the Complement Array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to Product Term E, which could be used in turn to reset the state machine to a known state. Without the Complement Array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, such an approach can be prohibitive, both in terms of time and wasted resources.

Note that the PLC415–16 has 2 Complement Arrays which allow the user to design 2 independent Complement functions. This is particularly useful if 2 independent state machines have been implemented on one device.

Note that use of the Complement Array adds an additional delay path through the device. Please refer to the AC Electrical Characteristics for details.

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

ORDERING INFORMATION

DESCRIPTION	OPERATING FREQUENCY	ORDER CODE
28-Pin Ceramic DIP with window; Reprogrammable (600mil-wide)	f_{MAX} = 16MHz	PLC415–16FA
28-Pin Plastic DIP; One-Time Programmable (600mil-wide)	f_{MAX} = 16MHz	PLC415–16N
28-Pin Plastic Leaded Chip Carrier; One-Time Programmable (450mil-wide)	f_{MAX} = 16MHz	PLC415–16A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION		LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]							
V_{IL}	Low	V_{CC} = MIN		–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX		2.0		V_{CC} + 0.3	V
Output voltage[2]							
V_{OL}	Low	V_{CC} = MIN I_{OL} = 16mA				0.5	V
V_{OH}	High	I_{OH} = –3.2mA		2.4			V
Input current							
I_{IL}	Low	V_{IN} = GND				–10	µA
I_{IH}	High	V_{IN} = V_{CC}				10	µA
Output current							
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC} V_{OUT} = GND				10 –10	µA µA
I_{OS}	Short-circuit[3, 6]	V_{OUT} = GND				–130	mA
I_{CCSB}	V_{CC} supply current with PD asserted[7]	V_{CC} = MAX V_{IN} = 0 or V_{CC}			50	100	µA
I_{CC}	V_{CC} supply current Active[4, 5]	I_{OUT} = 0mA	at f = 1MHz			55	mA
	(TTL or CMOS Inputs)	V_{CC} = MAX	at f = MAX			80	mA
Capacitance							
C_I	Input	V_{CC} = 5V V_{IN} = 2.0V			12		pF
C_B	I/O	V_B = 2.0V			15		pF

NOTES:
1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short-circuit should not exceed one second. Test one at a time.
4. Tested with TTL input levels: V_{IL} = 0.45V, V_{IH} = 2.4V. Measured with all inputs and outputs switching.
5. Refer to Figure 1, I_{CC} vs Frequency (worst case).
6. Refer to Figure 2 for Δt_{PD} vs output capacitance loading.
7. The outputs are automatically 3-Stated when the device is in the Power Down mode. To achieve the lowest possible current, the inputs and clocks should be at CMOS static levels.

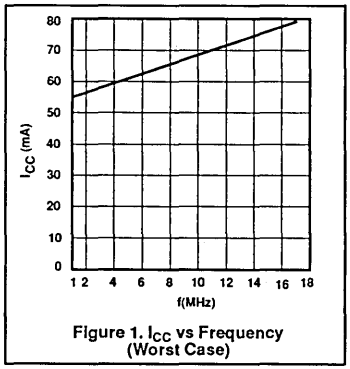


Figure 1. I_{CC} vs Frequency (Worst Case)

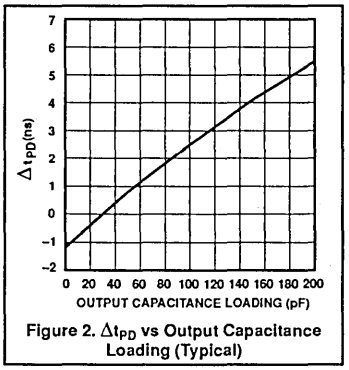


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

AC ELECTRICAL CHARACTERISTICS

$R_1 = 252\Omega$, $R_2 = 178\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock High	CK+	CK–	30pF	25	10		ns
t_{CKL}	Clock Low	CK–	CK+	30pF	25	10		ns
t_{INITH}	Initialization Input pulse	INIT+	INIT–	30pF	20			ns
Set–up time								
t_{IS1}	Input	(I) +/–	CK+	30pF	38	25		ns
t_{IS2}[1]	Input through Complement array	(I) +/–	CK+	30pF	60	40		ns
t_{ISPD}	Power Down Setup (from PD pin)	PD+	CK+	30pF	38	15		ns
t_{ISPU}	Power Up Setup (from PD pin)	PD–	First Valid CK+	30pF	38	30		ns
t_{VS}[1]	Power on Preset Setup	V_{CC}+	CK–	30pF	0			ns
t_{VCK1}	Clock resume (after INIT) when using INIT pin (pin 19)	INIT–	CK–	30pF	10	–5		ns
t_{VCK2}[1]	Clock resume (after INIT) when using P-term INIT (from AND array)	(I) +/–	CK–	30pF	20	8		ns
t_{NVCK1}	Clock lockout (before INIT) when using INIT pin (pin 19)	CK–	INIT–	30pF	10	–3		ns
t_{NVCK2}[1]	Clock lockout (before INIT) when using P-term INIT (from AND array)	CK–	INIT–	30pF	0	–5		ns
Propagation delays								
t_{CKO}	Clock to Output	CK+	(F) +/–	30pF		15	22	ns
t_{PDZ}	Power Down to outputs off	PD+	Outputs Off	5pF		25	30	ns
t_{PUA1}	Power Up to outputs Active with dedicated Output Enable	PD–	Outputs Active	30pF		20	35	ns
t_{PUA2}[1]	Power Up to outputs Active with P-term Output Enable[1]	PD–	Outputs Active	30pF		37	55	ns
t_{IHPU}	Last valid clock to Power Down delay (Hold)	Last Valid Clock	PD+	30pF	25	15		ns
t_{IHPD}	First valid clock cycle before Power Up	Beginning of First Valid Clock Cycle	PD–	30pF	0	–25		ns
t_{OE1}[3]	Output Enable: from /OE pin	OE–	Output Enabled	30pF		15	30	ns
t_{OE2}[1]	Output Enable; from P–term	(I) +/–	Output Enabled	30pF		25	40	ns
t_{OD1}[3]	Output Disable; from /OE pin	OE+	Output Disabled	5pF		20	30	ns
t_{OD2}[3]	Output Disable; from P-term	(I) +/–	Output Disabled	5pF		30	40	ns
t_{INIT1}	INIT to output when using INIT pin	INIT+	(F) +/–	30pF		22	35	ns
t_{INIT2}	INIT to output when using P-term INIT	(I) +/–	(F) +/–	30pF		35	45	ns
t_{PPR}[1]	Power-on Preset (F_n = 1)	V_{CC} +	(F) +	30pF			15	ns
t_{RP1}	Registered operating period; (t_{IS1} + t_{CKO1})	(I) +/–	(F) +/–	30pF		40	60	ns
t_{RP2}[1]	Registered operating period with Complement Array (t_{IS2} + t_{CKO1})	(I) +/–	(F) +/–	30pF		55	75	ns

Notes on following page

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

AC ELECTRICAL CHARACTERISTICS (Continued)

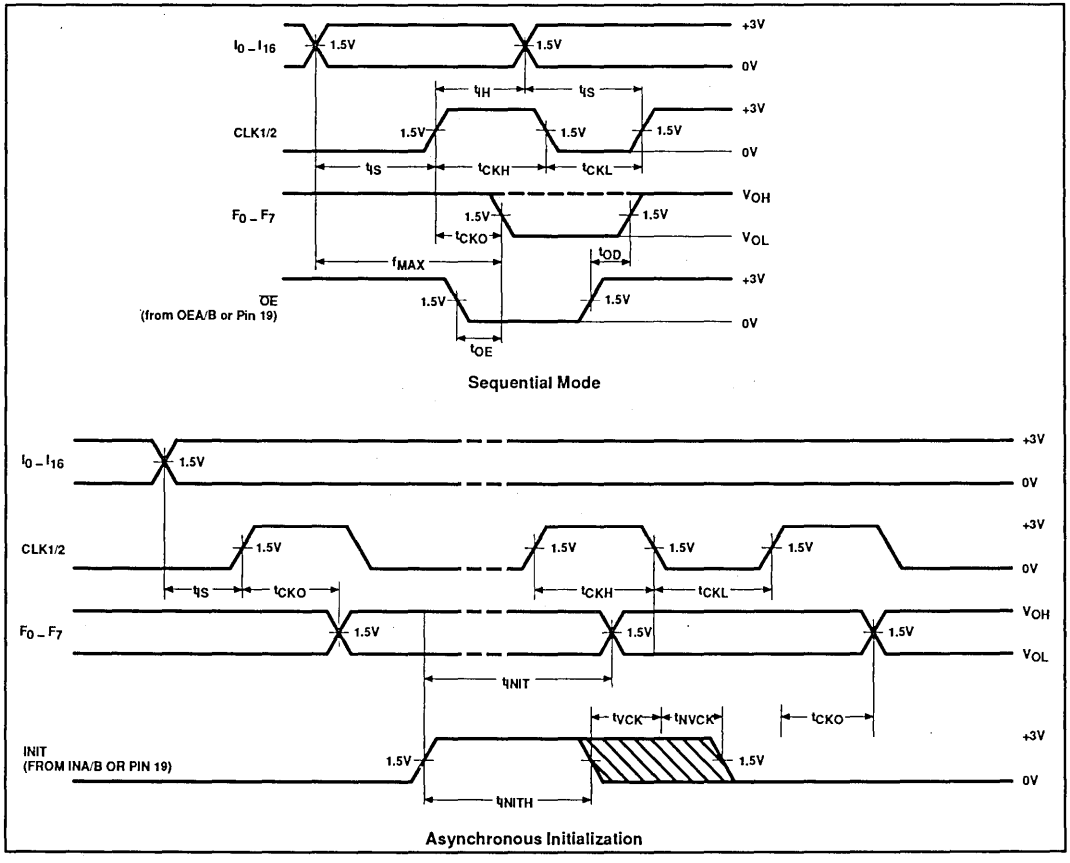
$R_1 = 252\Omega$, $R_2 = 178\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
Hold time								
t_{IH}	Input Hold	CK+	(F) +/–	30pF		–10	0	ns
Frequency of operation								
f_{CLK}[1]	Clock (toggle) frequency	C+	C+	30pF	20	50		MHz
f_{MAX1}	Registered operating frequency ($t_{IS1} + t_{CKO1}$)	(I) +/–	(F) +/–	30pF	16.7	25		MHz
f_{MAX2}	Registered operating frequency with Complement Array ($t_{IS2} + t_{CKO1}$)	(I) +/–	(F) +/–	30pF	13.3	18.2		MHz

NOTE:

1. Not 100% tested, but guaranteed by design/characterization.
2. All propagation delays and setup times are measured and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TIMING DIAGRAMS

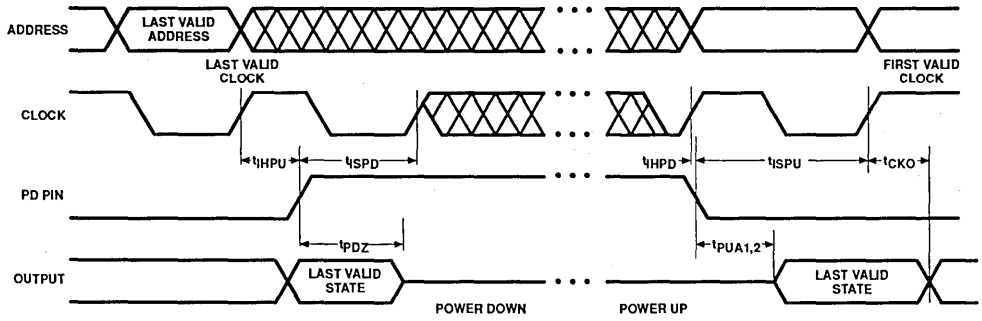


CMOS programmable logic sequencer (17 × 68 × 8)

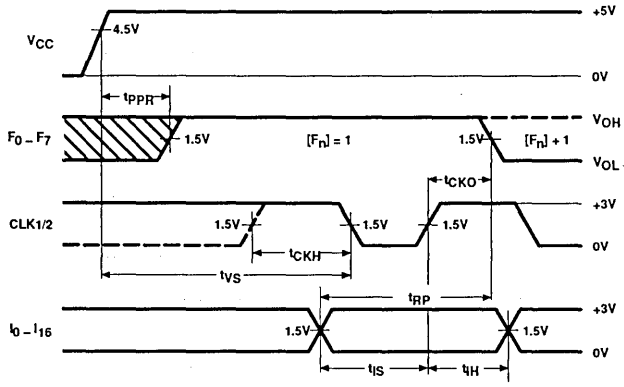
PLC415–16

The PLC415–16 has a unique power down feature that is ideal for power sensitive controller and state machine applications. During idle periods, the PLC415 can be powered down to a near zero power consumption level of less than 100 micro Amps. Externally controlled from Pin 19, the power down sequence first saves the data in all the State and Output registers. In order to insure that the last valid states are saved, there are certain hold times associated with the first and last valid clock edges and the Power Down input pulse. The Outputs are then automatically 3-Stated and power consumption is reduced to a minimum. Once in the power down mode, any or all of the inputs, including the clocks, may be toggled without the loss of data. To obtain the lowest possible power level, the inputs should be at static CMOS input levels during the power down period.

TIMING DIAGRAMS (Continued)

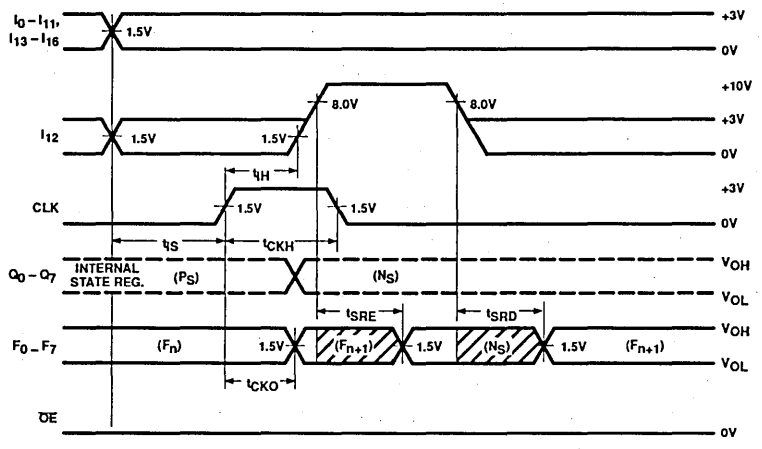


Power Down Enable and Disable

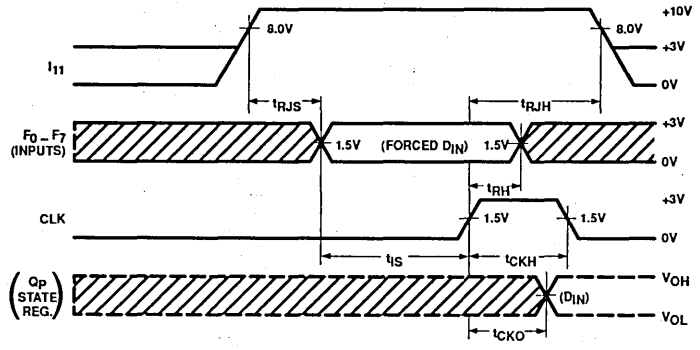


Power-On Preset

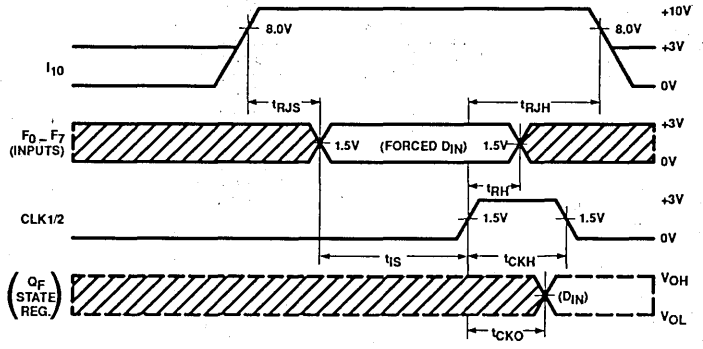
TIMING DIAGRAMS (Continued)



Diagnostic Mode—State Register Outputs



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—Output Register Input Jam

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

TIMING DEFINITIONS

SYMBOL	PARAMETER
f_{CLK}	Minimum guaranteed toggle frequency of the clock (from Clock HIGH to Clock HIGH).
$f_{MAX1,2}$	Minimum guaranteed operating frequency.
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{RP1}	Minimum guaranteed operating period – when not using Complement Array.
t_{RP2}	Minimum guaranteed operating period – when using Complement Array.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with outputs enabled).
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IHPD}	Required delay between the positive transition of the beginning of the first valid clock cycle to the beginning of Power Down LOW to insure that the last valid states are intact and that the next positive transition of the clock is valid.
t_{IHPU}	Required delay between the positive transition of the last valid clock and the beginning of Power Down HIGH to insure that last valid states are saved.
t_{INITH}	Width of initialization input pulse.
t_{INIT1}	Delay between positive transition of Initialization and when Outputs become valid when using external INIT control (from pin 19).
t_{INIT2}	Delay between positive transition of Initialization and when outputs become valid when using internal INIT control (from P-terms INA and INB).
t_{ISPD}	Required delay between the beginning of Power Down HIGH (from pin 19) and the positive transition of the next clock to insure that the clock edge is not detected as a valid Clock and that the last valid states are saved.

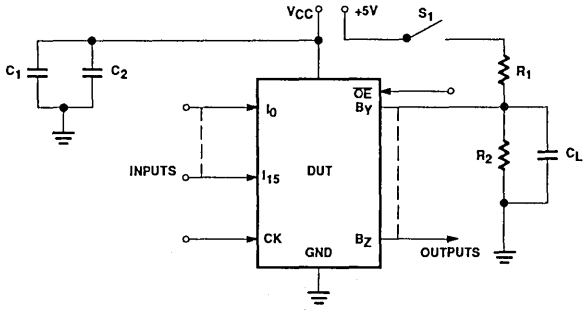
SYMBOL	PARAMETER
t_{ISPU}	Required delay between the beginning of Power Down LOW and the positive transition of the first valid clock.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{NVCK1}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization when using external INIT control (from pin 19) to guarantee that the clock edge is not detected as a valid negative transition.
t_{NVCK2}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization, when using the internal INIT control (from P-terms INA and INB), to guarantee that the clock edge is not detected as a valid negative transition.
t_{OD1}	Delay between beginning of Output Enable High and when Outputs are in the OFF-State, when using external OE control (from pin 19).
t_{OD2}	Delay between beginning of Output Enable High and when outputs are in the OFF-State when using internal OE control (from P-terms OEA and OEB).
t_{OE1}	Delay between beginning of Output Enable Low and when Outputs become valid when using external OE control from pin 19.
t_{OE2}	Delay between beginning of Output Enable Low and when outputs become valid when using internal OE control (from P-terms OEA and OEB).
t_{PDZ}	Delay between beginning of Power Down HIGH and when outputs are in OFF–State and the circuit is "powered down".

SYMBOL	PARAMETER
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
$t_{PUA1,2}$	Delay between beginning of Power Down LOW and when outputs become Active (valid) and the circuit is "powered up". See AC Specifications.
t_{RH}	Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
t_{RJH}	Required delay between positive transition of Clock and end of inputs I_{11} or I_{10} transition to State and Output Register Input Jam Diagnostic Modes, respectively.
t_{RJS}	Required delay between when inputs I_{11} or I_{10} transition to State and Output Register Input Jam Diagnostic Modes, respectively, and when the output pins become available as inputs.
t_{SRD}	Delay between input I_{12} transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I_{12} transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VCK1}	Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding the first valid clock pulse when using external INIT control (pin 19).
t_{VCK2}	Required delay between the negative transition of the Asynchronous Initialization and the negative transition of the clock preceding the first valid clock pulse when using internal INIT control (from P–terms INA and INB).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.

CMOS programmable logic sequencer (17 × 68 × 8)

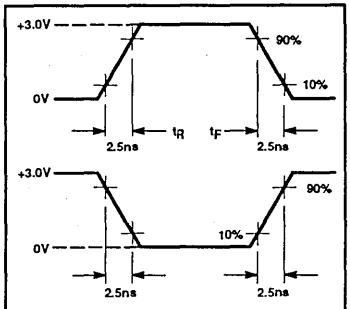
PLC415–16

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

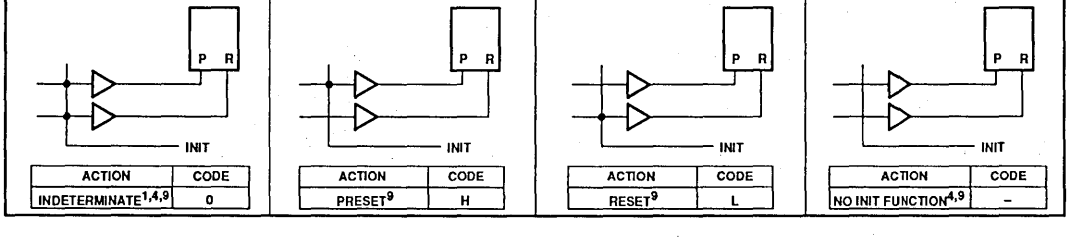
The PLC415–16 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLC415–16 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC415–16 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, each logic variable (I, B, P, S, T, etc.) from the logic equations if assigned a symbol. TRUE, COMPLEMENT, PRESET, RESET, OUTPUT ENABLE, INACTIVE, etc., symbols are defined below.

INITIALIZATION (PRESET/RESET)[11] OPTION – (P/R)



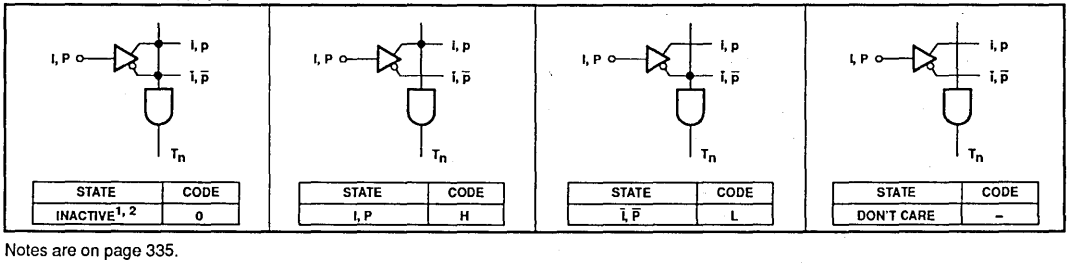
ACTION	CODE
INDETERMINATE[1,4,9]	0

ACTION	CODE
PRESET[9]	H

ACTION	CODE
RESET[9]	L

ACTION	CODE
NO INIT FUNCTION[4,9]	–

"AND" ARRAY – (I), (P)



STATE	CODE
INACTIVE[1, 2]	0

STATE	CODE
I, P	H

STATE	CODE
Ī, P̄	L

STATE	CODE
DON'T CARE	–

Notes are on page 335.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

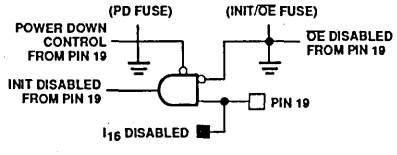
CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

LOGIC PROGRAMMING (Continued)

PIN 19 FUNCTION: POWER DOWN, INITALIZATION, $\overline{OE}$, OR INPUT

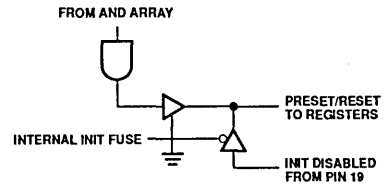
Power Down Mode



POWER DOWN FUSE	CODE
PIN 19 AS POWER DOWN	H[6]

EXTERNAL INIT/OE FUSE	CODE
EXTERNAL INIT/OE DISABLED	L

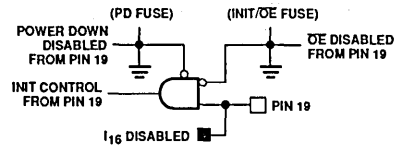
P-Term Initialization Control



INTERNAL INIT FUSES	CODE
P-TERM INIT CONTROL	H[7, 8]

POWER DOWN FUSE	CODE
POWER DOWN ENABLED OR DISABLED	H OR L

External Initialization Control

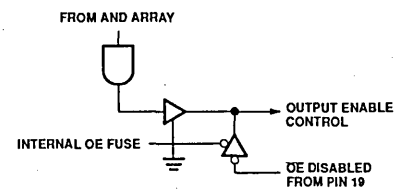


PD FUSE	CODE
POWER DOWN DISABLED	L[1]

EXTERNAL INIT/OE FUSE	CODE
PIN 19 AS EXTERNAL INIT	L[1]

INTERNAL INIT FUSES	CODE
P-TERM INIT ACTIVE OR INACTIVE	H OR L[7, 8]

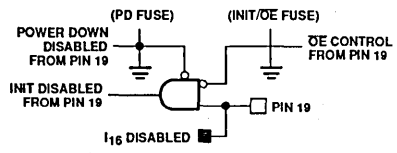
P-Term OE Control



INTERNAL OE FUSES	CODE
P-TERM OE CONTROL	H[7, 8]

POWER DOWN FUSE	CODE
POWER DOWN ENABLED OR DISABLED	H OR L

External Output Enable Control



PD FUSE	CODE
POWER DOWN DISABLED	L

EXTERNAL INIT/OE FUSE	CODE
PIN 19 AS EXTERNAL OE	H

INTERNAL INIT FUSES	CODE
P-TERM OE ACTIVE OR INACTIVE	H OR L[7, 8]

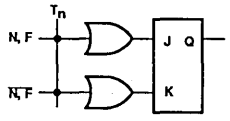
Notes are on page 335.

CMOS programmable logic sequencer (17 × 68 × 8)

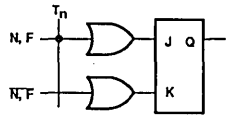
PLC415–16

LOGIC PROGRAMMING (Continued)

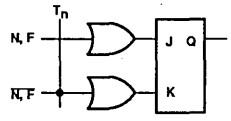
"OR" ARRAY – J-K FUNCTION – (N), (F)



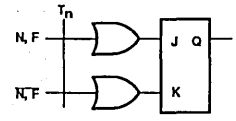
ACTION	CODE
TOGGLE[2]	0



ACTION	CODE
SET	H

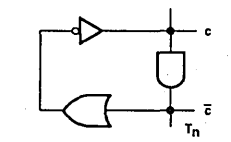


ACTION	CODE
RESET	L

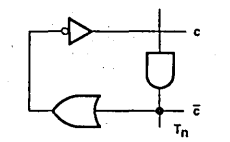


ACTION	CODE
DON'T CARE	–

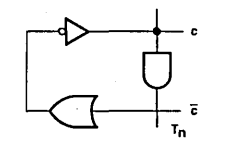
"COMPLEMENT" ARRAY – (C)



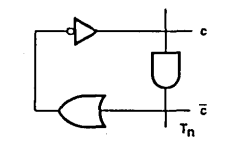
ACTION	CODE
INACTIVE[1, 3]	0



ACTION	CODE
GENERATE	A

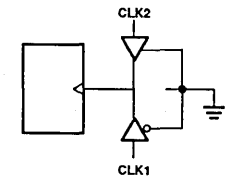


ACTION	CODE
PROPAGATE	•

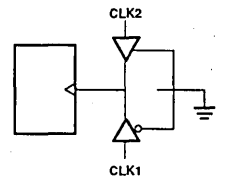


ACTION	CODE
TRANSPARENT	–

CLOCK OPTION – (CLK1/CLK2)



OPTION	CODE
CLK1 ONLY[1]	L



OPTION	CODE
CLK1 and CLK2[5]	H

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.
4. These states are not allowed when using PRESET/RESET option.
5. Input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using second clock option.
6. When using Power Down feature, INPUT 16 is automatically disabled via the design software.
7. If the internal (P–term) control fuse for INIT and/or OE is programmed as Active High, the associated External Control function will be permanently disabled, regardless of the state of the External INIT/OE fuse.
8. One internal control fuse exists for each group of 8 registers. P_{0-3} and F_{0-3} are banked together in one group, as are P_{4-7} and F_{4-7}. Control can be split between the INIT/OE pin (Pin 19) and P–terms INA, INB, OEA and OEB.
9. The PLC415–16 also has a power–up preset feature. This feature insures that the device will power–up in a known state with all register elements (State and Output Register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.
10. L = cell unprogrammed.
    H = cell programmed.
11. Inputs 10, 11 and 12 (pins 25, 24, & 23) can be used for supervoltage diagnostic mode tests. It is recommended that these inputs not be connected to product terms INA, INB, OEA or OEB if you intend to make use of the diagnostic modes due to the fact that the patterns associated with the internal INIT and OE control product terms may interfere with the diagnostic mode data loading and reading.

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

PROGRAM TABLE

AND

(Im, Ps)		(Cn)	
INACTIVE	O	INACTIVE	O
I, P	H	GENERATE	A
$\bar{I}$, $\bar{P}$	L	PROPAGATE	•
DON'T CARE	–	TRANSPARENT	–

OR (Ns, Fn)

TOGGLE	O
SET	H
RESET	L
NO CHANGE	–

OPTIONS

ENTERNAL INIT	L
EXTERNAL OE	H
INTERNAL INIT/OE ENABLED	H
INTERNAL INIT/OE DISABLED	L
POWER DOWN ENABLED	H
POWER DOWN DISABLED	L
CLOCK 1 ONLY	L
CLOCK 1 AND 2	H

INITIALIZATION	
PRESET	H
RESET	L
NO INIT	–
INDETERMINATE	O

PROJECT NAME _____ DATE _____ REVISION _____

CUSTOMER NAME _____ SIGNETICS PART # _____ CUSTOMER PART # _____

TERMS	COMP. ARRAY	INTERNAL INIT INB	INTERNAL INIT INA	INTERNAL OE OEB	INTERNAL OE OEA	EXTERNAL INIT/OE	CLK 1/1&2	PD	INITIALIZATION
	NOR	AND							OR
	Cn	INPUT (Im)					PRESENT STATE (Ps)		NEXT STATE (Ns) / OUTPUT (Fn)
	C1 C0	I16 I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0					P7 P6 P5 P4 P3 P2 P1 P0		N7 N6 N5 N4 N3 N2 N1 N0 / F7 F6 F5 F4 F3 F2 F1 F0
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									
51									
52									
53									
54									
55									
56									
57									
58									
59									
60									
61									
62									
63									
INB									
OEB									
INA									
OEA									
PIN NO.		19 20 21 22 23 24 25 26 27 2 3 4 5 6 7 8 9							10 11 12 13 15 16 17 18
PIN LABELS									

NOTES:

1. In the unprogrammed state all cells are conducting. Thus, the program table for an unprogrammed device would contain "0"s for all product terms (inactive) and initialization states (indeterminate). The default or unprogrammed state of all other options is "L".
2. Unused Cn, Im and Ps cells are normally programmed as Don't Care (–).
3. Unused product terms can be left blank (inactive) for future code modification.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC415 Series devices are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps has wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PLC415 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC415 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

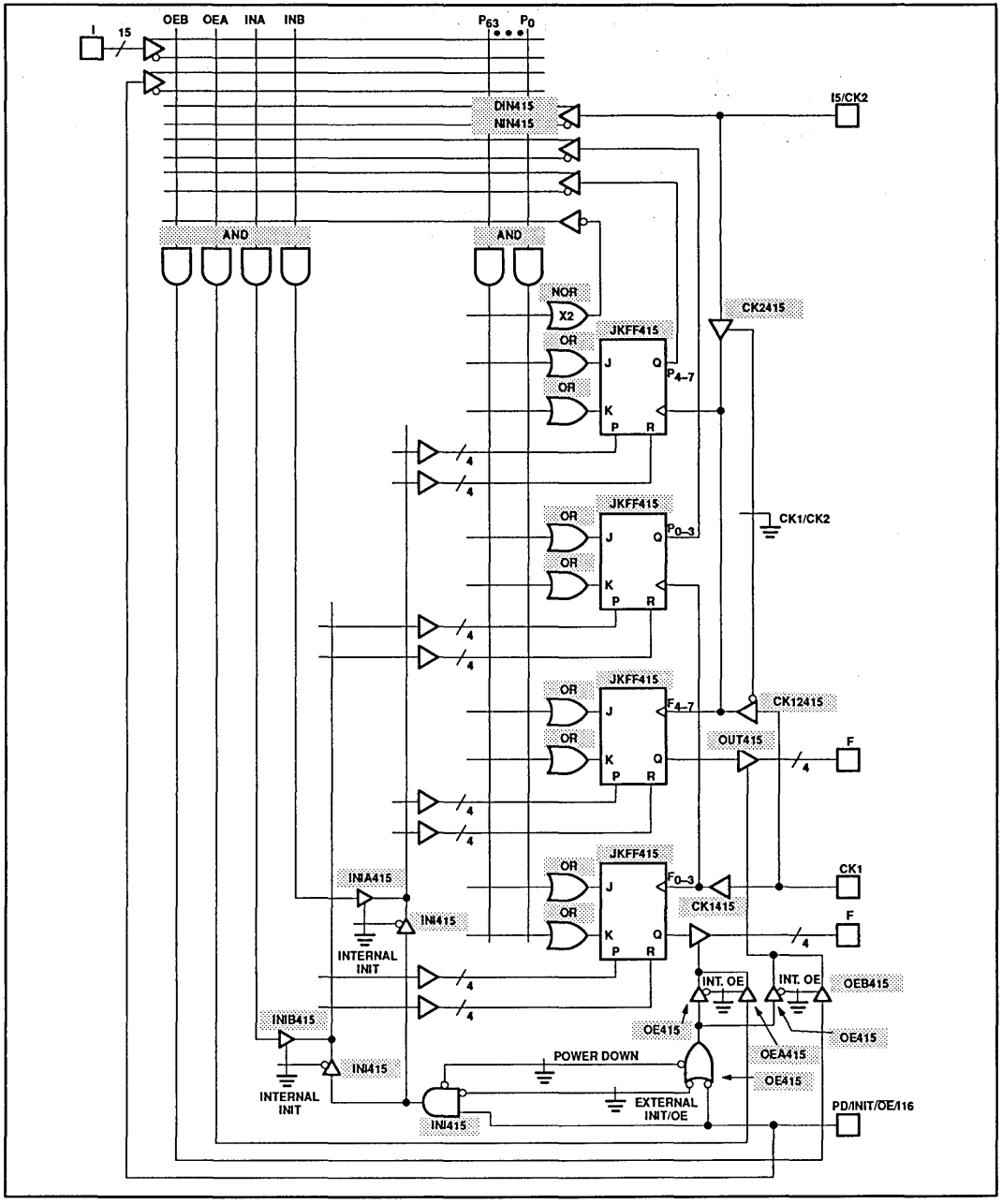
The recommended erasure procedure for the PLC415 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000μW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12000μW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415–16

SNAP RESOURCE SUMMARY DESIGNATIONS



Document No.	853–0310
ECN No.	97888
Date of Issue	October 16, 1989
Status	Product Specification
Programmable Logic Devices	

PLS105/A

Programmable logic sequencers ($16 \times 48 \times 8$)

DESCRIPTION

The PLS105and the PLS105A are bipolar Programmable Logic State machines of the Mealy type. They contain logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 6 Q_P, and 8 Q_F edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset option. all flip-flops are unconditionally preset to "1" during power turn on.

The AND array combines 16 external inputs $I_0 - I_{15}$ with six internal inputs P_{0-5}, which are fed back from the State Registers to form up to 48 transition terms (AND terms). All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse. Both True and Complement transition terms can be generated by optional use of the internal input variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to $\overline{\text{Output Enable}}$ function, as an additional user-programmable option.

Order codes are listed in the Ordering Information Table.

FEATURES

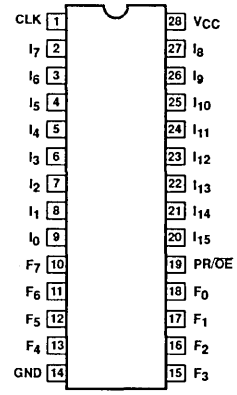
- PLS105
  - f_{MAX} = 13.9MHz
  - 20MHz clock rate
- PLS105A
  - f_{MAX} = 20MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked flip-flops
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to all "1" of internal registers
- Power dissipation: 600mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers

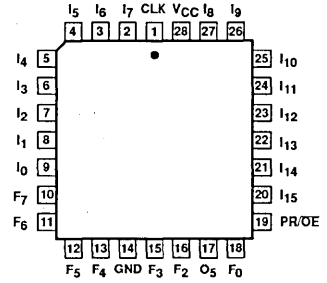
PIN CONFIGURATIONS

N Package



N = Plastic DIP (600mil-wide)

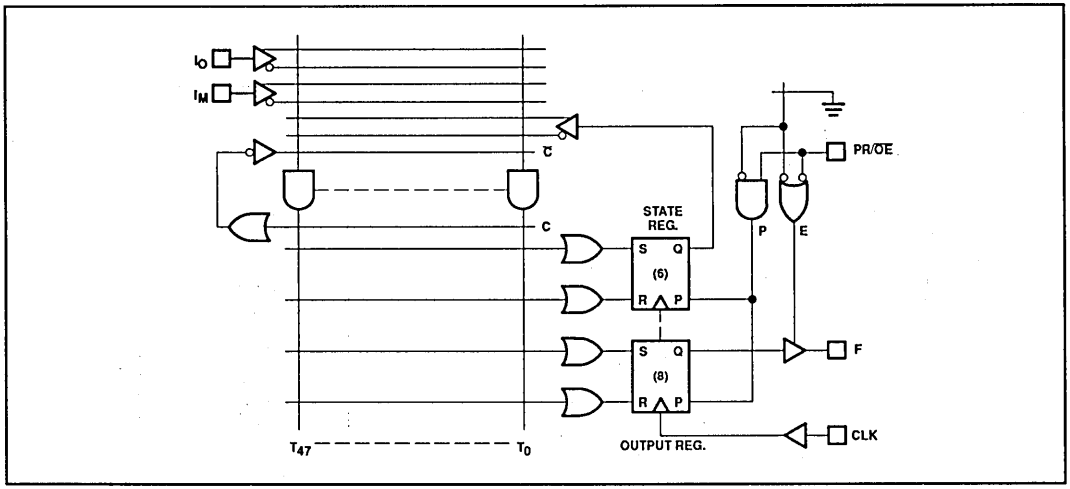
A Package



A = Plastic Leaded Chip Carrier

Programmable logic sequencers (16 × 48 × 8) — PLS105/A

FUNCTIONAL DIAGRAM

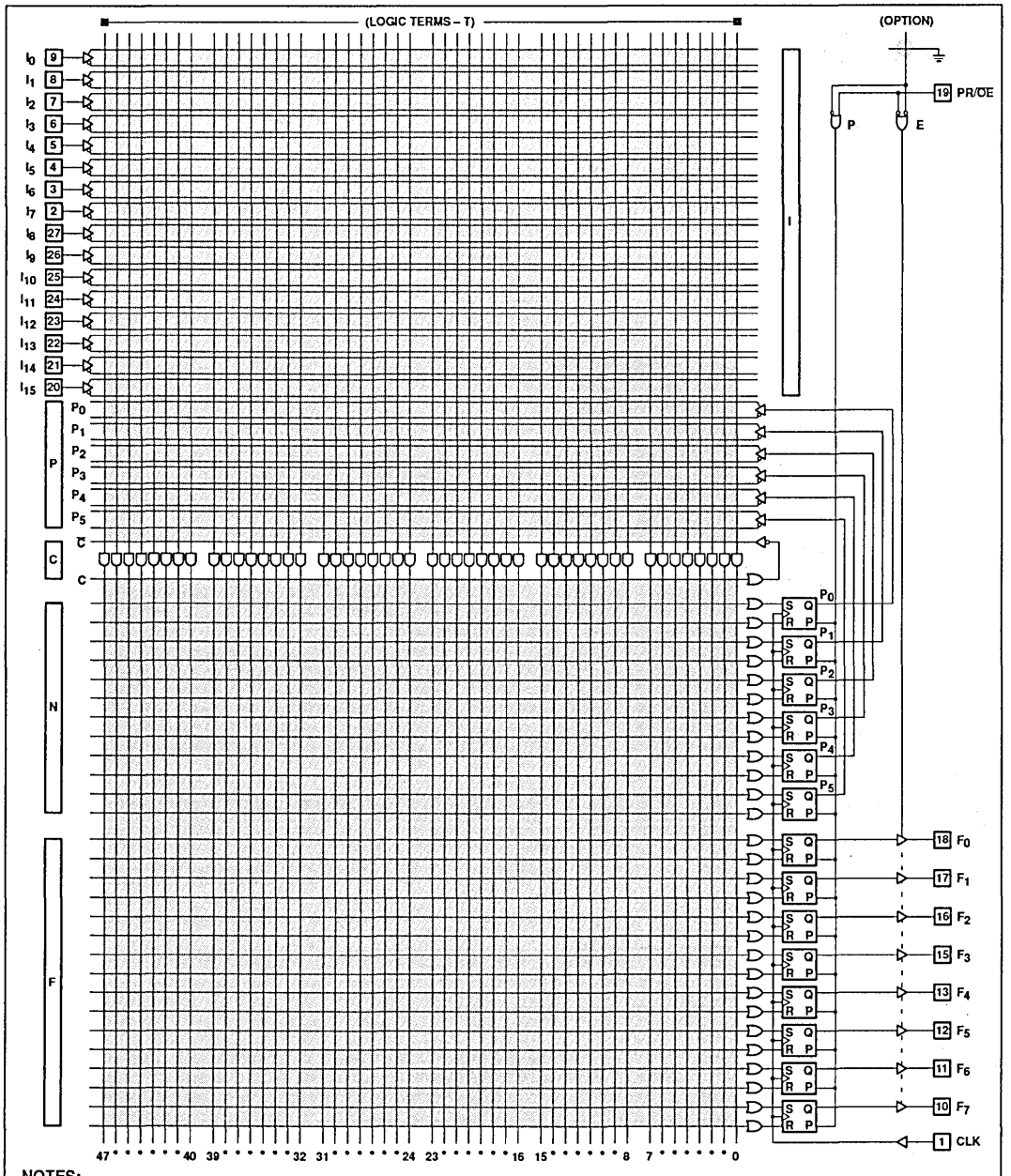


PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2 – 8 20 – 27	$I_1 - I_{15}$	**Logic Inputs:** The 15 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
9	I_0	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercised with standard TTL levels. When I_0 is held at +10V, device outputs F_{0-5} reflect the contents of State Register bits P_{0-5}. The contents each Output Register remains unaltered.	Active-High/Low
10 – 13 15 – 18	F_{0-7}	**Logic/Diagnostic Outputs:** Eight device outputs which normally reflect the contents of Output Register bits Q_{0-7}, when enabled. When I_0 is held at +10V, $F_{0-5} = (P_{0-5})$, and $F_{6,7}$ = Logic "1".	Active-High
19	PR/$\overline{OE}$	**Preset or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Preset:** Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and F_{0-7} are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an Output Enable function to all output buffers F_{0-7} from the Output Register.	Active-Low (L)

Programmable logic sequencers (16 × 48 × 8) PLS105/A

LOGIC DIAGRAM



NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ◌ Programmable connection.

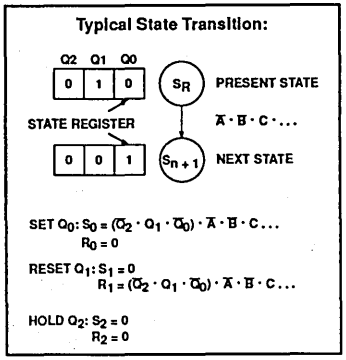
Programmable logic sequencers (16 × 48 × 8) — PLS105/A

TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I_0	CK	S	R	$Q_{P/F}$	F
+5V	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q_n	$(Q_P)_n$
	L		X	X	X	X	Q_n	$(Q_F)_n$
		H	*	X	X	X	Q_n	Hi–Z
		L	+10V	X	X	X	Q_n	$(Q_P)_n$
		L	X	X	X	X	Q_n	$(Q_F)_n$
		L	X	↑	L	L	Q_n	$(Q_F)_n$
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots + T_{47}$
   $T_n = C(I_0\, I_1\, I_2 \ldots)(P_0\, P_1 \ldots P_5)$
2. Either Preset (Active–High) or $\overline{\text{Output Enable}}$ (Active–Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. R = S = High is an illegal input condition.
5. * = H or L or +10V.
6. X = Don't Care (≤5.5V).

LOGIC FUNCTION



VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. PR/$\overline{OE}$ option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array pre–programmed with a standard test pattern.
   NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Signetics qualified programming equipment.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
28-Pin Plastic DIP (600mil-wide)	PLS105N, PLS105AN
28-Pin Plastic Leaded Chip Carrier	PLS105A, PLS105AA

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

Programmable logic sequencers (16 × 48 × 8) PLS105/A

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High[4]	I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
I_{IL}	Low (CK input)	V_{IN} = 0.45V		−50	−250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[6]	V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		120	180	mA
Capacitance[6]						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IL} applied to $\overline{OE}$ and a logic high stored, or with V_{IH} applied to PR.
5. Measured with a programmed logic condition for which the output is at a low logic level, and V_{IL} applied to PR/$\overline{OE}$ Output sink current is supplied through a resistor to V_{CC}.
6. Measured with V_{IH} applied to PR/$\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.
8. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (16 × 48 × 8) — PLS105/A

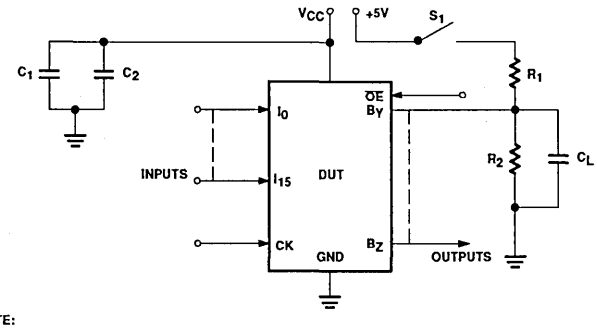
AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLS105			PLS105A			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width										
t_{CKH}	Clock[2] High	CK +	CK –	25	15		20	15		ns
t_{CKL}	Clock Low	CK –	CK +	25	15		20	15		ns
t_{CKP}	Clock period	CK +	CK +	50	30		40	30		ns
t_{PRH}	Preset pulse	PR +	PR –	25	15		25	15		ns
Setup time[3]										
$t_{IS1}A$	Input	Input ±	CK +	60			40			ns
$t_{IS1}B$	Input	Input ±	CK +	50			30			ns
$t_{IS1}C$	Input	Input ±	CK +	42			N/A			ns
$t_{IS2}A$	Input (through Complement Array)	Input ±	CK +	90			70			ns
$t_{IS2}B$	Input (through Complement Array)	Input	CK +	80			60			ns
$t_{IS2}C$	Input (through Complement Array)	Input	CK +	72			N/A			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		0	–10		ns
t_{PRS}	Preset	PR –	CK –	0	–10		0	–10		ns
Hold time										
t_{IH}	Input	CK +	Input ±	5	–10		5	–10		ns
Propagation delay										
t_{CKO}	Clock	CK +	Output ±		15	30		15	20	ns
t_{OE}	Output enable[4]	OE –	Output –		20	30		20	30	ns
t_{OD}	Output disable[4]	OE +	Output +		20	30		20	30	ns
t_{PR}	Preset	PR +	Output +		18	30		18	30	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10		0	10	ns
Frequency of operation[3]										
$f_{MAX}C$	Without Complement Array			13.9			20.0			MHz
$f_{MAX}C$	With Complement Array			9.8			12.5			MHz

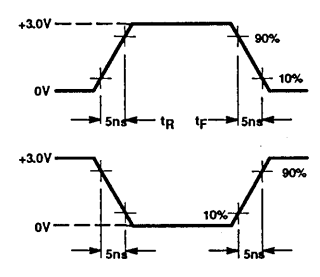
NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 30ns.
3. See "Speed vs. OR Loading" diagrams.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} – 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS

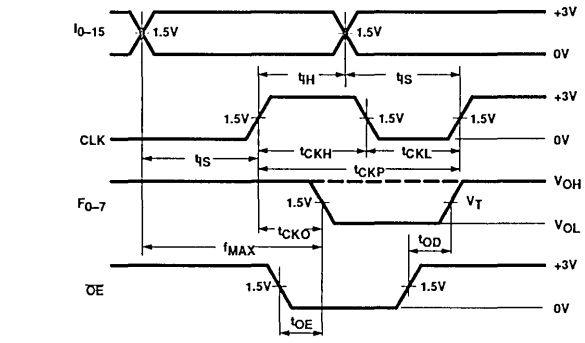


MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

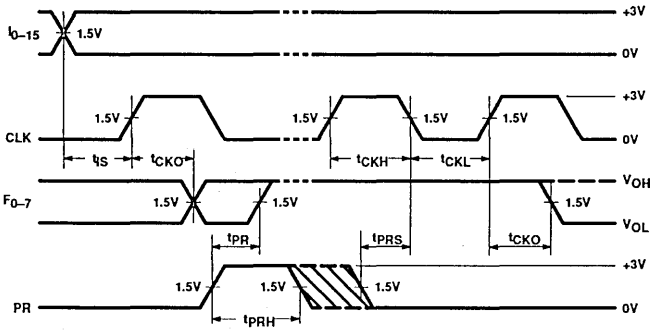
Input Pulses

Programmable logic sequencers (16 × 48 × 8) — PLS105/A

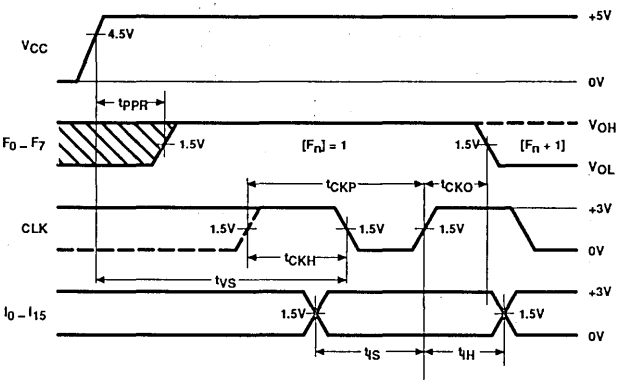
TIMING DIAGRAMS



Sequential Mode



Asynchronous Preset



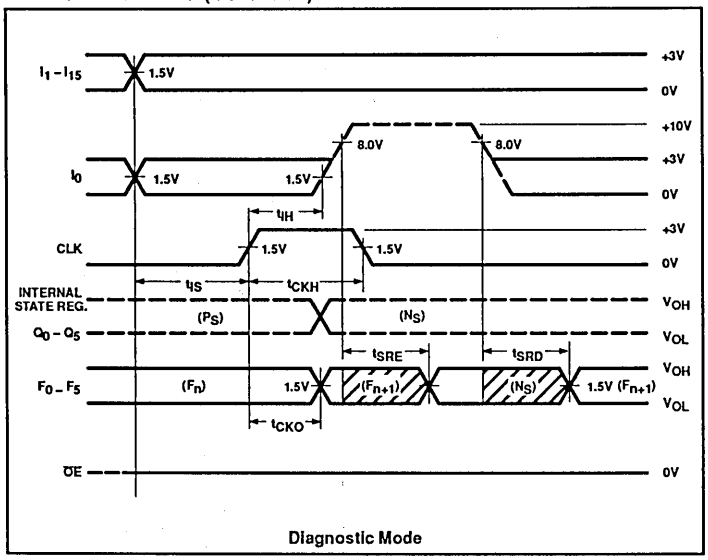
Power-On Preset

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed Clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND array).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and negative transition of Clock preceding first reliable clock pulse.
t_{IH}	Required delay between positive transition of Clock and end of valid input data.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with PR/OE Low).
t_{OE}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{SRE}	Delay between input I_0 transition to Diagnostic mode and when the outputs reflect the contents of the State Register.
t_{SRD}	Delay between input I_0 transition to Logic mode and when the outputs reflect the contents of the Output Register.
t_{PR}	Delay between positive transition of Preset and when outputs become valid at "1".
t_{PPR}	Delay between V_{CC} (after power-on) and when outputs become preset at "1".
t_{PRH}	Width of preset input pulse.
f_{MAX}	Minimum guaranteed operating frequency.

Programmable logic sequencers (16 × 48 × 8) — PLS105/A

TIMING DIAGRAMS (Continued)



Diagnostic Mode

SPEED VS. "OR" LOADING

The maximum frequency at which the PLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms T_n used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects t_{IS}, due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of t_{IS1} with the number of terms connected per OR.

The PLS105 AC electrical characteristics contain three limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 1). The first, t_{IS1A} is guaranteed for a device with 48 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 32 terms connected to any OR line. And t_{IS1C} is guranteed for a device with 24 terms conntected to any OR line.

The three other entries in the AC table, t_{IS2} A, B, and C are corresponding 48, 32, and 24 term limits when using the on–chip Complement Array.

The PLS105A AC electrical characteristics contain two limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 2). The first, t_{IS1A} is guaranteed for a device with 24 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 16 terms connected to any OR line.

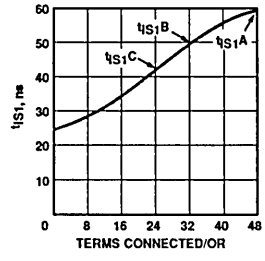


Figure 1. PLS105 t_{IS1} vs. Terms/OR Connected

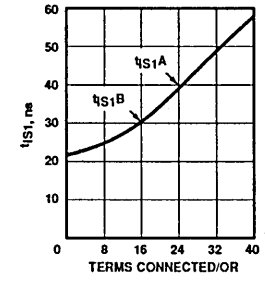


Figure 2. PLS105A t_{IS1} vs. Terms/OR Connected

The two other entries in the AC table, t_{IS2} A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of t_{IS} for a given application can be determined by identifying the OR line with the maximum number of T_n connections. This can be done by referring to the interconnect pattern in the PLS logic diagram, typically illustrated in Figure 3, or by counting the maximum number of "H" *or* "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 or 2 will yield the worst case t_{IS} and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

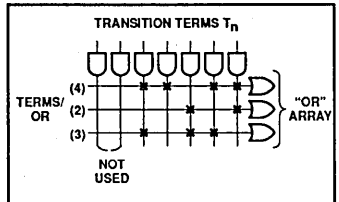


Figure 3. Typical OR Array Interconnect Pattern

Programmable logic sequencers (16 × 48 × 8) — PLS105/A

LOGIC PROGRAMMING

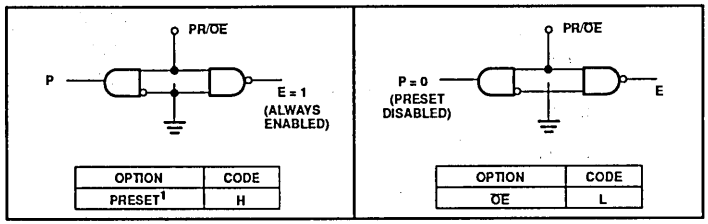
The PLS105/A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' AMAZE, Data I/O Corporation's ABEL and Logical Devices Inc.'s CUPL design software packages.

All packages allow Boolean and state equation entry formats. ABEL and CUPL also accept, as input, schematic capture format.

PLS105/A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Signetics AMAZE PLD design software (PTP module). AMAZE is available free of charge to qualified users.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PRESET/$\overline{OE}$ OPTION – (P/E)

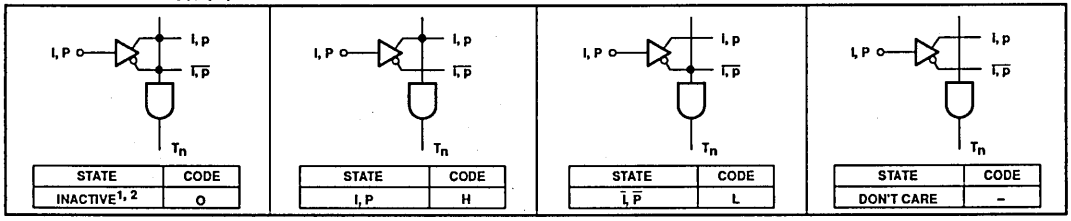


OPTION	CODE
PRESET[1]	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING:
The PLS105/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY – (I), (P)



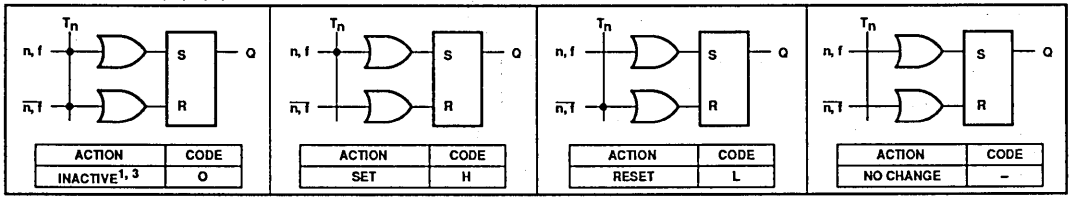
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I}, \overline{P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



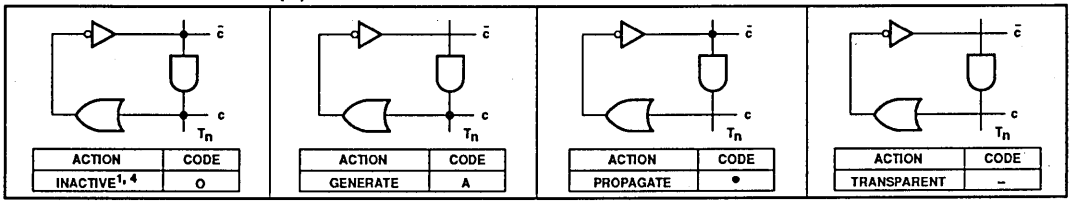
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I or P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencers (16 × 48 × 8) PLS105/A

PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
SIGNETICS DEVICE # CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND

INACTIVE	0	C_n
GENERATE	A	
PROPAGATE	•	
TRANSPARENT	—	

INACTIVE	0	I_m, P_s
I, P	H	
$\overline{I}$, $\overline{P}$	L	
DON'T CARE	—	

OR

INACTIVE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	—	

OPTION

PRESET	H	P/E
$\overline{OE}$	L	

	AND																	OPTION (P/E) OR														
TERM	C_n	INPUT (I_m)																PRESENT STATE (P_s)						REMARKS	NEXT STATE (Ns)						OUTPUT (Fr)							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0		5	4	3	2	1	0	7	6	5	4	3	2	1	0
0																																					
1																																					
2																																					
3																																					
4																																					
5																																					
6																																					
7																																					
8																																					
9																																					
10																																					
11																																					
12																																					
13																																					
14																																					
15																																					
16																																					
17																																					
18																																					
19																																					
20																																					
21																																					
22																																					
23																																					
24																																					
25																																					
26																																					
27																																					
28																																					
29																																					
30																																					
31																																					
32																																					
33																																					
34																																					
35																																					
36																																					
37																																					
38																																					
39																																					
40																																					
41																																					
42																																					
43																																					
44																																					
45																																					
46																																					
47																																					
PIN NO.		20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9														10	11	12	13	15	16	17	18
VARIABLE NAME																																					

NOTES:

1. The FPLS is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

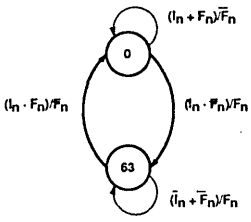
Programmable logic sequencers (16 × 48 × 8) — PLS105/A

TEST ARRAY

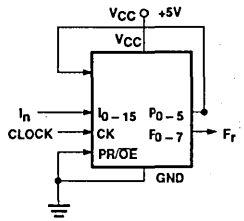
The PLS105/A may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the PLS105/A and applying the proper input sequence to $I_0 - I_{15}$ as shown in the test circuit timing diagram.



State Diagram



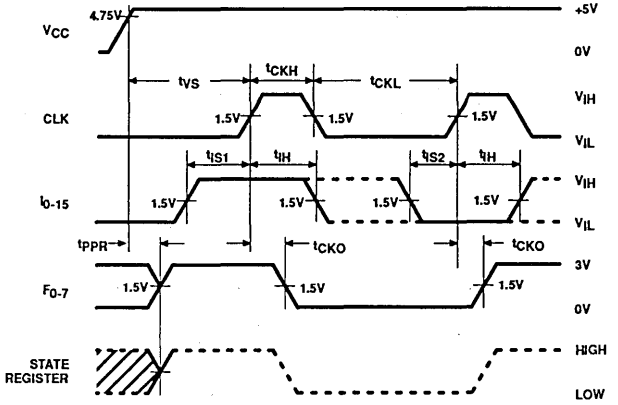
PLS Under Test

TERM	C	AND INPUT (Im) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 5	4	3	2	1	0
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

OPTION (P/E)	H
OR	
NEXT STATE (Ns) 5 4 3 2 1 0	OUTPUT (Fr) 7 6 5 4 3 2 1 0
L L L L L L	L L L L L L L L
H H H H H H	H H H H H H H H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any Signetics' qualified programming equipment.



Test Circuit Timing Diagram

TERM	C	AND INPUT (Im) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 5	4	3	2	1	0
48	–	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

OPTION (P/E)	H
OR	
NEXT STATE (Ns) 5 4 3 2 1 0	OUTPUT (Fr) 7 6 5 4 3 2 1 0
– – – – – –	– – – – – – – –
– – – – – –	– – – – – – – –

Test Array Deleted

Document No.	853–1351
ECN No.	99570
Date of Issue	May 9, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS105–45

Programmable logic sequencer ($16 \times 48 \times 8$)

DESCRIPTION

The PLUS105–45 is a bipolar programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 48 AND gates are connected to the 16 external dedicated inputs (I_0-I_{15}) and to the feedback paths of the 6 buried State Registers (Q_{P0}-Q_{P5}). Because the OR array is programmable, any one or all of the 48 transition terms can be connected to any or all of the State and Output Registers.

All state transition terms can include True, False and Don't Care states of the controlling state variables. A Complement Transition Array supports complex IF-THEN-ELSE state transitions with a single product term.

All buried State and Output registers are edge-triggered clocked S-R flip-flops. Asynchronous Preset/Output Enable functions are available.

The PLUS105–45 is pin-for-pin and software compatible with the Signetics PLS105 and PLS105A Logic Sequencers, as well as other commercially available 105-type programmable logic devices.

To facilitate testing of state machine designs, diagnostic mode features for register preset and buried state register observability have been incorporated into the PLUS105–45 device architecture.

Ordering codes are listed in the Ordering Information Table.

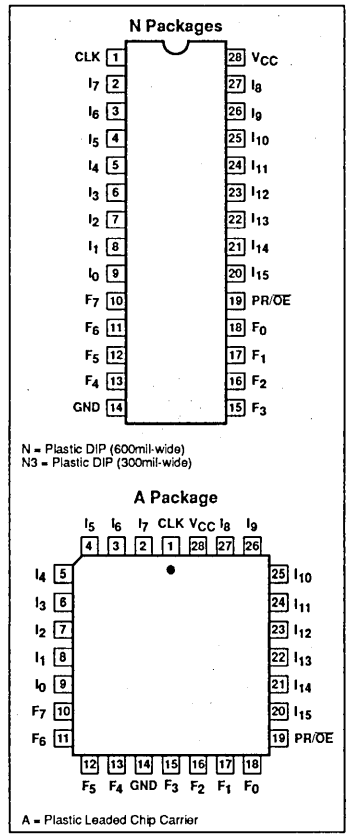
FEATURES

- 45MHz operating frequency
  - 55.6MHz clock rate
  - No OR term loading restrictions
- Available in 300mil skinny DIP, 600mil-wide Plastic DIP and PLCC packages
- Pin and software compatible with other commerically available 105 logic sequencers
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked S-R flip-flops
- Security fuse
- Programmable Asynchronous Preset or Output Enable
- Power-on preset (to all "1"s) of internal registers
- Power dissipation: 800mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security Locking systems
- Counters
- Shift registers

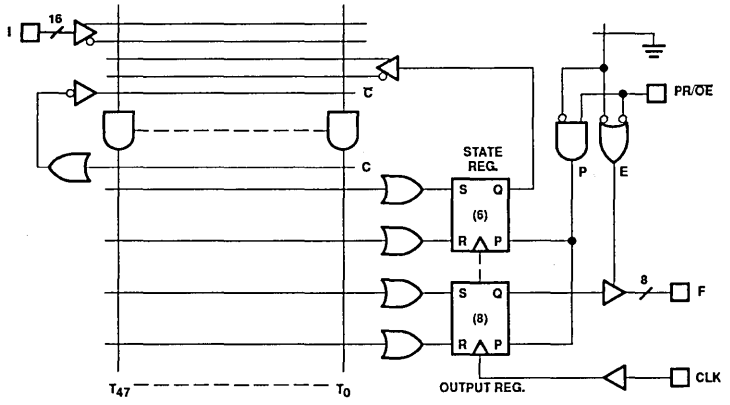
PIN CONFIGURATIONS



Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

FUNCTIONAL DIAGRAM



PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High (H)
2-9, 26, 27 20-22	I_0 - I_9, I_{13} - I_{15}	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/ Low (H/L)
23	I_{12}	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{12} is held at +10V, device outputs F_0 - F_5 reflect the contents of State Register bits P_0 - P_5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
24	I_{11}	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{11} is held at +10V, device outputs F_0 - F_5 become direct inputs for State Register bits P_0 - P_5; a Low-to-High transition on the clock line loads the values on pins F_0 - F_5 into the State Register bits P_0 - P_5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
25	I_{10}	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{10} is held at +10V, device outputs F_0 - F_7 become direct inputs for Output Register bits Q_0 - Q_7; a Low-to-High transition on the clock line loads the values on pins F_0 - F_7 into the Output Register bits Q_0 - Q_7. The contents of each State Register remains unaltered.	Active-High/ Low (H/L)
10-13 15-18	F_0 - F_7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register bits Q_0 - Q_7, when enabled. When I_{12} is held at +10V, F_0 - F_5 = (P_0 - P_5). When I_{11} is held at +10V, F_0 - F_5 become inputs to State Register bits P_0 - P_5. When I_{10} is held at +10V, F_0 - F_7 become inputs to Output Register bits Q_0 - Q_7.	Active-High (H)
19	PR/$\overline{OE}$	**Preset or Output Enable Input:** **A user programmable function:**	
		• **Preset:** Provides an asynchronous preset to logic "1" of all State and Output Register bits. PR overrides Clock, and when held High, clocking is inhibited and F_0 - F_7 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the Preset signal goes low. See timing definitions.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers F_0 - F_7 from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

TRUTH TABLE [1, 2, 3, 4, 5, 6, 7]

V_{CC}	OPTION										
	PR	$\overline{OE}$	I_{10}	I_{11}	I_{12}	CK	S	R	Q_P	Q_F	F
	H		*	*	*	X	X	X	H	H	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	IND.	IND.	IND.
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots T_{47}$
   $T_n = (C_0)\ (I_0, I_1, I_2, \ldots)\ (P_0, P_1, \ldots P_5)$
2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤ 5.5V)
6. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.
7. IND. = Indeterminent; R = S = H is an illegal input condition.

VIRGIN STATE

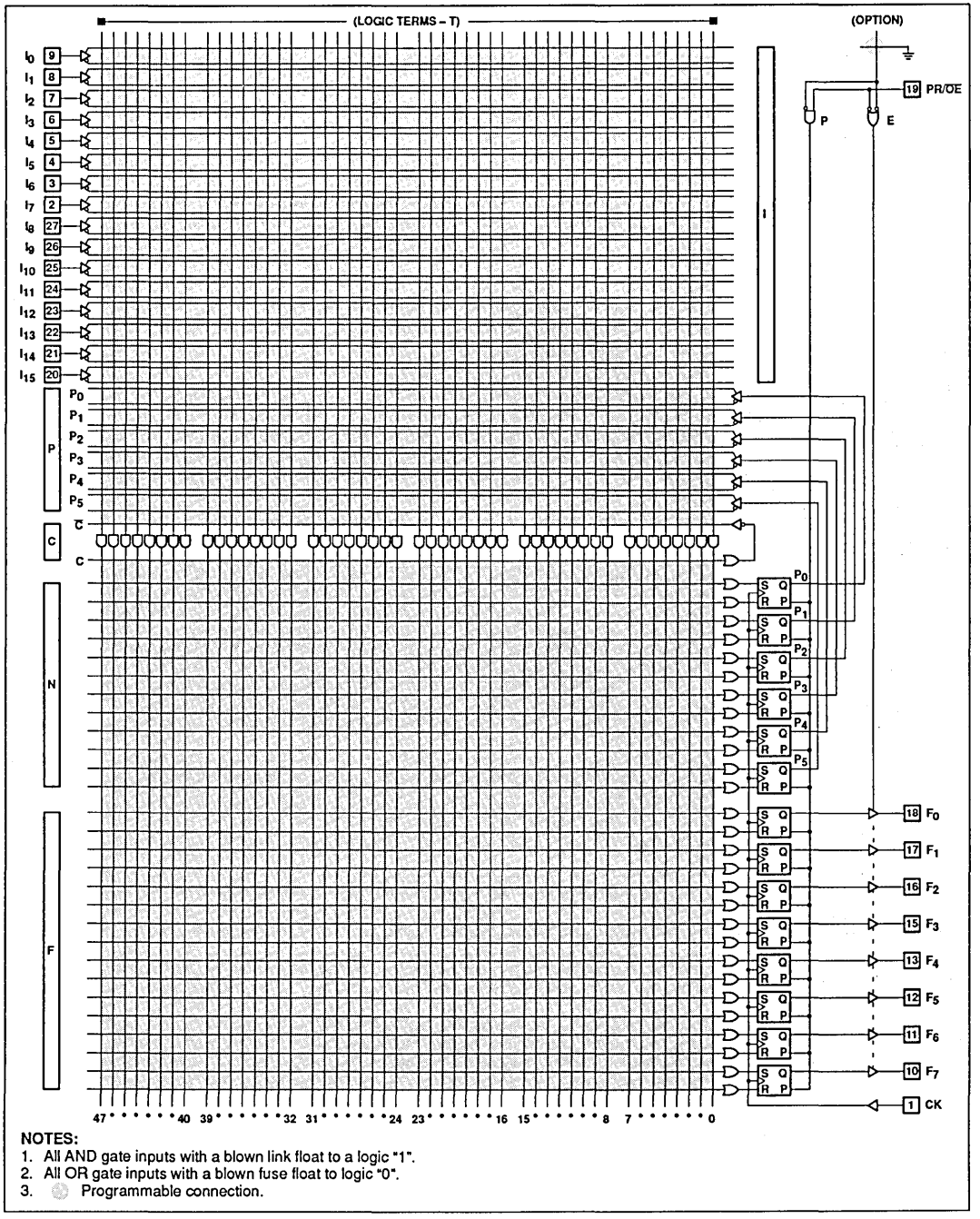
A factory-shipped virgin device contains all fusible links intact, such that:

1. PR/$\overline{OE}$ option is set to PR. Note that even if the PR function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array preprogrammed with a standard test pattern. NOTE: The Test Array pattern must be deleted before incorporating a user program.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

LOGIC DIAGRAM

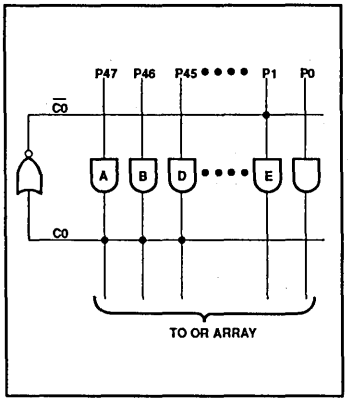


NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ◌ Programmable connection.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

COMPLEMENT ARRAY DETAIL



The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $\overline{(A + B + C)}$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

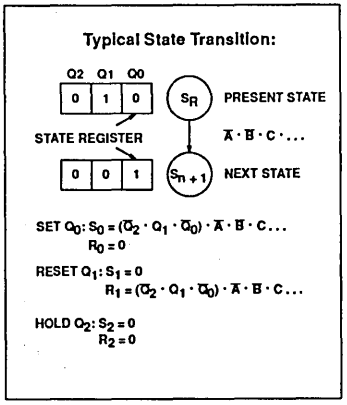
Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to a known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that use of the Complement Array adds an additional delay path through the device. Refer to the AC Electrical Characteristics for details.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
28-pin Plastic Dual-In-Line, 600mil-wide	PLUS105–45N
28-pin Plastic Dual-In-Line, 300mil-wide	PLUS105–45N3
28-pin Plastic Leaded Chip Carrier, 450mil-square	PLUS105–45A

LOGIC FUNCTION



THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7.0	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq 75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = –2mA	2.4			V
V_{OL}	Low	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = V_{CC}		<1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		–20	–250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = 2.7V		1	40	µA
		V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX			200	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short circuit should not exceed 1 second.
5. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30PF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAx	UNIT
Pulse Width							
t_{CKH}	Clock High	CK +	CK –	9	8		ns
t_{CKL}	Clock Low	CK –	CK +	9	8		ns
t_{CKP}	Clock Period	CK +	CK +	18	16		ns
t_{PRH}	Preset pulse	PR +	PR -	10	8		ns
Setup Time							
t_{IS1}	Input	Input ±	CK+	13	12		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK +	23	20		ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		ns
t_{PRS}	Clock resume (after preset)	PR –	CK –	0	–5		ns
t_{NVCK}	Clock lockout (before preset)	CK –	PR –	10	5		ns
Hold Time							
t_{IH}	Input	CK +	Input ±	0	–5		ns
Diagnostic Mode							
t_{RJS}	Initialization of diagnostic mode	I_{10} or I_{11} + (to 8V)	F_n as inputs	50	25		ns
t_{RJH}	Clock for diagnostic mode	CK +	Register input jam	50	25		ns
Propagation Delay[3]							
t_{CKO}	Clock	CK +	Output ±		8	9	ns
t_{OE}	Output enable[2]	OE –	Output –		8	9	ns
t_{OD}	Output disable[2]	OE +	Output +		8	9	ns
t_{PR}	Preset	PR +	Output +		12	15	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10	ns
Frequency of Operation							
f_{MAX1}	Without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO}}\right)$	Input ±	Output ±	45.0	50.0		MHz
f_{MAX2}	With Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO}}\right)$	Input thru Complement Array ±	Output ±	31.3	35.7		MHz
f_{MAX3}	Internal feedback without Complement Array $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	55.6	62.5		MHz
f_{MAX4}	Internal feedback with Complement Array $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Input ±	43.5	50.0		MHz
f_{CLK}	Clock frequency	CK +	CK +	55.6	62.5		MHz

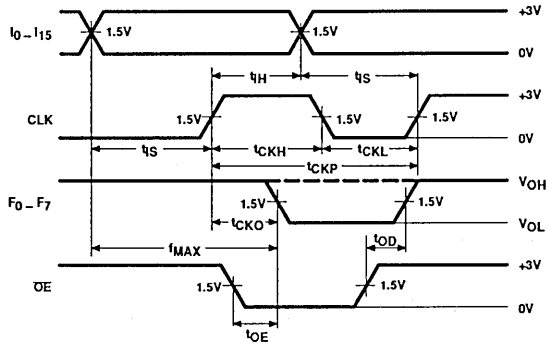
NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays and setup times are meausred and specified under worst case conditions.

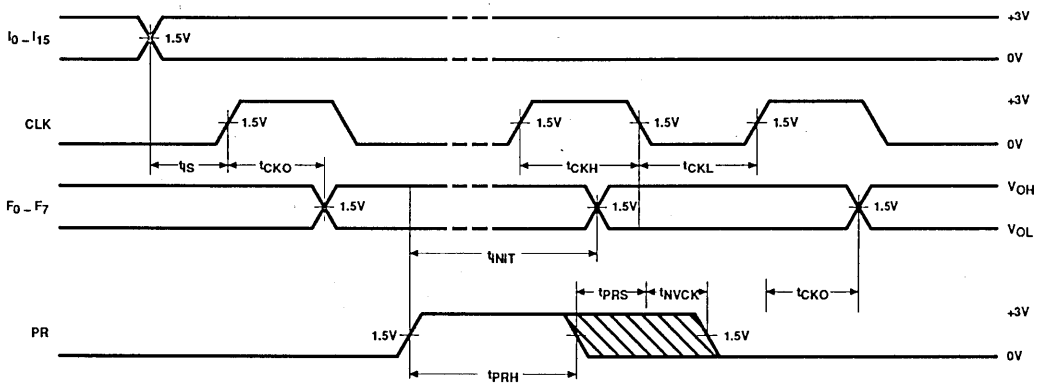
Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

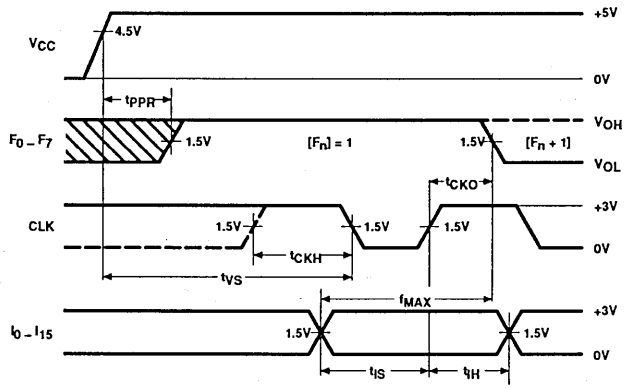
TIMING DIAGRAMS



Sequential Mode



Asynchronous Preset

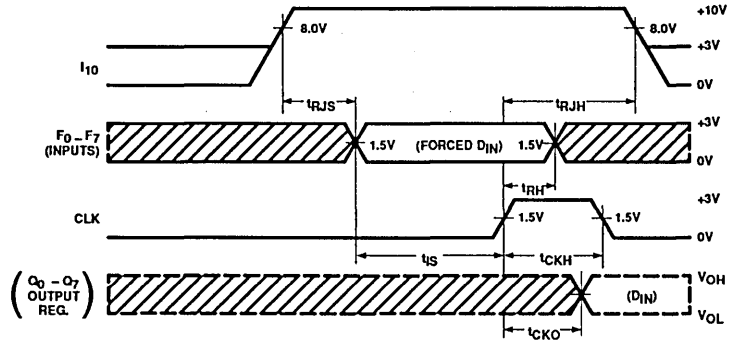


Power-On Preset

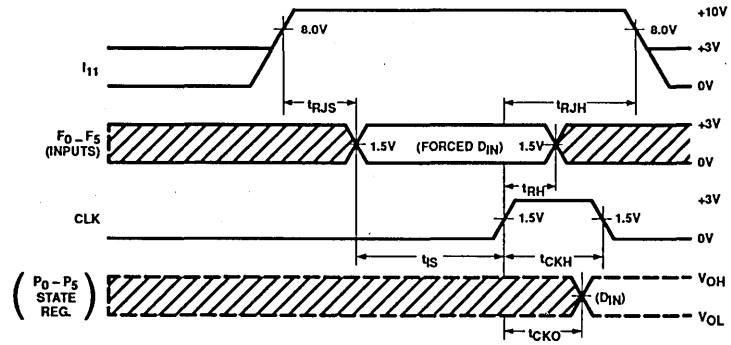
Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

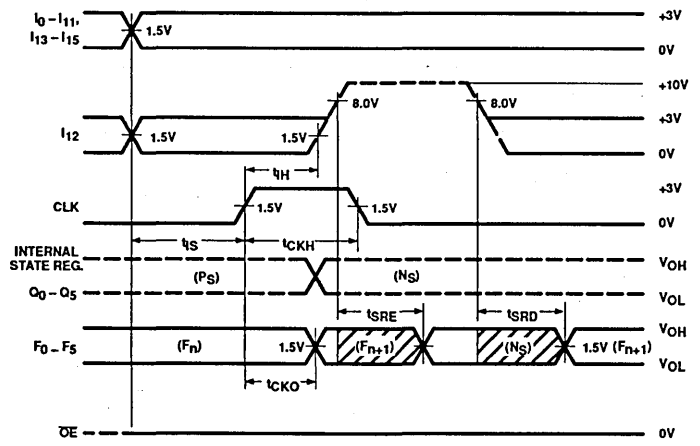
TIMING DIAGRAMS (Continued)



Diagnostic Mode—Output Register Input Jam



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—State Register Outputs

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

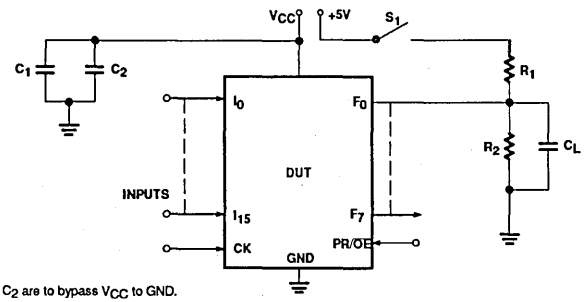
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{CKH}	Width of input clock pulse
t_{CKL}	Interval between clock pulses.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with PR/$\overline{OE}$ Low).
t_{CKP}	Minimum guaranteed clock period.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous PRESET to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF-state.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
t_{PR}	Delay between positive transition of Preset and when Outputs become valid at "1".
t_{PRH}	Width of preset input pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and the first positive transition of Clock.
t_{RH}	Required delay between positive transition of clock and the end of valid input data (F_0 – F_7 as inputs), when jamming data into the State or Output registers in the Diagnostic Mode.

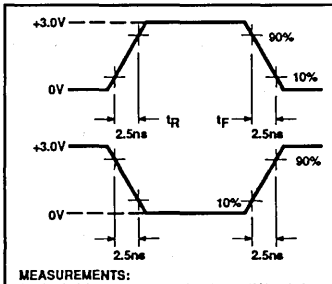
SYMBOL	PARAMETER
t_{RJH}	Required delay between positive transition of clock and return of input I_{10} or I_{11} from Diagnostic Mode (8V).
t_{RJS}	Required delay between inputs I_{10} or I_{11} transition to Diagnostic Mode (8V), and when the output pins become available as inputs.
t_{SRD}	Delay between input (I_{12}) transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I_{12} transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency)
f_{MAX}	Minimum guaranteed operating frequency.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

PLUS105-45 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' SLICE and SNAP design software packages. Signetics' AMAZE design software may also be used to compile PLUS105 designs, however, the timing simulator does not reflect the actual performance of the device. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLUS105–45 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS105-45 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

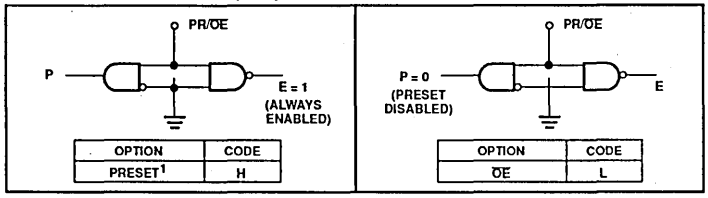
To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

PRESET/OE OPTION – (P/E)

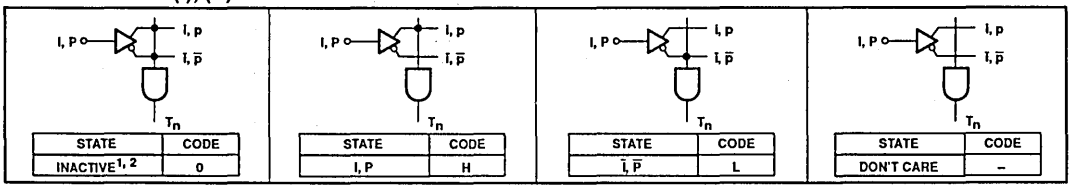


OPTION	CODE
PRESET[1]	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING THE PLUS105–45:
The PLUS105–45 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY – (I), (P)



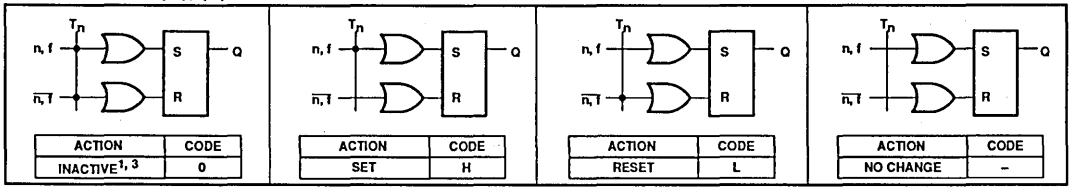
STATE	CODE
INACTIVE[1, 2]	0

STATE	CODE
I, P	H

STATE	CODE
$\overline{I}$, $\overline{P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



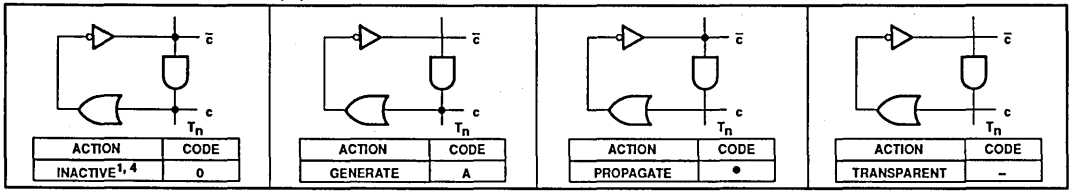
ACTION	CODE
INACTIVE[1, 3]	0

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	0

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates T_n.
2. Any gate T_n will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

PLUS105 PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
SIGNETICS DEVICE # CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND		C_n	OR		Ns, Fr
INACTIVE	0		INACTIVE	0	
GENERATE	A		SET	H	
PROPAGATE	•		RESET	L	
TRANSPARENT	—		NO CHANGE	—	

		I_m, P_s	OPTION		P/E
INACTIVE	0		PRESET	H	
I, P	H		$\overline{OE}$	L	
$\overline{I}$, $\overline{P}$	L				
DON'T CARE	—				

	AND																		OPTION (P/E) OR															
TERM	C_n	INPUT (I_m)																PRESENT STATE (P_s)						REMARKS	NEXT STATE (Ns)						OUTPUT (Fr)							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0		5	4	3	2	1	0	7	6	5	4	3	2	1	0
0																																					
1																																					
2																																					
3																																					
4																																					
5																																					
6																																					
7																																					
8																																					
9																																					
10																																					
11																																					
12																																					
13																																					
14																																					
15																																					
16																																					
17																																					
18																																					
19																																					
20																																					
21																																					
22																																					
23																																					
24																																					
25																																					
26																																					
27																																					
28																																					
29																																					
30																																					
31																																					
32																																					
33																																					
34																																					
35																																					
36																																					
37																																					
38																																					
39																																					
40																																					
41																																					
42																																					
43																																					
44																																					
45																																					
46																																					
47																																					
PIN NO.		20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9														10	11	12	13	15	16	17	18
VARIABLE NAME																																					

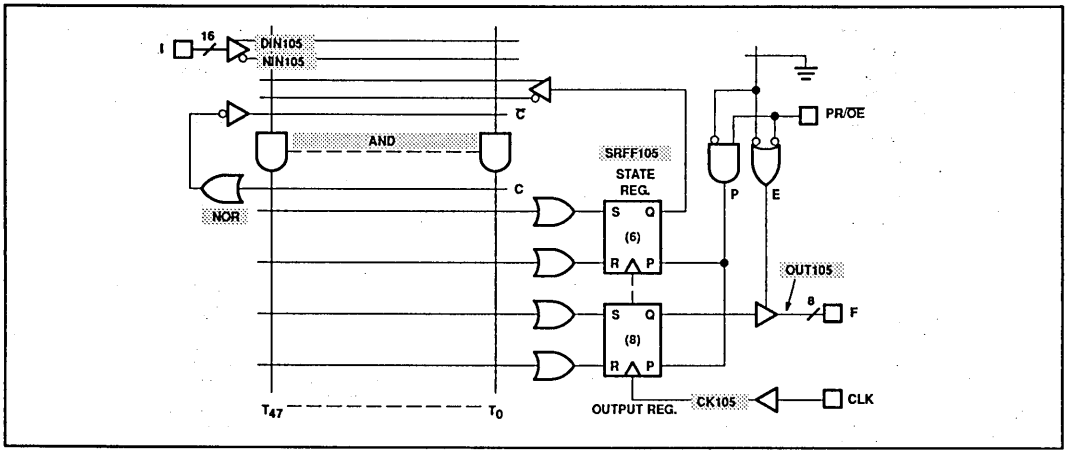
NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–45

SNAP RESOURCE SUMMARY DESIGNATIONS



Philips Components-Signetics

Document No.	853–1429
ECN No.	99569
Date of Issue	May 9, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS105–55

Programmable logic sequencer (16 × 48 × 8)

DESCRIPTION

The PLUS105–55 is a bipolar programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 48 AND gates are connected to the 16 external dedicated inputs (I0 - I15) and to the feedback paths of the 6 buried State Registers (Q_{P0}-Q_{P5}). Because the OR array is programmable, any one or all of the 48 transition terms can be connected to any or all of the State and Output Registers.

All state transition terms can include True, False and Don't Care states of the controlling state variables. A Complement Transition Array supports complex IF-THEN-ELSE state transitions with a single product term.

All buried State and Output registers are edge-triggered clocked S-R flip-flops. Asynchronous Preset/Output Enable functions are available.

The PLUS105–55 is pin-for-pin and software compatible with Signetics PLS105 and PLS105A Logic Sequencers, as well as other commercially available 105-type programmable logic devices.

To facilitate testing of state machine designs, diagnostic mode features for register preset and buried state register observability have been incorporated into the PLUS105–55 device architecture.

Ordering codes are listed in the Ordering Information Table.

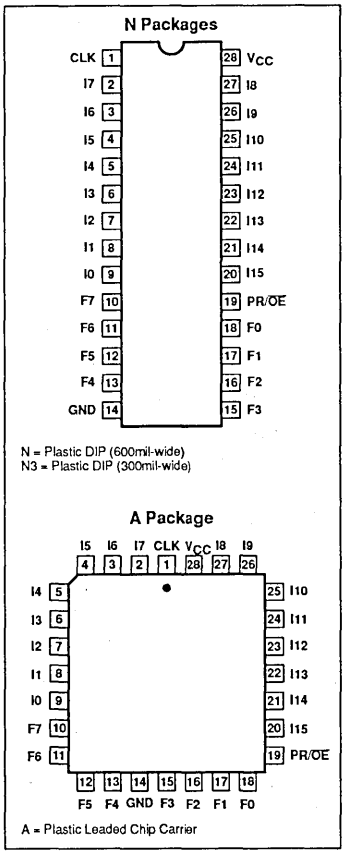
FEATURES

- 55MHz operating frequency
  - 71.4MHz clock rate
  - No OR term loading restrictions
- Available in 300mil skinny DIP, 600mil-wide DIP, and PLCC packages
- Pin and software compatible with other commerically available 105 sequencers
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked S-R flip-flops
- Security fuse
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to (all "1"s) of internal registers
- Power dissipation: 800mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security Locking systems
- Counters
- Shift registers

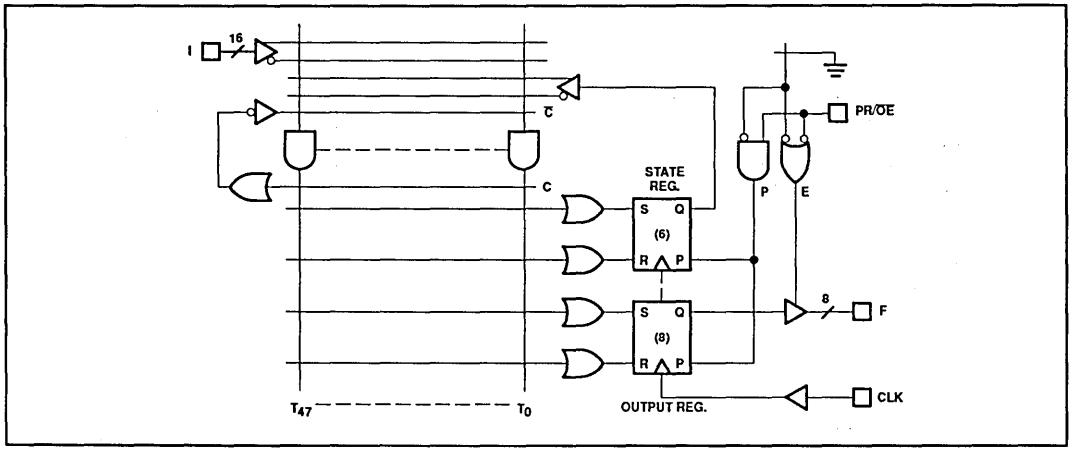
PIN CONFIGURATIONS



Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

FUNCTIONAL DIAGRAM



PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High (H)
2-9, 26, 27 20-22	I0 - I9, I13 - I15	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/ Low (H/L)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL levels. When I12 is held at +10V, device outputs F0 - F5 reflect the contents of State Register bits P0 - P5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I11 is held at +10V, device outputs F0 - F5 become direct inputs for State Register bits P0 - P5; a Low-to-High transition on the clock line loads the values on pins F0 - F5 into the State Register bits P0 - P5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I10 is held at +10V, device outputs F0 - F7 become direct inputs for Output Register bits Q0 - Q7; a Low-to-High transition on the clock line loads the values on pins F0 - F7 into the Output Register bits Q0 - Q7. The contents of each State Register remains unaltered.	Active-High/ Low (H/L)
10-13 15-18	F0 - F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register bits Q0 - Q7, when enabled. When I12 is held at +10V, F0 - F5 = (P0 - P5). When I11 is held at +10V, F0 - F5 become inputs to State Register bits P0 - P5. When I10 is held at +10V, F0 - F7 become inputs to Output Register bits Q0 - Q7.	Active-High (H)
19	PR/$\overline{OE}$	**Preset or Output Enable Input:** **A user programmable function:**	
		• **Preset:** Provides an asynchronous preset to logic "1" of all State and Output Register bits. PR overrides Clock, and when held High, clocking is inhibited and F0 - F7 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the Preset signal goes Low. See timing definitions.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers F0 - F7 from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

TRUTH TABLE [1, 2, 3, 4, 5, 6, 7]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I10	I11	I12	CK	S	R	Q_P	Q_F	F
	H		*	*	*	X	X	X	H	H	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	IND.	IND.	IND.
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots T_{48}$
   $T_n = (C_0)\ (I_0, I_1, I_2, \ldots)\ (P_0, P_1, \ldots P_5)$
2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤ 5.5V)
6. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.
7. IND. = Indeterminent; R = S = H is an illegal input condition.

VIRGIN STATE

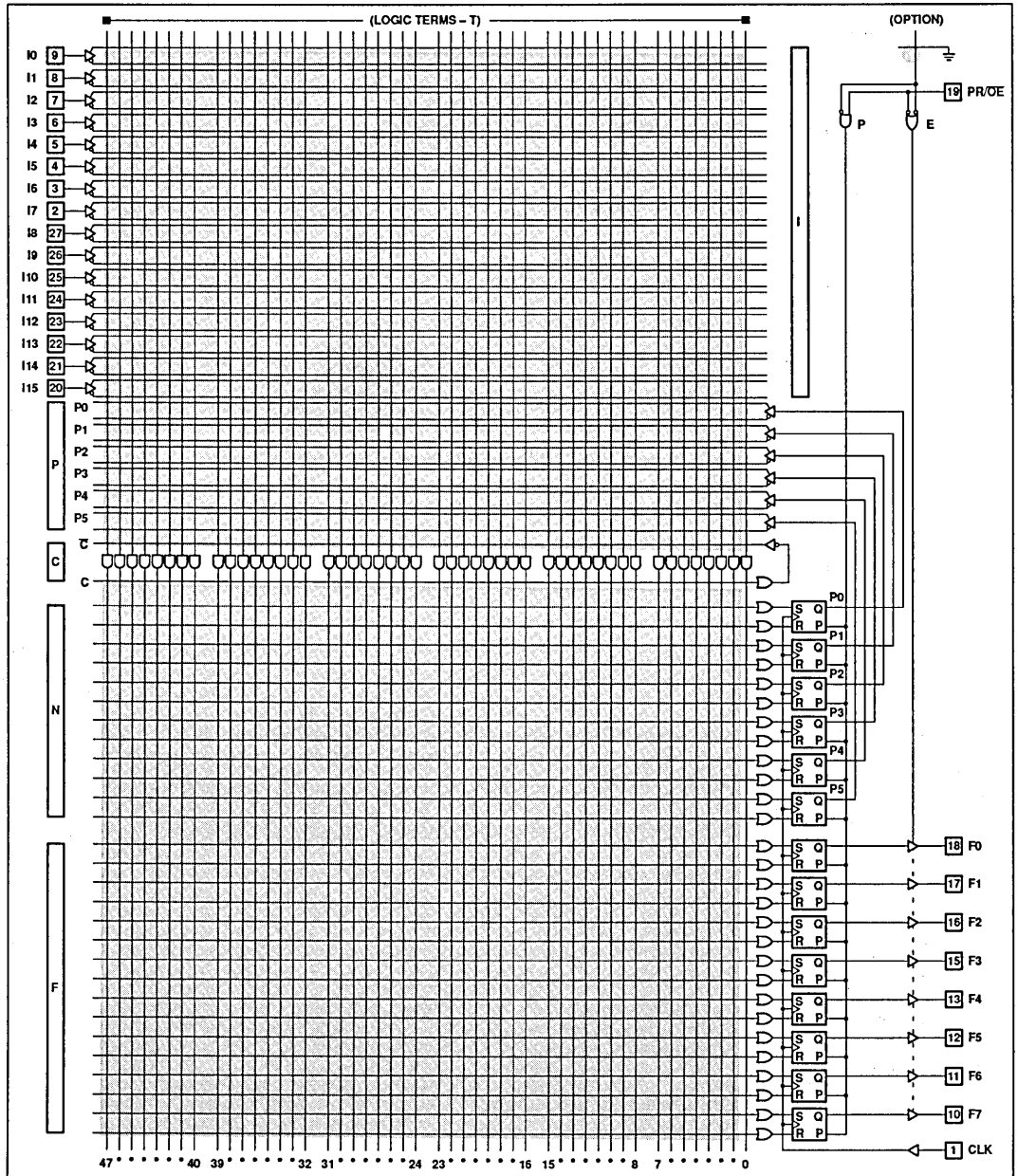
A factory-shipped virgin device contains all fusible links intact, such that:

1. PR/$\overline{OE}$ option is set to PR. Note that even if the PR function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).

Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

LOGIC DIAGRAM

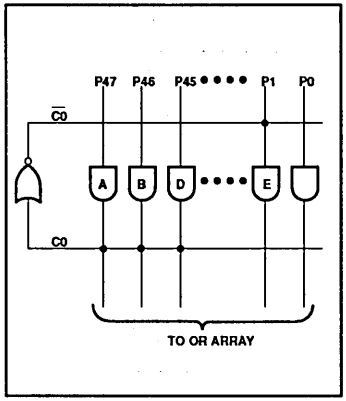


NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ● Programmable connection.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

COMPLEMENT ARRAY DETAIL



The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to a known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that use of the Complement Array adds an additional delay path through the device. Refer to the AC Electrical Characteristics for details.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
28-pin Plastic Dual-In-Line, 600mil-wide	PLUS105–55N
28-pin Plastic Dual-In-Line, 300mil-wide	PLUS105–55N3
28-pin Plastic Leaded Chip Carrier, 450mil-square	PLUS105–55A

THERMAL RATINGS

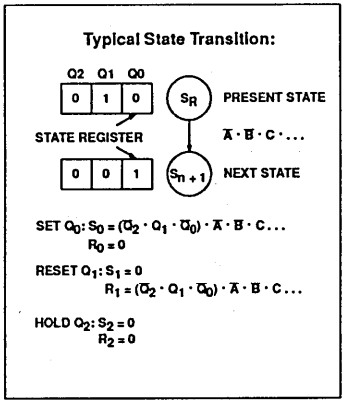
TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage		+7.0	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	-30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	-65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION



Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq 75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = V_{CC}		<1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		−20	−250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = 2.7V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		160	200	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short circuit should not exceed 1 second.
5. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

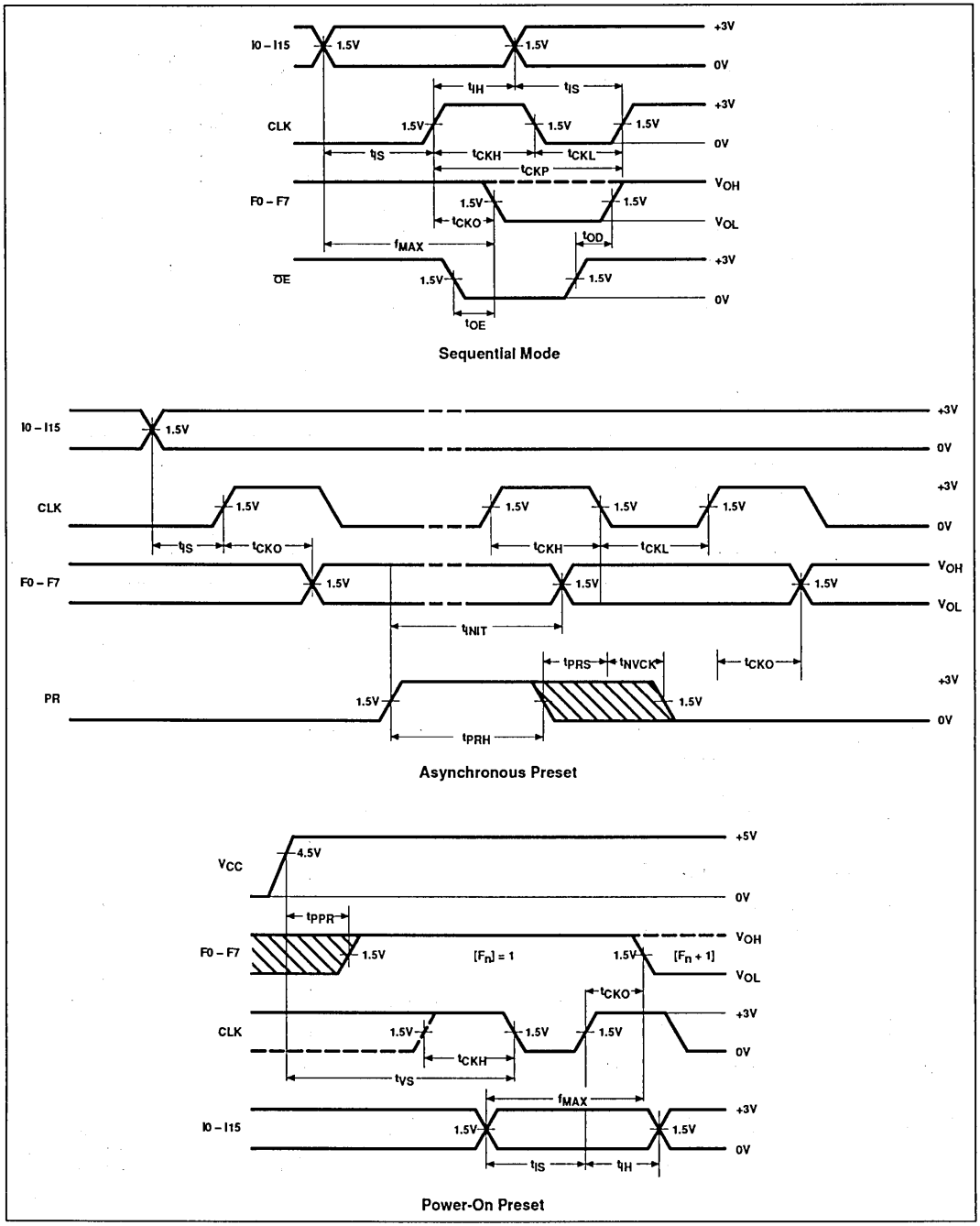
AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1K\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse Width							
t_{CKH}	Clock High	CK +	CK –	7	6.5		ns
t_{CKL}	Clock Low	CK –	CK +	7	6.5		ns
t_{CKP}	Clock Period	CK +	CK +	14	13		ns
t_{PRH}	Preset pulse	PR +	PR –	10	8.0		ns
Setup Time							
t_{IS1}	Input	Input ±	CK +	10	9.5		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK +	20	18.0		ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	0		ns
t_{PRS}	Clock resume (after preset)	PR –	CK –	0	0		ns
t_{NVCK}	Clock lockout (before preset)	CK –	PR –	12	10.0		ns
Hold Time							
t_{IH}	Input	CK +	Input ±	0	–5		ns
Diagnostic Mode							
t_{RJS}	Initialization of diagnostic mode	I10 or I11 + (to 8V)	F_n as inputs	50	25		ns
t_{RJH}	Clock for diagnostic mode	CK +	Register input jam	50	25		ns
Propagation Delay[2]							
t_{CKO}	Clock	CK +	Output ±		7	8	ns
t_{OE}	Output enable[3]	OE –	Output –		6	8	ns
t_{OD}	Output disable[3]	OE +	Output +		6	8	ns
t_{PR}	Preset	PR +	Output +		12	15	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		5	10	ns
Frequency of Operation							
f_{MAX1}	Without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO}}\right)$	Input ±	Output ±	55.0	60.6		MHz
f_{MAX2}	With Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO}}\right)$	Input thru Complement Array ±	Output ±	35.7	40.0		MHz
f_{MAX3}	Internal feedback without Complement Array $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	71.4	76.9		MHz
f_{MAX4}	Internal feedback with Complement Array $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Input ±	50.0	55.6		MHz
f_{CLK}	Clock period	CK +	CK +	71.4	76.9		MHz

NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. All propagation delays and setup times are meausred and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TIMING DIAGRAMS



Sequential Mode

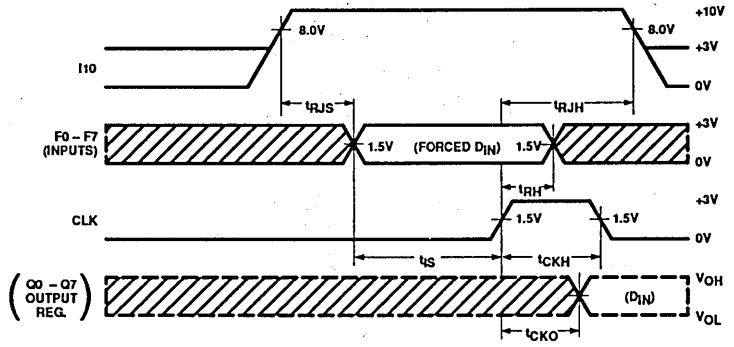
Asynchronous Preset

Power-On Preset

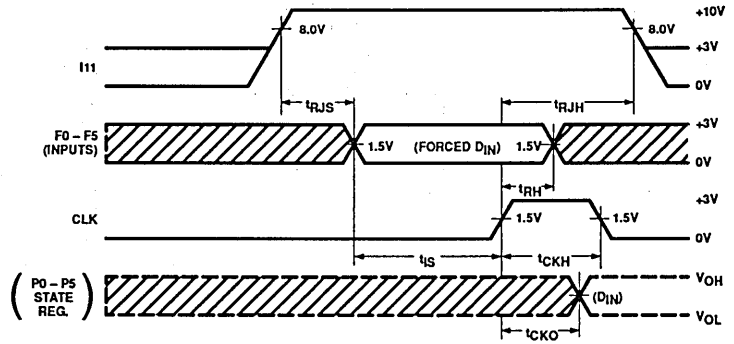
Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

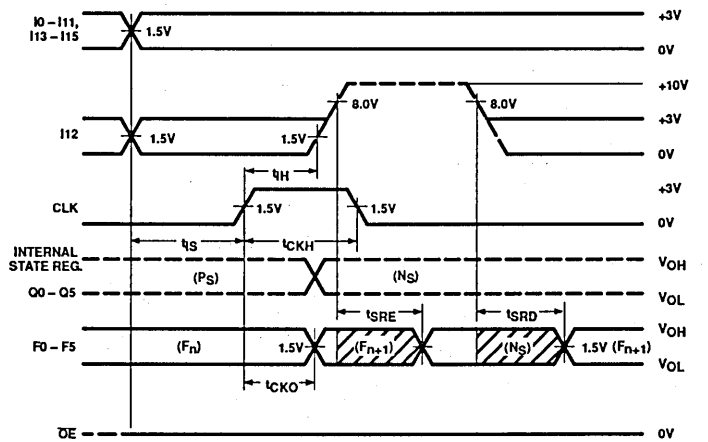
TIMING DIAGRAMS (Continued)



Diagnostic Mode—Output Register Input Jam



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—State Register Outputs

Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

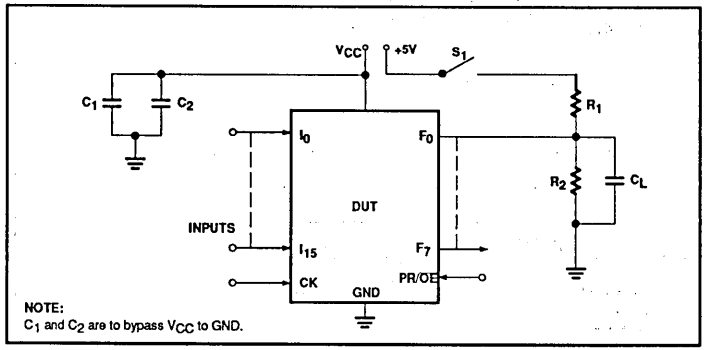
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{CKH}	Width of input clock pulse
t_{CKL}	Interval between clock pulses.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with PR/$\overline{OE}$ Low).
t_{CKP}	Minimum guaranteed clock period.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous PRESET to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF-state.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
t_{PR}	Delay between positive transition of Preset and when Outputs become valid at "1".
t_{PRH}	Width of preset input pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and the first positive transition of Clock.
t_{RH}	Required delay between positive transition of clock and the end of valid input data (F_0 – F_7 as inputs), when jamming data into the State or Output registers in the Diagnostic Mode.

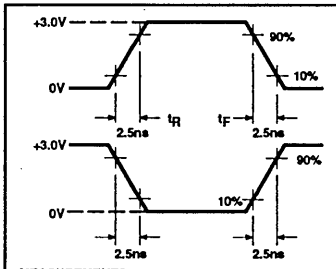
SYMBOL	PARAMETER
t_{RJH}	Required delay between positive transition of clock and return of input I_{10} or I_{11} from Diagnostic Mode (8V).
t_{RJS}	Required delay between inputs I_{10} or I_{11} transition to Diagnostic Mode (8V), and when the output pins become available as inputs.
t_{SRD}	Delay between input (I_{12}) transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I_{12} transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency)
f_{MAX}	Minimum guaranteed operating frequency.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

PLUS105-55 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics' SLICE and SNAP design software packages. Signetics' AMAZE design software may also be used to compile PLUS105 designs, however, the timing simulator does not reflect the actual performance of the device. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLUS105–55 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS105-55 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

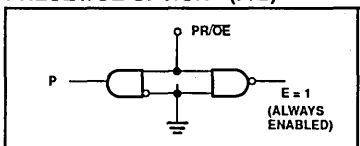
To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

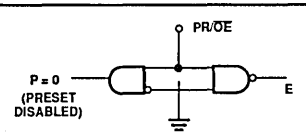
Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

PRESET/OE OPTION - (P/E)



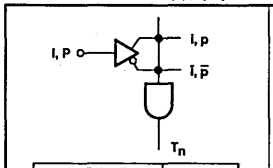
OPTION	CODE
PRESET[1]	H



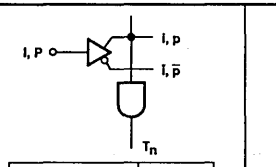
OPTION	CODE
$\overline{OE}$	L

PROGRAMMING THE PLUS105–55:
The PLUS105–55 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

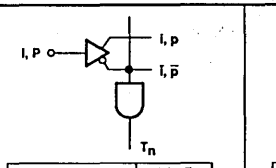
"AND" ARRAY - (I), (P)



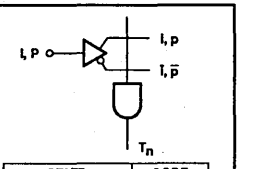
STATE	CODE
INACTIVE[1, 2]	0



STATE	CODE
I, P	H

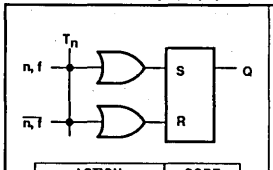


STATE	CODE
$\overline{I}$, $\overline{P}$	L

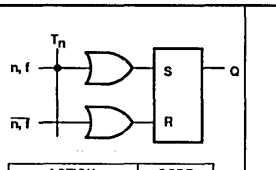


STATE	CODE
DON'T CARE	–

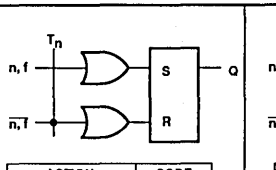
"OR" ARRAY - (N), (F)



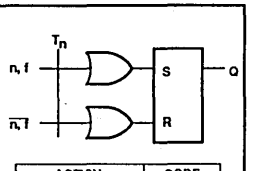
ACTION	CODE
INACTIVE[1, 3]	0



ACTION	CODE
SET	H

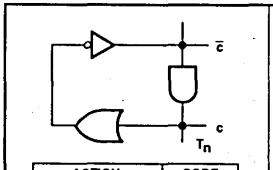


ACTION	CODE
RESET	L

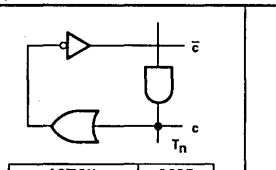


ACTION	CODE
NO CHANGE	–

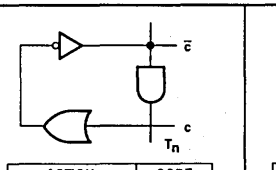
"COMPLEMENT" ARRAY - (C)



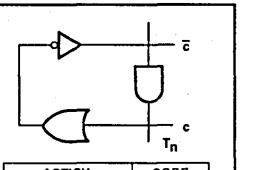
ACTION	CODE
INACTIVE[1, 4]	0



ACTION	CODE
GENERATE	A



ACTION	CODE
PROPAGATE	•



ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates T_n.
2. Any gate T_n will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencer (16 × 48 × 8)

PLUS105–55

PLUS105 PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
SIGNETICS DEVICE # CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND

C_n	
INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

I_m, P_s	
INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

OR

N_s, Fr	
INACTIVE	0
SET	H
RESET	L
NO CHANGE	—

OPTION

P/E	
PRESET	H
$\overline{OE}$	L

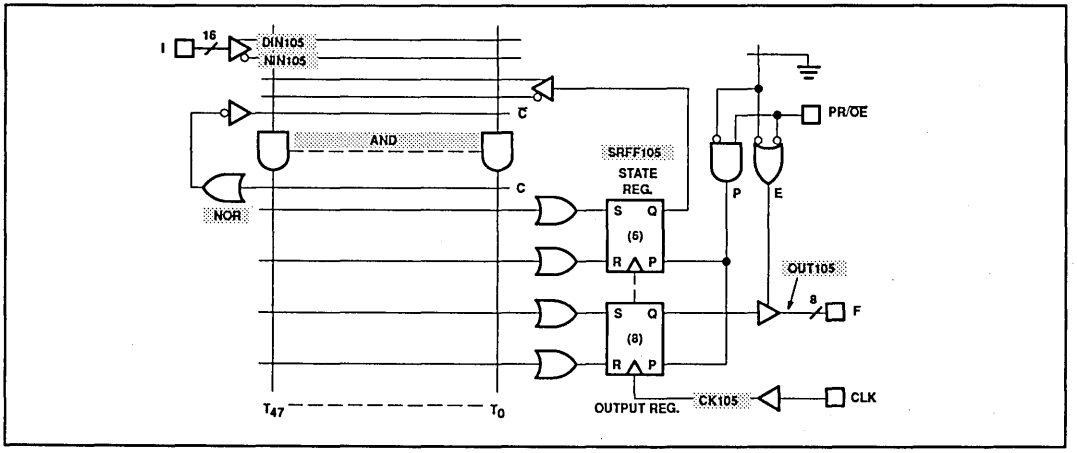
TERM	C_n	AND: INPUT (I_m) 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (N_s) 5 4 3 2 1 0	OPTION (P/E) OR: OUTPUT (Fr) 7 6 5 4 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		20 21 22 23 24 25 26 27 2 3 4 5 6 7 8 9				10 11 12 13 15 16 17 18
VARIABLE NAME						

NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencer (16 × 48 × 8) PLUS105–55

SNAP RESOURCE SUMMARY DESIGNATIONS



Philips Components-Signetics

Document No.	853–1280
ECN No.	00748
Date of Issue	October 18, 1990
Status	Product Specification
Programmable Logic Devices	

PLUS405–37/–45

Programmable logic sequencers ($16 \times 64 \times 8$)

DESCRIPTION

The PLUS405 devices are bipolar, programmable state machines of the Mealy type. Both the AND and the OR array are user-programmable. All 64 AND gates are connected to the 16 external dedicated inputs (I_0 - I_{15}) and to the feedback paths of the 8 on-chip State Registers (Q_{P0} - Q_{P7}). Two complement arrays support complex IF-THEN-ELSE state transitions with a single product term (input variables C_0, C_1).

All state transition terms can include True, False and Don't Care states of the controlling state variables. All AND gates are merged into the programmable OR array to issue the next-state and next-output commands to their respective registers. Because the OR array is programmable, any one or all of the 64 transition terms can be connected to any or all of the State and Output Registers.

All state (Q_{P0} - Q_{P7}) and output (Q_{F0} - Q_{F7}) registers are edge-triggered, clocked J-K flip-flops, with Asynchronous Preset and Reset options. The PLUS405 architecture provides the added flexibility of the J-K toggle function which is indeterminate on S-R flip-flops. Each register may be individually programmed such that a specific Preset-Reset pattern is initialized when the initialization pin is raised to a logic level "1". This feature allows the state machine to be asynchronously initialized to known internal state and output conditions prior to proceeding through a sequence of state transitions. Upon power-up, all registers are unconditionally preset to "1". If desired, the initialization input pin (INIT) can be converted to an $\overline{\text{Output Enable}}$ ($\overline{\text{OE}}$) function as an additional user-programmable feature.

Availability of two user-programmable clocks allows the user to design two independently clocked state machine functions consisting of four state and four output bits each.

Order codes are listed in the Ordering Information Table.

FEATURES

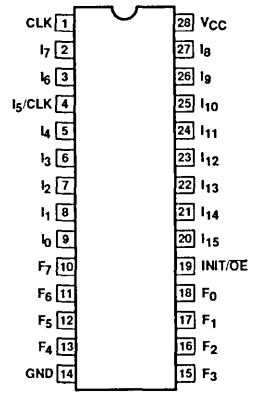
- PLUS405–37
  - f_{MAX} = 37MHz
  - 50MHz clock rate
- PLUS405–45
  - f_{MAX} = 45MHz
  - 58.8MHz clock rate
- Functional superset of PLS105/105A
- Field-programmable (Ti-W fusible link)
- 16 input variables
- 8 output functions
- 64 transition terms
- 8-bit State Register
- 8-bit Output Register
- 2 transition Complement Arrays
- Multiple clocks*
- Programmable Asynchronous Initialization or $\overline{\text{Output Enable}}$
- Power-on preset of all registers to "1"
- "On-chip" diagnostic test mode features for access to state and output registers
- 950mW power dissipation (typ.)
- TTL compatible
- J-K or S-R flip-flop functions
- Automatic "Hold" states
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator contollers
- Security locking systems
- Counters
- Shift registers

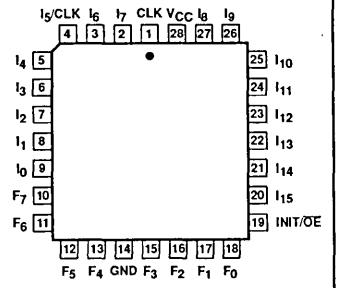
PIN CONFIGURATIONS

N Package



N = Plastic DIP (600mil-wide)

A Package



A = Plastic Leaded Chip Carrier

*Refer to AC Specifications for clock and operating frequencies when using multiple clocks.

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK1	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers. Pin 1 only clocks P_{0-3} and F_{0-3} if Pin 4 is also being used as a clock.	Active-High (H)
2, 3, 5–9, 26–27 20–22	$I_0 - I_4$, I_7, I_8 $I_8 - I_9$ $I_{13} - I_{15}$	**Logic Inputs:** The 12 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/Low (H/L)
4	CLK2	**Logic Input/Clock:** A user programmable function:	
		• **Logic Input:** A 13th external logic input to the AND array, as above.	Active-High/Low (H/L)
		• **Clock:** A 2nd clock for the State Registers P_{4-7} and Output Registers F_{4-7}, as above. Note that input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using Pin 4 as a Clock.	Active-High (H)
23	I_{12}	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I_{12} is held at +10V, device outputs $F_0 - F_7$ reflect the contents of State Register bits $P_0 - P_7$. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
24	I_{11}	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{11} is held at +10V, device outputs $F_0 - F_7$ become direct inputs for State Register bits $P_0 - P_7$; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0 - F_7$ into the State Register bits $P_0 - P_7$. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
25	I_{10}	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{10} is held at +10V, device outputs $F_0 - F_7$ become direct inputs for Output Register bits $Q_0 - Q_7$; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0 - F_7$ into the Output Register bits $Q_0 - Q_7$. The contents of each State Register remains unaltered.	Active-High/Low (H/L)
10–13 15–18	$F_0 - F_7$	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register Bits $Q_0 - Q_7$, when enabled. When I_{12} is held at +10V, $F_0 - F_7 = (P_0 - P_7)$. When I_{11} is held at +10V, $F_0 - F_7$ become inputs to State Register bits $P_0 - P_7$. When I_{10} is held at +10V, $F_0 - F_7$ become inputs to Output Register bits $Q_0 - Q_7$.	Active-High (H)
19	INIT/$\overline{OE}$	**Initialization or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Initialization:** Provides an asynchronous preset to logic "1" or reset to logic "0" of all State and Output Register bits, determined individually for each register bit through user programming. INIT overrides Clock, and when held High, clocking is inhibited and $F_0 - F_7$ and $P_0 - P_7$ are in their initialization state. Normal ~~clocking~~ resumes with the first ~~diff clock~~ pulse following a High-to-Low clock transition, after INIT goes Low. See timing definition for t_{NVCK} and t_{VCK}.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an output enable function to buffers $F_0 - F_7$ from the Output Registers.	Active-Low (L)

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

TRUTH TABLE [1, 2, 3, 4, 5, 6, 7]

V_{CC}	OPTION INIT	OPTION $\overline{OE}$	I_{10}	I_{11}	I_{12}	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H/L	H/L	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

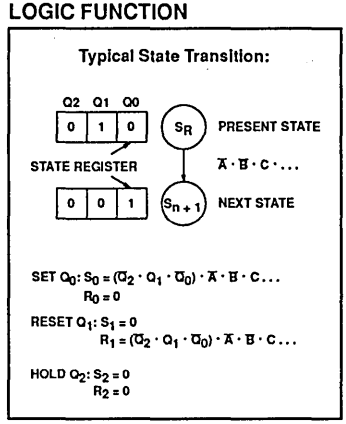
1. Positive Logic:
   S/R (or J/K) = $T_0 + T_1 + T_2 + \ldots T_{63}$
   $T_n = (C_0, C_1)(I_0, I_1, I_2, \ldots)(P_0, P_1, \ldots P_7)$
2. Either Initialization (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤5.5V)
6. H/L implies that either a High or a Low can occur, depending upon user-programmed selection (each State and Output Register individually programmable).
7. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

A factory-shipped virgin device contains all fusible links intact, such that:

1. INIT/$\overline{OE}$ is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user-programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are inactive (0).
3. All S/R (or J/K) flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array preprogrammed with a standard test pattern.
5. Clock 2 is inactive.

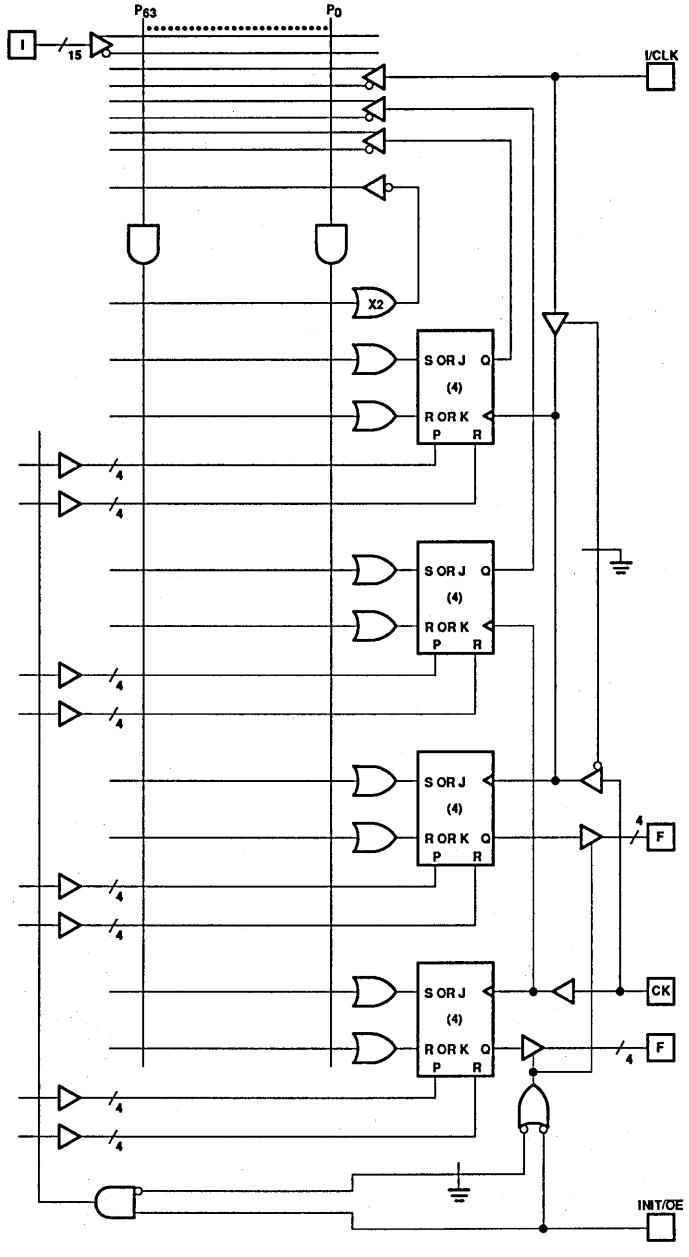
LOGIC FUNCTION



Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

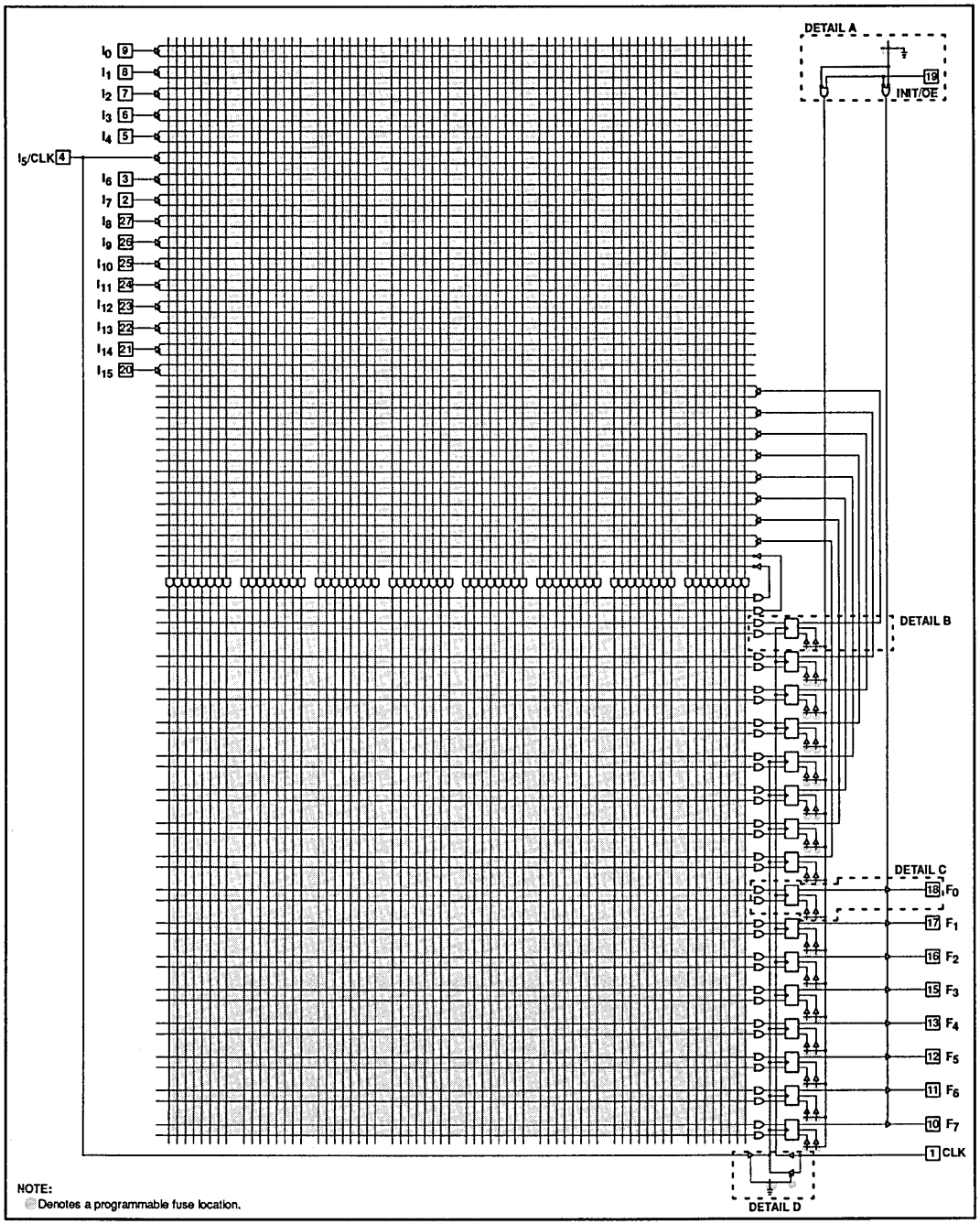
FUNCTIONAL DIAGRAM



Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

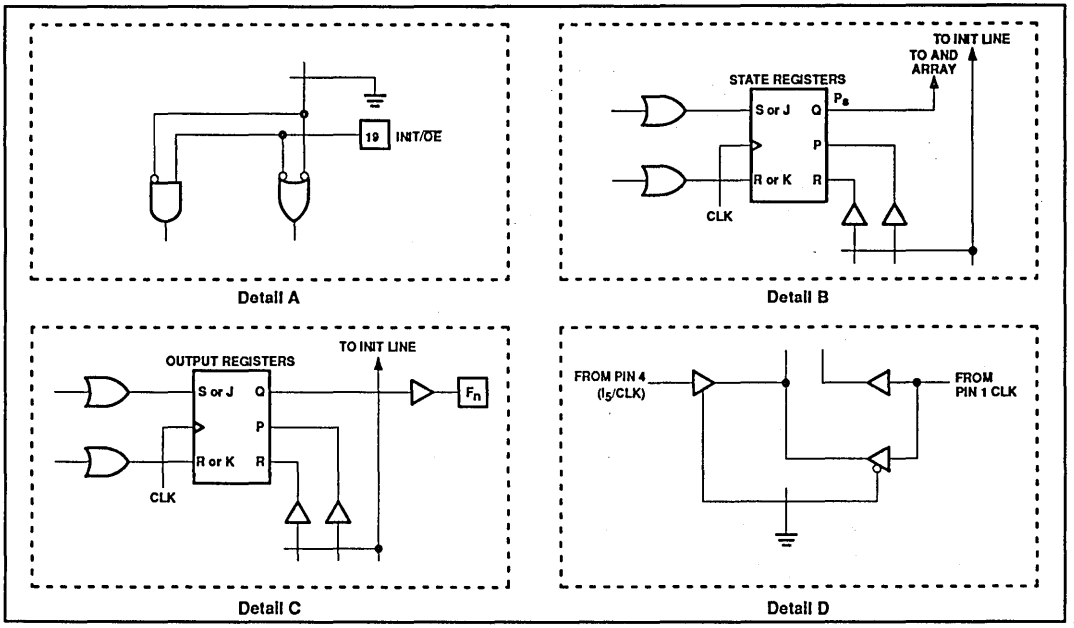
LOGIC DIAGRAM



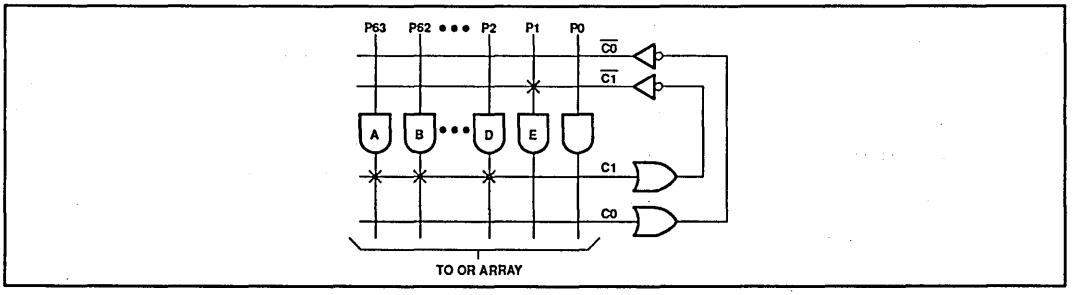
Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

DETAILS FOR REGISTERS FOR PLUS405



COMPLEMENT ARRAY DETAIL



The Complement Array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The Complement Array is a single OR gate with inputs from the AND array. The output of the Complement Array is inverted and fed back to the AND array (NOR). The output of the array will be Low if any one or more of the AND terms connected to it are active (High). If, however, all the connected terms are inactive (Low), which is a classic unknown state, the output of the Complement Array will be High.

Consider the Product Terms A, B and D that represent defined states. They are also connected to the input of the Complement Array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to Product Term E, which could be used in turn to reset the state machine to a known state. Without the Complement Array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that the PLUS405 has 2 Complement Arrays which allow the user to design 2 independent Complement functions. This is particularly useful if 2 independent state machines have been implemented on one device.

Note that use of the Complement Array adds an additional delay path through the device. Please refer to the AC Electrical Characteristics for details.

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

ORDERING INFORMATION

DESCRIPTION	OPERATING FREQUENCY	ORDER CODE
28-Pin Plastic DIP (600mil-wide)	45MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–45N
28-Pin Plastic DIP (600mil-wide)	37MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–37N
28-Pin Plastic Leaded Chip Carrier	45MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–45A
28-Pin Plastic Leaded Chip Carrier	37MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–37A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{Sstg}	Storage temperature range	–65 to +150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = V_{CC}		<1	25	µA
I_{IL}	Low	V_{CC} = MAX, V_{IN} = 0.45V		–20	–250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{CC} = MAX, V_{OUT} = 2.7V		1	40	µA
		V_{CC} = MAX, V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		190	225	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{CC} = 5.0V, V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short-circuit should not exceed one second.
5. I_{CC} is measured with the INIT/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLUS405–37			PLUS405–45			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width										
t_{CKH1}	Clock High; CLK1 (Pin 1)	CK+	CK–	10	8		8.5	7		ns
t_{CKL1}	Clock Low; CLK1 (Pin 1)	CK–	CK+	10	8		8.5	7		ns
t_{CKP1}	CLK1 Period	CK+	CK+	20	16		17	14		ns
t_{CKH2}	Clock High; CLK2 (Pin 4)	CK+	CK–	10	8		10	8		ns
t_{CKL2}	Clock Low; CLK2 (Pin 4)	CK–	CK+	10	8		10	8		ns
t_{CKP2}	CLK2 Period	CK+	CK+	20	16		20	16		ns
t_{INITH}	Initialization pulse	INIT–	INIT+	15	10		15	8		ns
Setup time										
t_{IS1}	Input	Input ±	CK+	15	12		12	10		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK+	25	20		22	18		ns
t_{VS}	Power-on preset	V_{CC}+	CK–	0	–10		0	–10		ns
t_{VCK}	Clock resume (after Initialization)	INIT–	CK–	0	–5		0	–5		ns
t_{NVCK}	Clock lockout (before Initialization)	CK–	INIT–	15	5		15	5		ns
Hold time										
t_{IH}	Input	CK+	Input ±	0	–5		0	–5		ns
Propagation delay										
t_{CKO1}	Clock1 (Pin 1)	CK1+	Output ±		10	12		8	10	ns
t_{CKO2}	Clock2 (Pin 4)	CK2+	Output ±		12	15		10	12	ns
t_{OE}[2]	Output Enable	OE–	Output –		12	15		12	15	ns
t_{OD}[2]	Output Disable	OE+	Output +		12	15		12	15	ns
t_{INIT}	Initialization	INIT+	Output +		15	20		15	20	ns
t_{PPR}	Power-on Preset	V_{CC} +	Output +		0	10		0	10	ns

Notes on following page

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

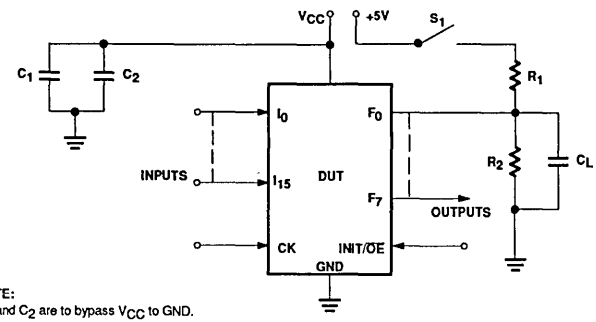
AC ELECTRICAL CHARACTERISTICS (Continued)

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLUS405–37			PLUS405–45			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Frequency of operation										
f_{MAX1}	CLK1; without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO1}}\right)$	Input ±	Output ±	37.0	45.5		45.5	55.6		MHz
f_{MAX2}	CLK2; without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO2}}\right)$	Input ±	Output ±	33.0	41.7		41.7	50.0		MHz
f_{MAX3}	CLK1; with Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO1}}\right)$	Input thru Complement Array ±	Output ±	27.0	33.3		31.3	38.5		MHz
f_{MAX4}	CLK2; with Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO2}}\right)$	Input thru Complement Array ±	Output ±	25.0	31.3		29.4	35.7		MHz
f_{MAX5}	Internal feedback without Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Intput ±	50.0	62.5		58.8	72.4		MHz
f_{MAX6}	Internal feedback with Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Intput ±	40.0	50.0		45.5	55.6		MHz
f_{CLK}	Minimum guaranteed clock frequency	CK +	CK +	50.0	62.5		58.8	72.4		MHZ

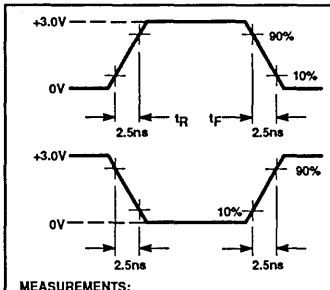
NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays and setup times re measured and specified under worst case conditions.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

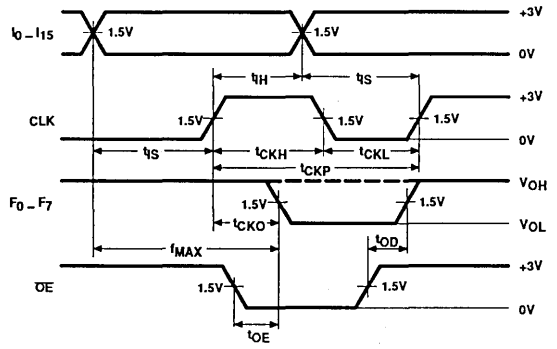
VOLTAGE WAVEFORMS



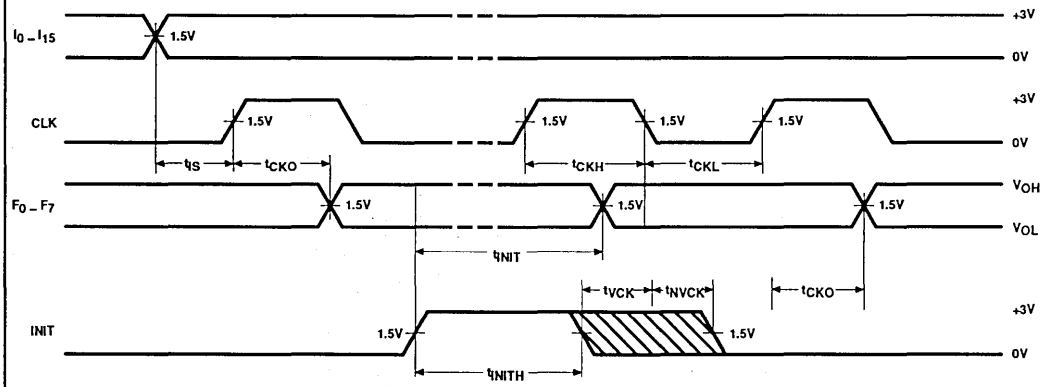
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

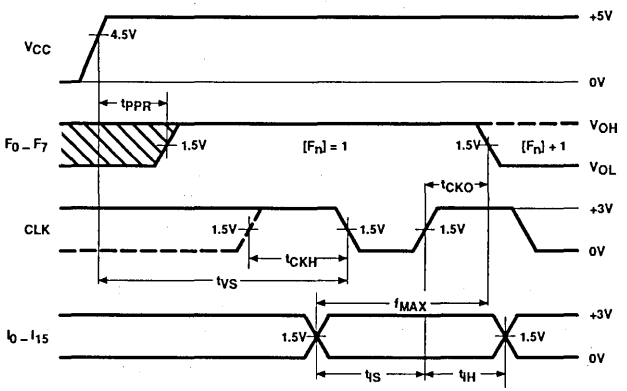
TIMING DIAGRAMS



Sequential Mode

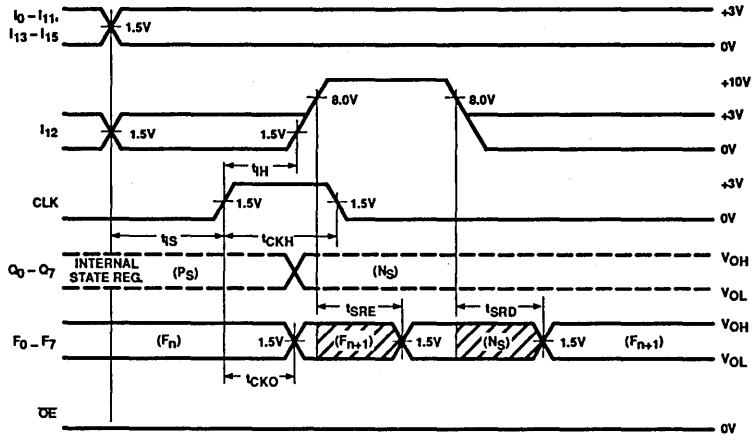


Asynchronous Initialization

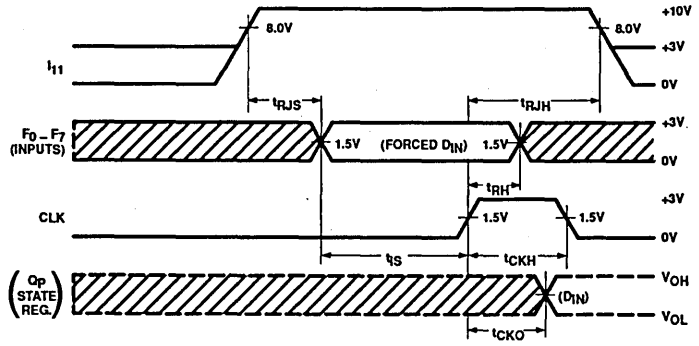


Power-On Preset

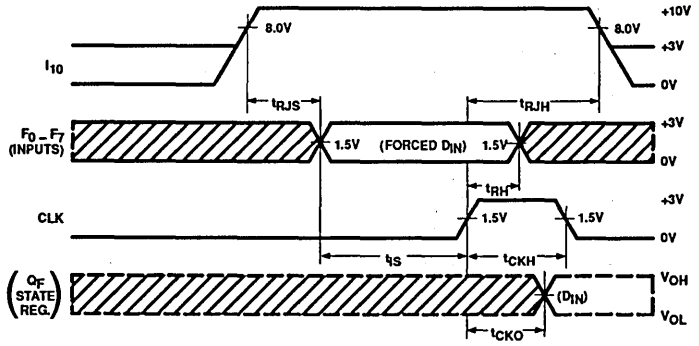
TIMING DIAGRAMS (Continued)



Diagnostic Mode – State Register Outputs



Diagnostic Mode – State Register Input Jam



Diagnostic Mode – Output Register Input Jam

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

TIMING DEFINITIONS

SYMBOL	PARAMETER
$t_{CKH1,2}$	Width of input clock pulse.
$t_{CKP1,2}$	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
$t_{CKO1,2}$	Delay between positive transition of Clock and when Outputs become valid (with INIT/$\overline{OE}$ Low).
t_{PPR}	Delay between V_{CC} (after power–on) and when Outputs become preset at "1".
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{RJH}	Required delay between positive transition of clock, and return of input I_{10} or I_{11} from Diagnostic Mode (8V).
$f_{MAX1,2}$	Minimum guaranteed operating frequency; input to output (CLK1 and CLK2).
$f_{MAX3,4}$	Minimum guaranteed operating frequency; input through Complement Array, to output (CLK1 and CLK2).
f_{MAX5}	Minimum guaranteed internal operating frequency; with internal feedback from state register to state register.

SYMBOL	PARAMETER
f_{MAX6}	Minimum guaranteed internal operating frequency with Complement Array; with internal feedback from state register through Complement Array, to state register.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency).
$t_{CKL1,2}$	Interval between clock pulses.
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{SRE}	Delay between input I_{12} transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{RJS}	Required delay between inputs I_{11} or I_{10} transition to Diagnostic Mode (8V), and when the output pins become available as inputs.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{INITH}	Width of initialization input pulse.
t_{VS}	Required delay between V_{CC} (after power–on) and negative transition of Clock preceding first reliable clock pulse.
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF–state.
t_{INIT}	Delay between positive transition of Initialization and when Outputs become valid.
t_{SRD}	Delay between input I_{12} transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{RH}	Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
t_{VCK}	Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding first reliable clock pulse.

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

LOGIC PROGRAMMING

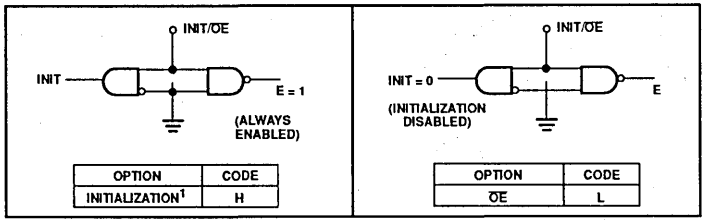
The PLUS405–37/–45 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS405–37/–45 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS405–37/–45 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, each logic variable (I, B, P, S, T, etc.) from the logic equations is assigned a symbol. TRUE, COMPLEMENT, PRESET, RESET, OUTPUT ENABLE, INACTIVE, etc., symbols are defined below.

INITIALIZATION/$\overline{OE}$ OPTION – (INIT/$\overline{OE}$)



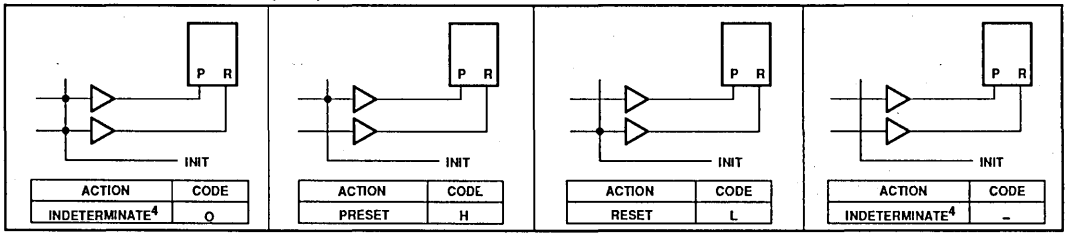
OPTION	CODE
INITIALIZATION[1]	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING THE PLUS405:

The PLUS405 has a power–up preset feature. This feature insures that the device will power–up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

INITIALIZATION OPTION – (INIT)



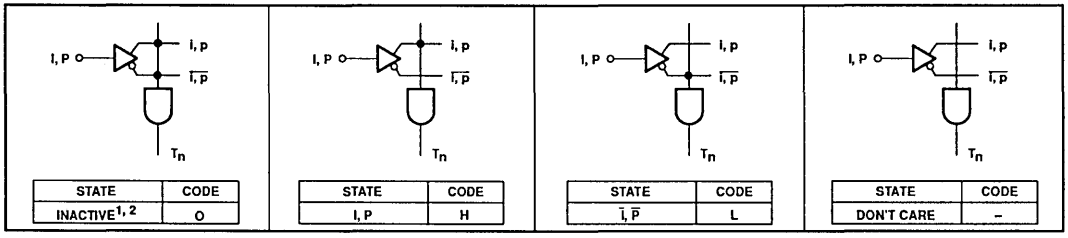
ACTION	CODE
INDETERMINATE[4]	O

ACTION	CODE
PRESET	H

ACTION	CODE
RESET	L

ACTION	CODE
INDETERMINATE[4]	–

"AND" ARRAY – (I), (P)



STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I, P}$	L

STATE	CODE
DON'T CARE	–

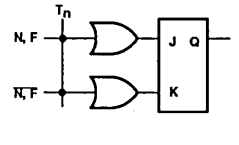
Notes are on next page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

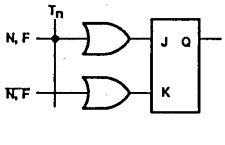
Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

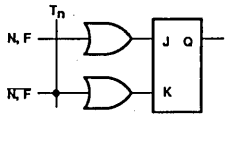
"OR" ARRAY – J-K FUNCTION – (N), (F)



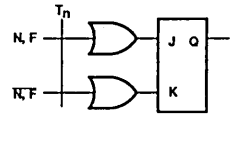
ACTION	CODE
TOGGLE[6]	O



ACTION	CODE
SET	H

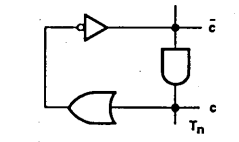


ACTION	CODE
RESET	L

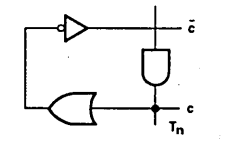


ACTION	CODE
NO CHANGE	–

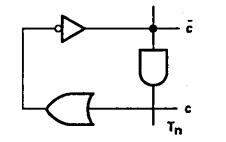
"COMPLEMENT" ARRAY – (C)



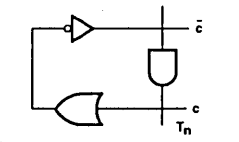
ACTION	CODE
INACTIVE[1, 3]	O



ACTION	CODE
GENERATE	A

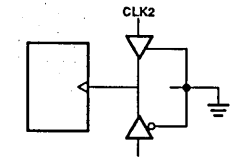


ACTION	CODE
PROPAGATE	•

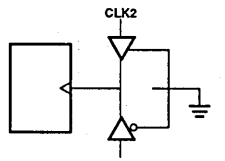


ACTION	CODE
TRANSPARENT	–

CLOCK OPTION – (CLK1/CLK2)



OPTION	CODE
CLK1 ONLY[1]	L



OPTION	CODE
CLK1 and CLK2[5]	H

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.
4. These states are not allowed when using INITIALIZATION option.
5. Input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using second clock option.
6. A single product term cannot drive more than 8 registers by itself when used in TOGGLE mode.

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

PLUS405 PROGRAM TABLE

AND

INACTIVE	0	INACTIVE	0
I, P	H	GENERATE	A
$\overline{I}$, $\overline{P}$	L	PROPAGATE	•
DON'T CARE	—	TRANSPARENT	—

Im, Ps — Cn

OR

INACTIVE OR TOGGLE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	—	

OPTIONS

INIT	H	INIT/$\overline{OE}$
$\overline{OE}$	L	
CLK1 ONLY	L	CLK1/CLK2
CLK1 AND 2	H	

CUSTOMER NAME_____ SIGNETICS DEVICE #_____ CF (XXXX) CUSTOMER SYMBOLIZED PART #_____

REV_____ DATE_____ PROGRAM TABLE_____

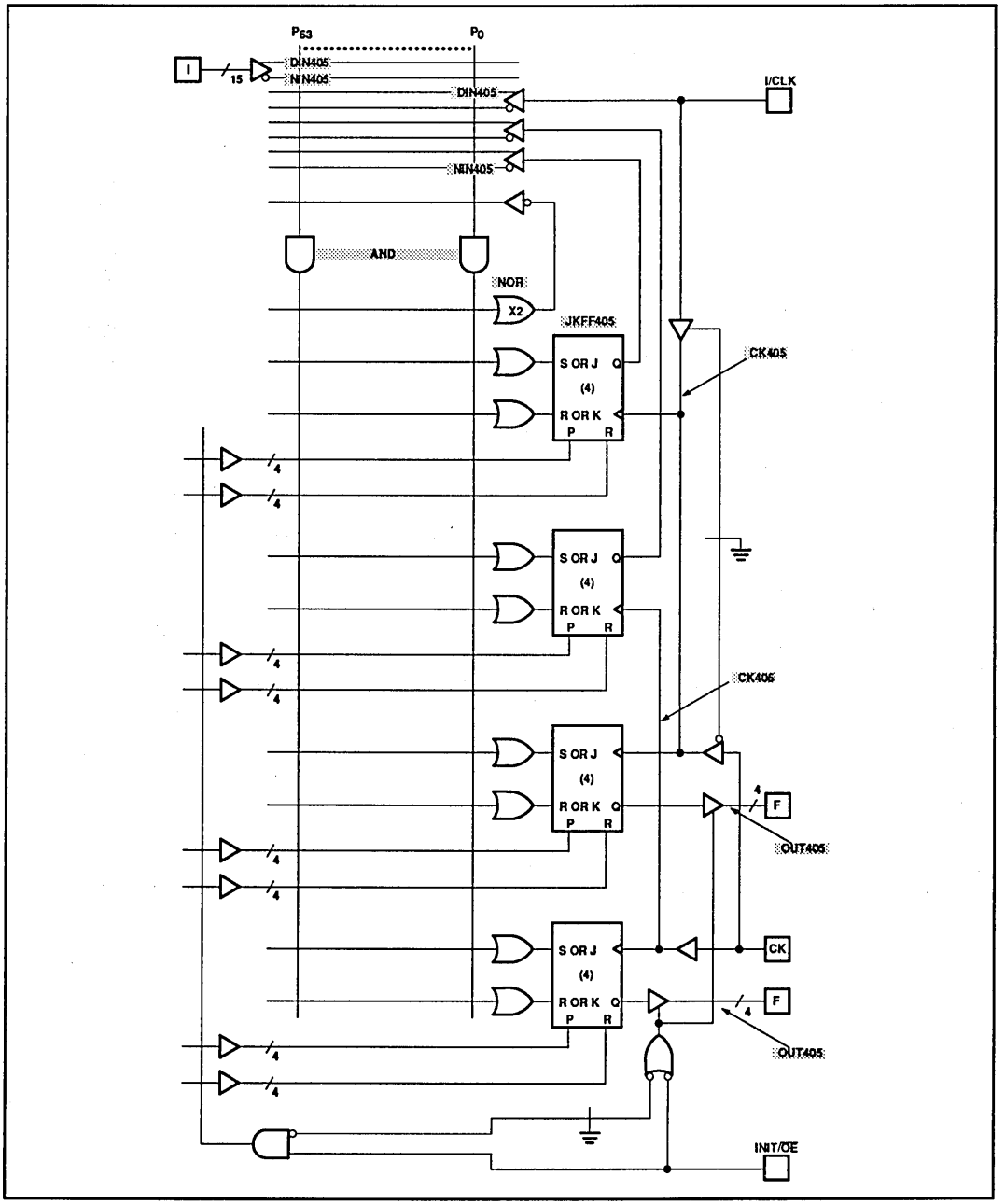
	AND																				OR															
	COMP. ARRAY		INPUT (Im)														CLOCK 1/2 PRESENT STATE (Ps)								INITIALIZATION/OUTPUT ENABLE NEXT STATE (Ns)								OUTPUT (Fr)							
	C1	C0	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0	P7	P6	P5	P4	P3	P2	P1	P0	N7	N6	N5	N4	N3	N2	N1	N0	F7	F6	F5	F4	F3	F2	F1	F0
0																																								
1																																								
2																																								
3																																								
4																																								
5																																								
6																																								
7																																								
8																																								
9																																								
10																																								
11																																								
12																																								
13																																								
14																																								
15																																								
16																																								
17																																								
18																																								
19																																								
20																																								
21																																								
22																																								
23																																								
24																																								
25																																								
26																																								
27																																								
28																																								
29																																								
30																																								
31																																								
32																																								
33																																								
34																																								
35																																								
36																																								
37																																								
38																																								
39																																								
40																																								
41																																								
42																																								
43																																								
44																																								
45																																								
46																																								
47																																								
48																																								
49																																								
50																																								
51																																								
52																																								
53																																								
54																																								
55																																								
56																																								
57																																								
58																																								
59																																								
60																																								
61																																								
62																																								
63																																								
PIN NO.			20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9																	10	11	12	13	15	16	17	18
PIN LABELS																																								

NOTES:
1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the IN/E and H for the clock option, exists in the table, shown BLANK instead for clarity.
2. Unused Cn Im, and Ps bits are normally programmed Don't Care (—).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (—) for maximum speed.

Programmable logic sequencers (16 × 64 × 8)

PLUS405–37/–45

SNAP RESOURCE SUMMARY DESIGNATIONS



Philips Components–Signetics

Document No.	
ECN No.	
Date of Issue	October 1990
Status	Preliminary Specification
Programmable Logic Devices	

PLUS405–55

Programmable logic sequencer (16 × 64 × 8)

DESCRIPTION

The PLUS405–55 device is a bipolar, programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 64 AND gates are connected to the 16 external dedicated inputs (I_0 - I_{15}) and to the feedback paths of the 8 on-chip State Registers (Q_{P0} - Q_{P7}). Two complement arrays support complex IF-THEN-ELSE state transitions with a single product term (input variables C_0, C_1).

All state transition terms can include True, False and Don't Care states of the controlling state variables. All AND gates are merged into the programmable OR array to issue the next-state and next-output commands to their respective registers. Because the OR array is programmable, any one or all of the 64 transition terms can be connected to any or all of the State and Output Registers.

All state (Q_{P0} - Q_{P7}) and output (Q_{F0} - Q_{F7}) registers are edge-triggered, clocked J-K flip-flops, with Asynchronous Preset and Reset options. The PLUS405 architecture provides the added flexibility of the J-K toggle function which is indeterminate on S-R flip-flops. Each register may be individually programmed such that a specific Preset-Reset pattern is initialized when the initialization pin is raised to a logic level "1". This feature allows the state machine to be asynchronously initialized to known internal state and output conditions prior to proceeding through a sequence of state transitions. Upon power-up, all registers are unconditionally preset to "1". If desired, the initialization input pin (INIT) can be converted to an Output Enable ($\overline{OE}$) function as an additional user-programmable feature.

Availability of two user-programmable clocks allows the user to design two independently clocked state machine functions consisting of four state and four output bits each.

Order codes are listed in the Ordering Information Table.

FEATURES

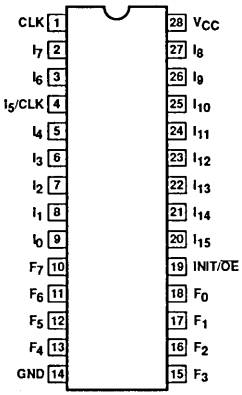
- 62.5MHz minimum guaranteed clock rate
- 55MHz minimum guaranteed operating frequency ($1/(t_{IS1} + t_{CKO1})$)
- Functional superset of PLS105/105A
- Field-programmable (Ti-W fusible link)
- 16 input variables
- 8 output functions
- 64 transition terms
- 8-bit State Register
- 8-bit Output Register
- 2 transition Complement Arrays
- Multiple clocks
- Programmable Asynchronous Initialization or Output Enable
- Power-on preset of all registers to "1"
- "On-chip" diagnostic test mode features for access to state and output registers
- 950mW power dissipation (typ.)
- TTL compatible
- J-K or S-R flip-flop functions
- Automatic "Hold" states
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator contollers
- Security locking systems
- Counters
- Shift registers

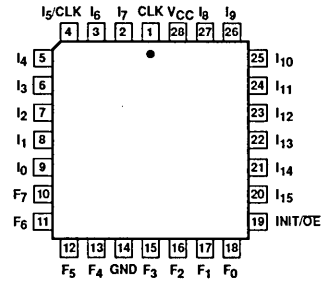
PIN CONFIGURATIONS

N Package



N = Plastic DIP (600mil-wide)

A Package



A = Plastic Leaded Chip Carrier

Programmable logic sequencer ($16 \times 64 \times 8$)

PLUS405–55

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK1	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers. Pin 1 only clocks P_{0-3} and F_{0-3} if Pin 4 is also being used as a clock.	Active-High (H)
2, 3, 5–9, 26–27 20–22	$I_0 - I_4$, I_7, I_6 $I_8 - I_9$ $I_{13} - I_{15}$	**Logic Inputs:** The 12 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/Low (H/L)
4	CLK2	**Logic Input/Clock:** A user programmable function:	
		• **Logic Input:** A 13th external logic input to the AND array, as above.	Active-High/Low (H/L)
		• **Clock:** A 2nd clock for the State Registers P_{4-7} and Output Registers F_{4-7}, as above. Note that input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using Pin 4 as a Clock.	Active-High (H)
23	I_{12}	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I_{12} is held at +10V, device outputs $F_0 - F_7$ reflect the contents of State Register bits $P_0 - P_7$. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
24	I_{11}	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{11} is held at +10V, device outputs $F_0 - F_7$ become direct inputs for State Register bits $P_0 - P_7$; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0 - F_7$ into the State Register bits $P_0 - P_7$. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
25	I_{10}	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{10} is held at +10V, device outputs $F_0 - F_7$ become direct inputs for Output Register bits $Q_0 - Q_7$; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0 - F_7$ into the Output Register bits $Q_0 - Q_7$. The contents of each State Register remains unaltered.	Active-High/Low (H/L)
10–13 15–18	$F_0 - F_7$	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register Bits $Q_0 - Q_7$, when enabled. When I_{12} is held at +10V, $F_0 - F_7 = (P_0 - P_7)$. When I_{11} is held at +10V, $F_0 - F_7$ become inputs to State Register bits $P_0 - P_7$. When I_{10} is held at +10V, $F_0 - F_7$ become inputs to Output Register bits $Q_0 - Q_7$.	Active-High (H)
19	INIT/$\overline{OE}$	**Initialization or Output Enable Input:** A user programmable function:	
		• **Initialization:** Provides an asynchronous preset to logic "1" or reset to logic "0" of all State and Output Register bits, determined individually for each register bit through user programming. INIT overrides Clock, and when held High, clocking is inhibited and $F_0 - F_7$ and $P_0 - P_7$ are in their initialization state. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after INIT goes Low. See timing definition for t_{NVCK} and t_{VCK}.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers $F_0 - F_7$ from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

TRUTH TABLE [1, 2, 3, 4, 5, 6, 7]

V_{CC}	OPTION										
	INIT	$\overline{OE}$	I_{10}	I_{11}	I_{12}	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H/L	H/L	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

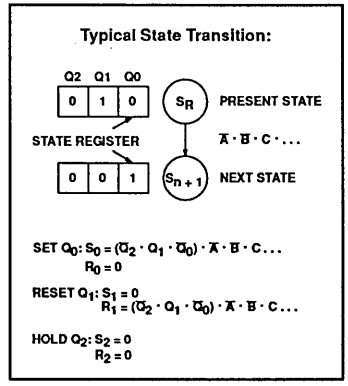
1. Positive Logic:
   S/R (or J/K) = $T_0 + T_1 + T_2 + \ldots T_{63}$
   $T_n = (C_0, C_1)(I_0, I_1, I_2, \ldots)(P_0, P_1, \ldots P_7)$
2. Either Initialization (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤5.5V)
6. H/L implies that either a High or a Low can occur, depending upon user-programmed selection (each State and Output Register individually programmable).
7. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

A factory-shipped virgin device contains all fusible links intact, such that:

1. INIT/$\overline{OE}$ is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user-programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are inactive (0).
3. All S/R (or J/K) flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array preprogrammed with a standard test pattern.
5. Clock 2 is inactive.

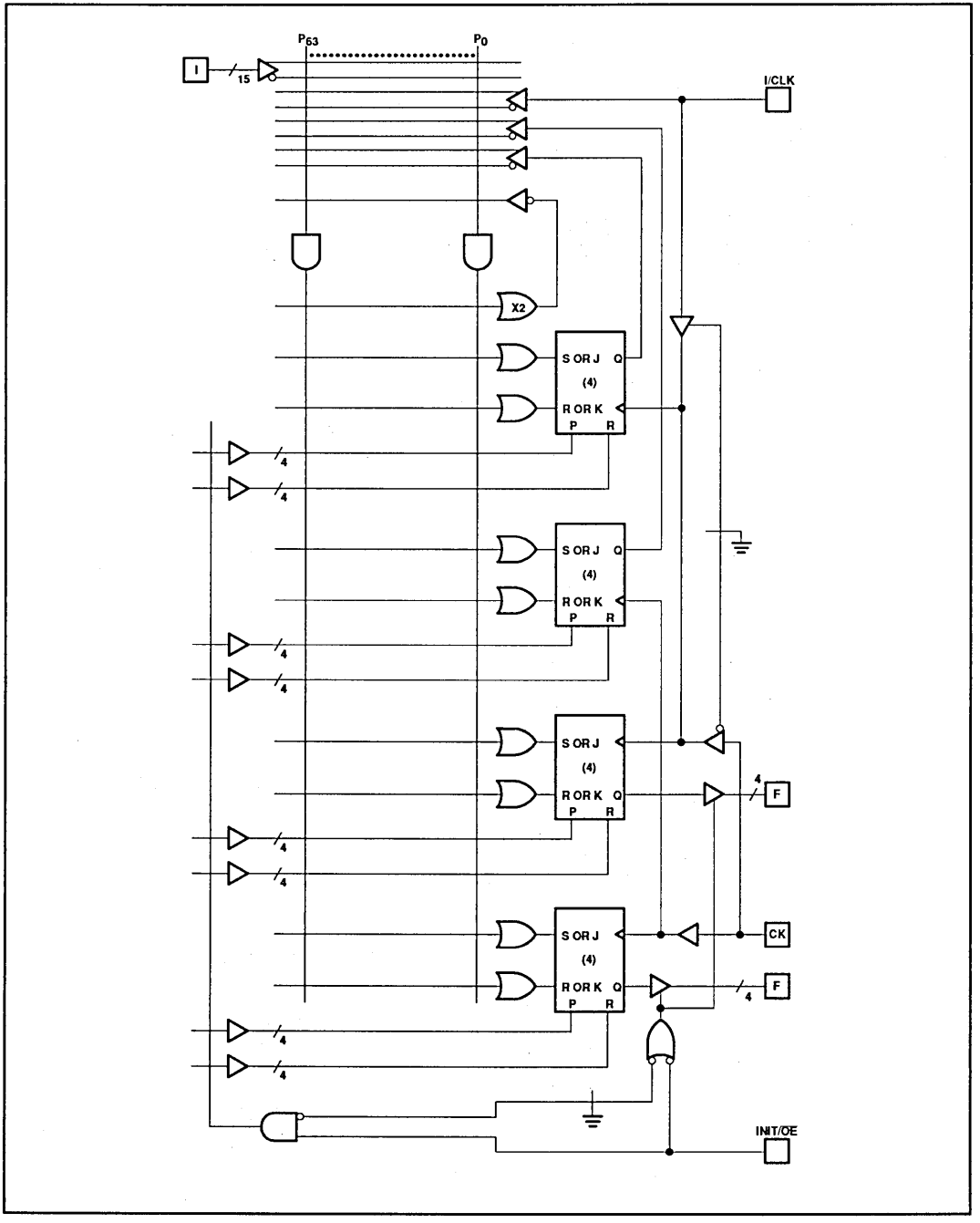
LOGIC FUNCTION



Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

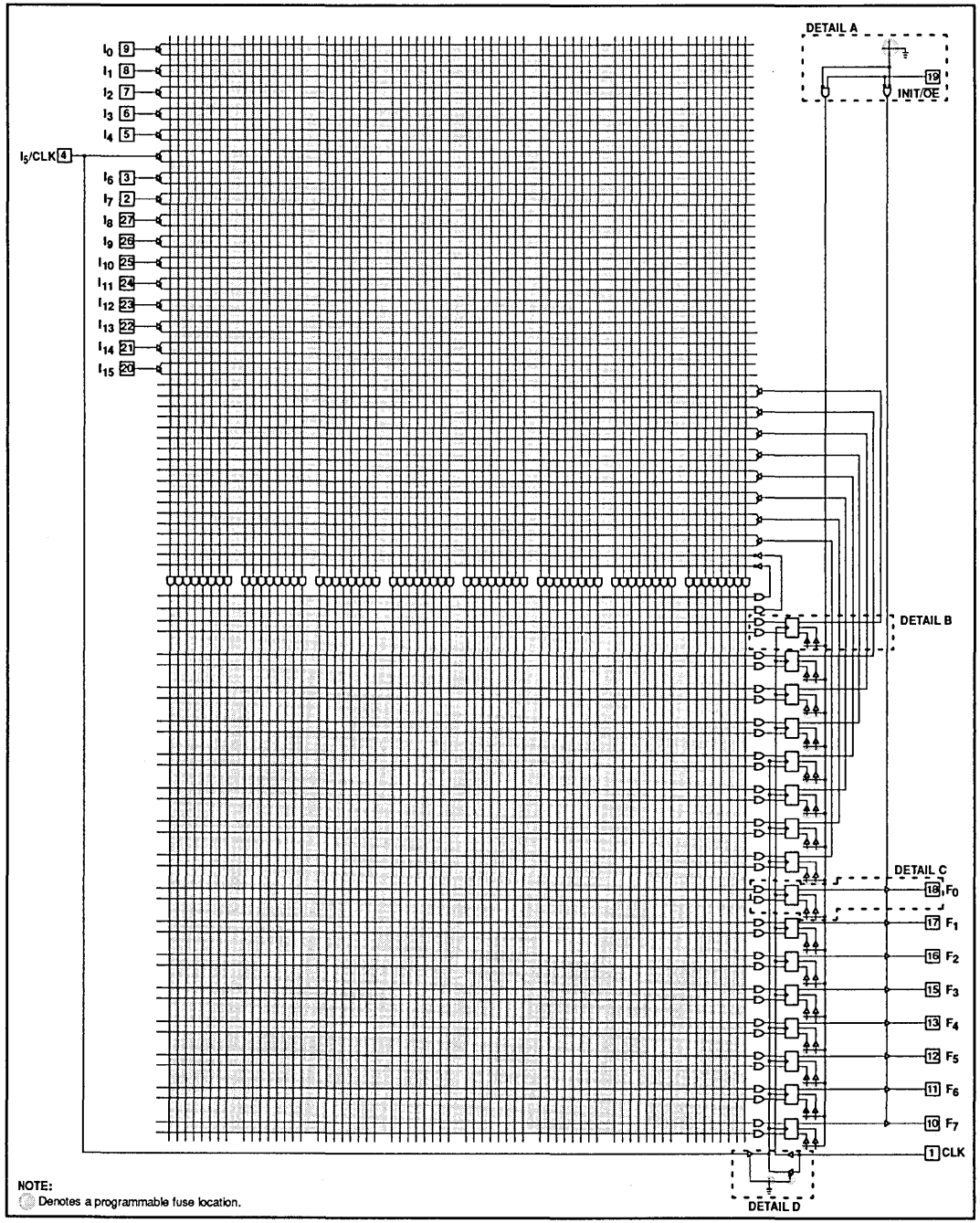
FUNCTIONAL DIAGRAM



Programmable logic sequencer ($16 \times 64 \times 8$)

PLUS405–55

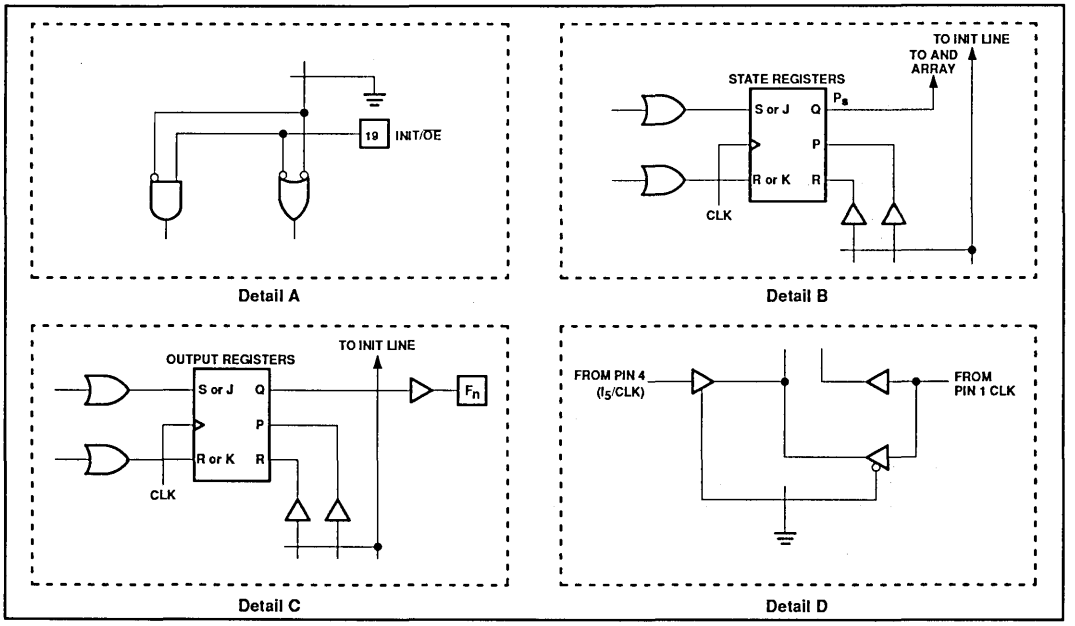
LOGIC DIAGRAM



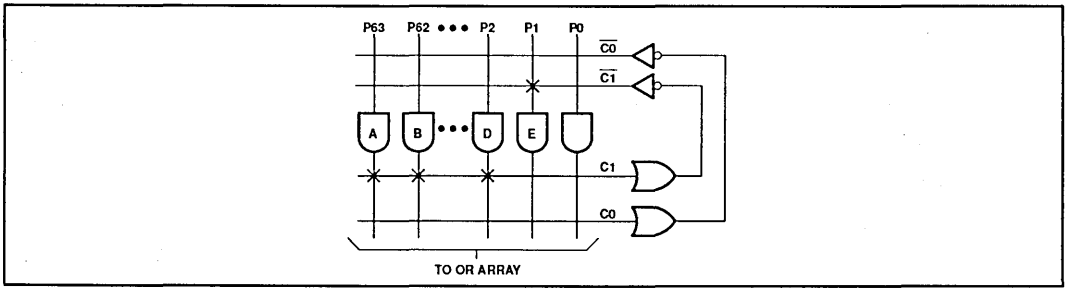
Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

DETAILS FOR REGISTERS FOR PLUS405



COMPLEMENT ARRAY DETAIL



The Complement Array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A • /B • /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The Complement Array is a single OR gate with inputs from the AND array. The output of the Complement Array is inverted and fed back to the AND array (NOR). The output of the array will be Low if any one or more of the AND terms connected to it are active (High). If, however, all the connected terms are inactive (Low), which is a classic unknown state, the output of the Complement Array will be High.

Consider the Product Terms A, B and D that represent defined states. They are also connected to the input of the Complement Array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to Product Term E, which could be used in turn to reset the state machine to a known state. Without the Complement Array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that the PLUS405 sequencers have 2 Complement Arrays which allow the user to design 2 independent Complement functions. This is particularly useful if 2 independent state machines have been implemented on one device.

Note that use of the Complement Array adds an additional delay path through the device. Please refer to the AC Electrical Characteristics for details.

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

ORDERING INFORMATION

DESCRIPTION	OPERATING FREQUENCY	ORDER CODE
28-Pin Plastic DIP (600mil-wide)	55MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–55N
28-Pin Plastic Leaded Chip Carrier	55MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–55A

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_A	Operating temperature range	0 to +75	°C
T_{STG}	Storage temperature range	–65 to +150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = V_{CC}		<1	25	µA
I_{IL}	Low	V_{CC} = MAX, V_{IN} = 0.45V		–20	–250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{CC} = MAX, V_{OUT} = 2.7V		1	40	µA
		V_{CC} = MAX, V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit[3,4]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		190	225	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{CC} = 5.0V, V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short-circuit should not exceed one second.
5. I_{CC} is measured with the INIT/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS			UNIT
				MIN	TYP[1]	MAX	
Pulse width							
t_{CKH1}	Clock High; CLK1 (Pin 1)	CK+	CK–	7.5	6		ns
t_{CKL1}	Clock Low; CLK1 (Pin 1)	CK–	CK+	7.5	6		ns
t_{CKP1}	CLK1 Period	CK+	CK+	15	12		ns
t_{CKH2}	Clock High; CLK2 (Pin 4)	CK+	CK–	7.5	6		ns
t_{CKL2}	Clock Low; CLK2 (Pin 4)	CK–	CK+	7.5	6		ns
t_{CKP2}	CLK2 Period	CK +	CK +	15	12		ns
t_{INITH}	Initialization pulse	INIT–	INIT+	12	8		ns
Setup time							
t_{IS1}	Input	Input ±	CK+	10	9		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK+	18	15		ns
t_{VS}	Power-on preset	V_{CC}+	CK–	0	–10		ns
t_{VCK}	Clock resume (after Initialization)	INIT–	CK–	0	–5		ns
t_{NVCK}	Clock lockout (before Initialization)	CK–	INIT–	12	5		ns
Hold time							
t_{IH}	Input	CK+	Input ±	0	–5		ns
Propagation delay							
t_{CKO1}	Clock1 (Pin 1)	CK1+	Output ±		6.5	8	ns
t_{CKO2}	Clock2 (Pin 4)	CK2+	Output ±		6.5	8	ns
t_{OE}[2]	Output Enable	OE–	Output –		6.5	8	ns
t_{OD}[2]	Output Disable	OE+	Output +		6.5	8	ns
t_{INIT}	Initialization	INIT+	Output +		12	18	ns
t_{PPR}	Power-on Preset	V_{CC} +	Output +		0	10	ns

Notes on following page

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

AC ELECTRICAL CHARACTERISTICS (Continued)

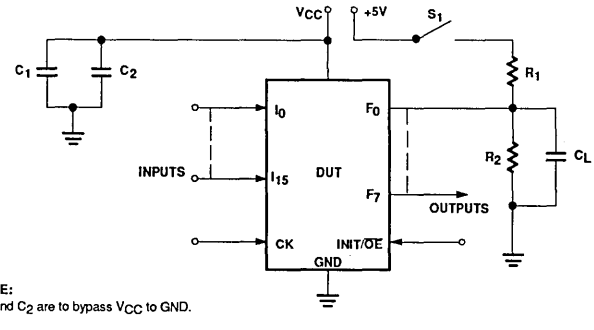
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Frequency of operation							
f_{MAX1}	CLK1; (without Complement Array) $\left(\frac{1}{t_{IS1} + t_{CKO1}}\right)$	Input ±	Output ±	55.6	64.5		MHz
f_{MAX2}	CLK2; (without Complement Array) $\left(\frac{1}{t_{IS1} + t_{CKO2}}\right)$	Input ±	Output ±	55.6	64.5		MHz
f_{MAX3}	CLK1; (with Complement Array) $\left(\frac{1}{t_{IS2} + t_{CKO1}}\right)$	Input through Complement Array ±	Output ±	38.5	46.5		MHz
f_{MAX4}	CLK2; (with Complement Array) $\left(\frac{1}{t_{IS2} + t_{CKO2}}\right)$	Input through Complement Array ±	Output ±	38.5	46.5		MHz
f_{MAX5}	Internal feedback without Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	66.7	83.3		MHz
f_{MAX6}	Internal feedback with Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{IS2}}\right)$	Register Output through Complement Array ±	Register Input ±	55.6	66.7		MHz
f_{CLK}	Minimum guaranteed Clock frequency	CK +	CK +	66.7	83.3		MHz

NOTES:

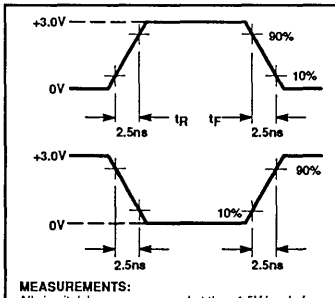
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays and setup times re measured and specified under worst case conditions.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



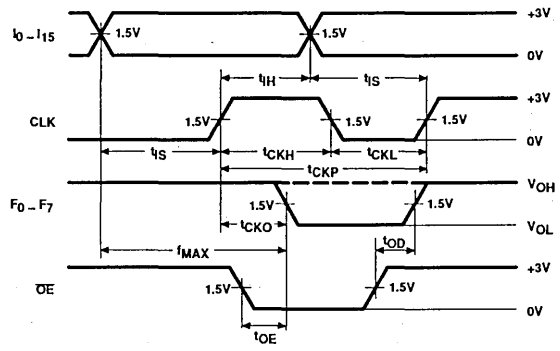
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

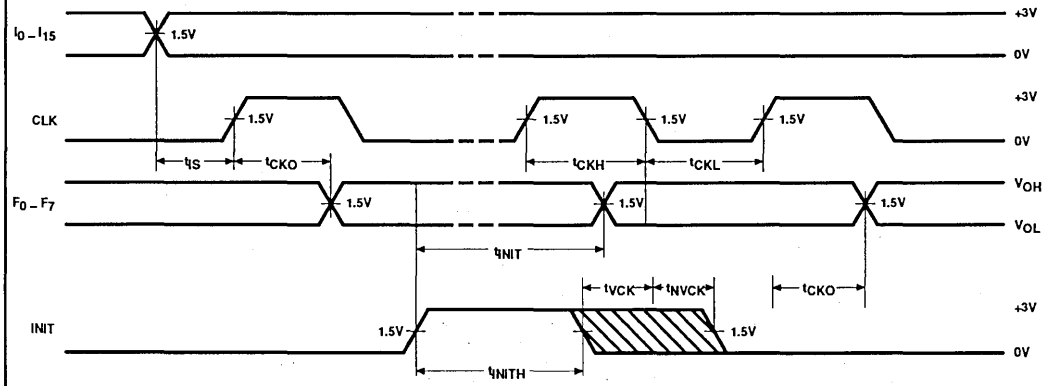
Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

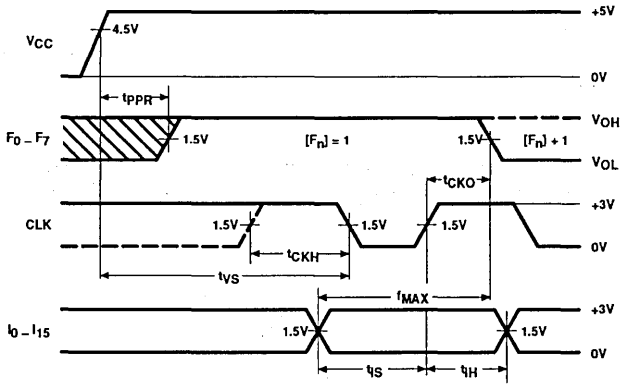
TIMING DIAGRAMS



Sequential Mode



Asynchronous Initialization

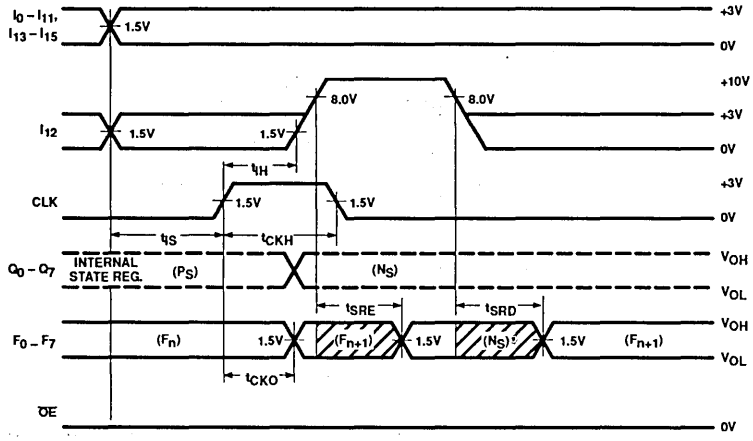


Power-On Preset

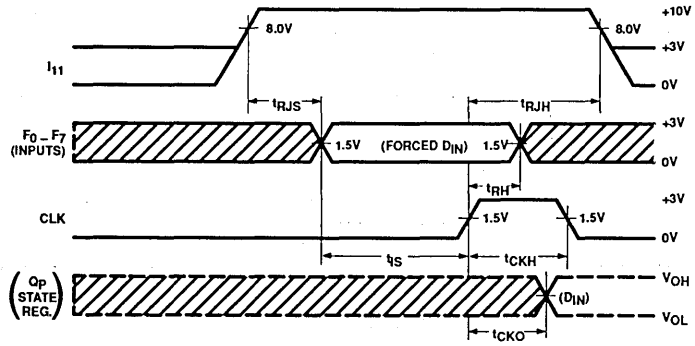
Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

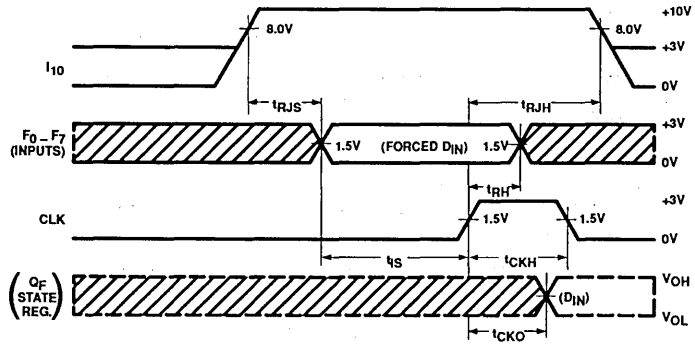
TIMING DIAGRAMS (Continued)



Diagnostic Mode – State Register Outputs



Diagnostic Mode – State Register Input Jam



Diagnostic Mode – Output Register Input Jam

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

TIMING DEFINITIONS

SYMBOL	PARAMETER
$t_{CKH1,2}$	Width of input clock pulse.
$t_{CKP1,2}$	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
$t_{CKO1,2}$	Delay between positive transition of Clock and when Outputs become valid (with INIT/$\overline{OE}$ Low).
t_{PPR}	Delay between V_{CC} (after power–on) and when Outputs become preset at "1".
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{RJH}	Required delay between positive transition of clock, and return of input I_{10} or I_{11} from Diagnostic Mode (8V).
$f_{MAX1,2}$	Minimum guaranteed operating frequency; input to output (CLK1 and CLK2).
$f_{MAX3,4}$	Minimum guaranteed operating frequency; input through Complement Array, to output (CLK1 and CLK2).
f_{MAX5}	Minimum guaranteed internal operating frequency; with internal feedback from state register to state register.

SYMBOL	PARAMETER
f_{MAX6}	Minimum guaranteed internal operating frequency with Complement Array, with internal feedback from state register through Complement Array, to state register.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency).
$t_{CKL1,2}$	Interval between clock pulses.
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{SRE}	Delay between input I_{12} transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{RJS}	Required delay between inputs I_{11} or I_{10} transition to Diagnostic Mode (8V), and when the output pins become available as inputs.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{INITH}	Width of initialization input pulse.
t_{VS}	Required delay between V_{CC} (after power–on) and negative transition of Clock preceding first reliable clock pulse.
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF–state.
t_{INIT}	Delay between positive transition of Initialization and when Outputs become valid.
t_{SRD}	Delay between input I_{12} transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{RH}	Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
t_{VCK}	Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding first reliable clock pulse.

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

LOGIC PROGRAMMING

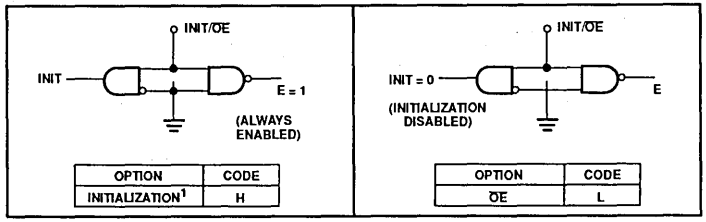
The PLUS405–55 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Signetics AMAZE, SLICE and SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS405–55 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS405–55 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by AMAZE and SLICE only. Both AMAZE and SLICE design packages are available, free of charge, to qualified users.

To implement the desired logic functions, each logic variable (I, B, P, S, T, etc.) from the logic equations is assigned a symbol. TRUE, COMPLEMENT, PRESET, RESET, OUTPUT ENABLE, INACTIVE, etc., symbols are defined below.

INITIALIZATION/$\overline{OE}$ OPTION – (INIT/$\overline{OE}$)



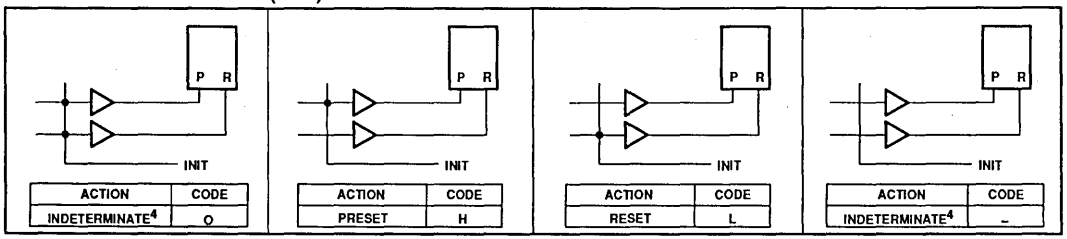
OPTION	CODE
INITIALIZATION[1]	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING THE PLUS405:

The PLUS405 has a power–up preset feature. This feature insures that the device will power–up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

INITIALIZATION OPTION – (INIT)



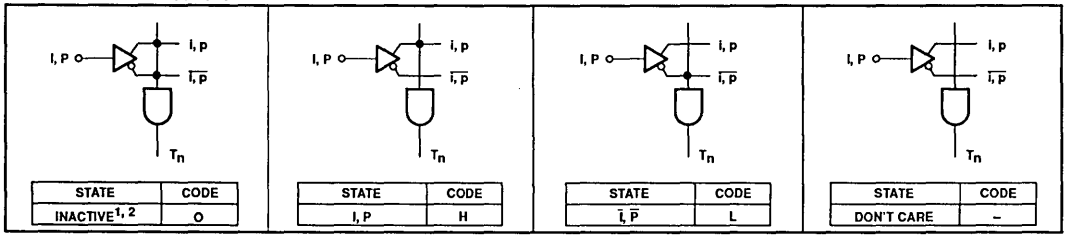
ACTION	CODE
INDETERMINATE[4]	Q

ACTION	CODE
PRESET	H

ACTION	CODE
RESET	L

ACTION	CODE
INDETERMINATE[4]	–

"AND" ARRAY – (I), (P)



STATE	CODE
INACTIVE[1, 2]	Q

STATE	CODE
I, P	H

STATE	CODE
$\overline{I, P}$	L

STATE	CODE
DON'T CARE	–

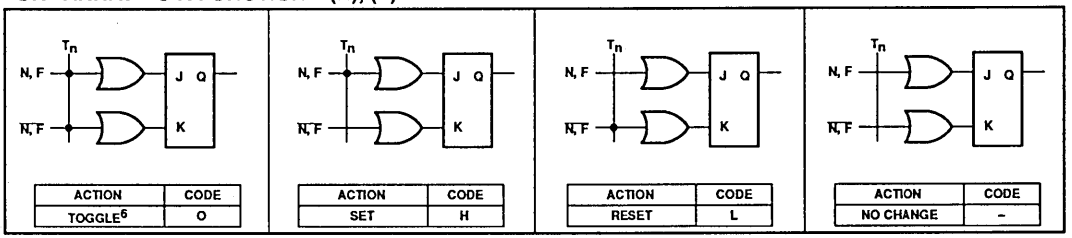
Notes are on next page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

"OR" ARRAY – J-K FUNCTION – (N), (F)



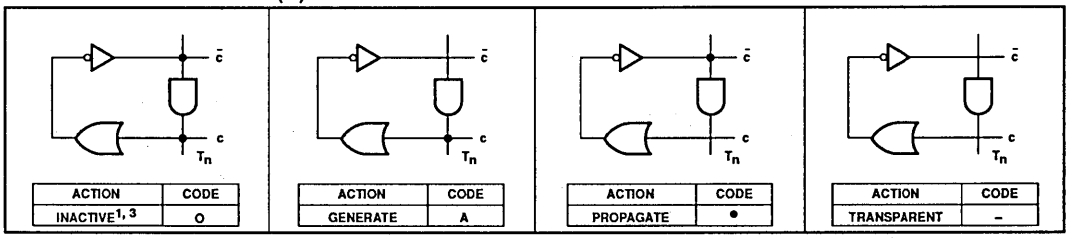
ACTION	CODE
TOGGLE[6]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



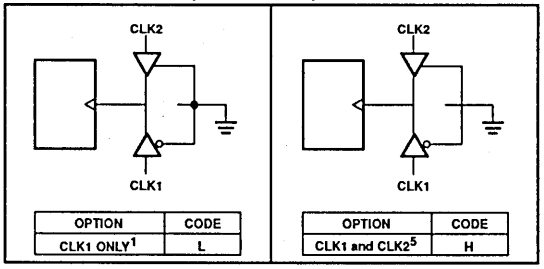
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

CLOCK OPTION – (CLK1/CLK2)



OPTION	CODE
CLK1 ONLY[1]	L

OPTION	CODE
CLK1 and CLK2[5]	H

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.
4. These states are not allowed when using INITIALIZATION option.
5. Input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using second clock option.
6. A single product term cannot drive more than 8 registers by itself when used in TOGGLE mode.

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

PLUS405 PROGRAM TABLE

AND

INACTIVE	0		INACTIVE	0	
I, P	H	Im, Ps	GENERATE	A	Cn
$\overline{I}$, $\overline{P}$	L		PROPAGATE	•	
DON'T CARE	–		TRANSPARENT	–	

OR

INACTIVE OR TOGGLE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	–	

OPTIONS

INIT	H	INIT/$\overline{OE}$
$\overline{OE}$	L	
CLK1 ONLY	L	CLK1/CLK2
CLK1 AND 2	H	

CUSTOMER NAME _____ REV _____ DATE _____

SIGNETICS DEVICE # _____ CF (XXXX) _____ PROGRAM TABLE _____

CUSTOMER SYMBOLIZED PART # _____

	AND																				OR													
	COMP. ARRAY		INPUT (Im)															CLOCK 1/2	PRESENT STATE (Ps)								INITIALIZATION/OUTPUT ENABLE	NEXT STATE (Ns)							OUTPUT (Fr)							
	C1	C0	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0	P7	P6	P5	P4	P3	P2	P1	P0		N7	N6	N5	N4	N3	N2	N1	N0	F7	F6	F5	F4	F3	F2	F1	F0
0																																										
1																																										
2																																										
3																																										
4																																										
5																																										
6																																										
7																																										
8																																										
9																																										
10																																										
11																																										
12																																										
13																																										
14																																										
15																																										
16																																										
17																																										
18																																										
19																																										
20																																										
21																																										
22																																										
23																																										
24																																										
25																																										
26																																										
27																																										
28																																										
29																																										
30																																										
31																																										
32																																										
33																																										
34																																										
35																																										
36																																										
37																																										
38																																										
39																																										
40																																										
41																																										
42																																										
43																																										
44																																										
45																																										
46																																										
47																																										
48																																										
49																																										
50																																										
51																																										
52																																										
53																																										
54																																										
55																																										
56																																										
57																																										
58																																										
59																																										
60																																										
61																																										
62																																										
63																																										
PIN NO.			20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9																	10	11	12	13	15	16	17	18
PIN LABELS																																									

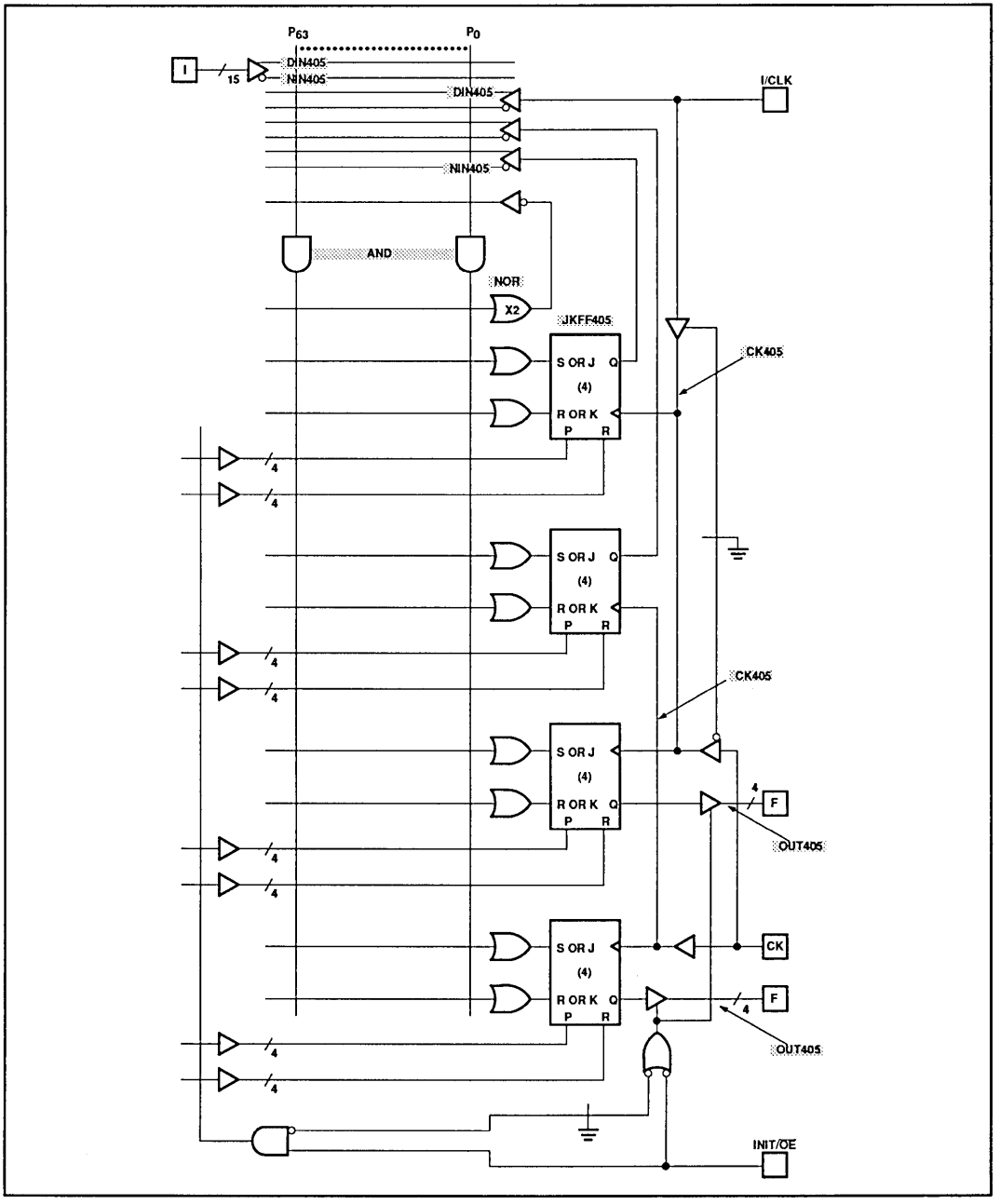
NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the IN/E and H for the clock option, exists in the table, shown BLANK instead for clarity.
2. Unused Cn Im, and Ps bits are normally programmed Don't Care (—).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (—) for maximum speed.

Programmable logic sequencer (16 × 64 × 8)

PLUS405–55

SNAP RESOURCE SUMMARY DESIGNATIONS



Section 6
Programmable Macro Logic Data Sheets

INDEX

Philips Components–Signetics

Document No.	853–1207
ECN No.	98513
Date of Issue	January 9, 1990
Status	Product Specification
Programmable Logic Devices	

PLHS501

Programmable macro logic

PML™

FEATURES

- Programmable Macro Logic device
- Full connectivity
- TTL compatible
- SNAP development system:
  - Supports third-party schematic entry formats
  - Macro library
  - Versatile netlist format for design portability
  - Logic, timing, and fault simulation
- SLICE development system:
  - Easy to learn and use
  - State or Boolean equation entry
  - Fuse table editor
  - Test vector editor
  - Boolean equation extractor
  - JEDEC fusemap compiler
  - Upgradeable to SNAP
- AMAZE development system:
  - Supports third-party schematic entry formats
  - State or Boolean equation entry
  - Logic, timing, and fault simulation
- Delay per internal NAND function = 6.5ns (typ)
- Testable in unprogrammed state
- Security fuse allows protection of proprietary designs

STRUCTURE

- NAND gate based architecture
  - 72 foldback NAND terms
- 136 input-wide logic terms
- 44 additional logic terms
- 24 dedicated inputs ($I_0 - I_{23}$)
- 8 bidirectional I/Os with individual 3-State enable:
  - 4 Active-High ($B_4 - B_7$)
  - 4 Active-Low ($\overline{B}_0 - \overline{B}_3$)
- 16 dedicated outputs:
  - 4 Active-High outputs
    O_0, O_1 with common 3-State enable
    O_2, O_3 with common 3-State enable
  - 4 Active-Low outputs:
    $\overline{O}_4$, $\overline{O}_5$ with common 3-State enable
    $\overline{O}_6$, $\overline{O}_7$ with common 3-State enable
  - 8 Exclusive-OR outputs:
    X_0, X_1 with common 3-State enable
    X_2, X_3 with common 3-State enable
    X_4, X_5 with common 3-State enable
    X_6, X_7 with common 3-State enable

DESCRIPTION

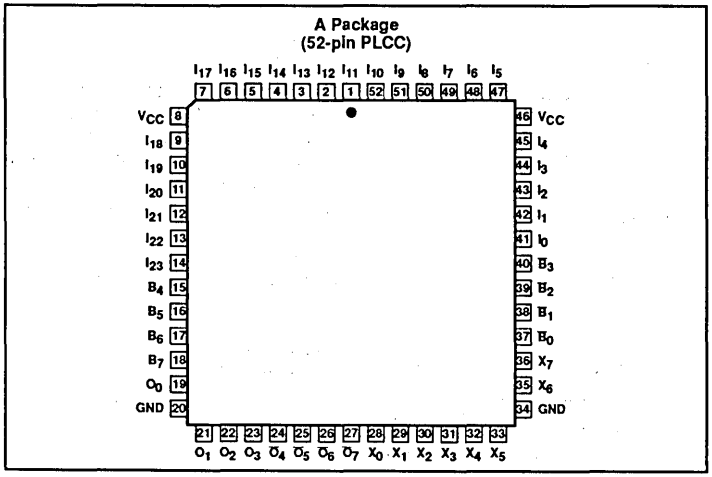
The PLHS501 is a high-density Bipolar Programmable Macro Logic device. PML incorporates a programmable NAND structure. The NAND architecture is an efficient method for implementing any logic function. The SNAP software development system provides a user friendly environment for design entry. SNAP eliminates the need for a detailed understanding of the PLHS501 architecture and makes it transparent to the user. PLHS501 is also supported on the Signetics SLICE and AMAZE software development systems.

The PLHS501 is ideal for a wide range of microprocessor support functions, including bus interface and control applications.

PML is a trademark of Philips Components–Signetics.

Programmable macro logic | PLHS501

PIN CONFIGURATION



ARCHITECTURE

The core of the PLHS501 is a programmable fuse array of 72 NAND gates. The output of each gate folds back upon itself and all other NAND gates. In this manner, full connectivity of all logic functions is achieved in the PLHS501. Any logic function can be created within the core of the device without wasting valuable I/O pins. Furthermore, a speed advantage is acquired by implementing multi-level logic within a fast internal core without incurring any delays from the I/O buffers.

DESIGN DEVELOPMENT TOOLS

SNAP

The SNAP Software Development System provides the necessary tools for designing with PML. SNAP provides the following:

- Schematic entry netlist generation from third-party schematic design packages such as OrCAD/SDT III™ and FutureNet™.
- Macro library for standard TTL functions and user defined functions
- Boolean equation entry
- State equation entry
- Syntax and design entry checking
- Simulator includes logic simulation, fault simulation and timing simulation.

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. The minimum system configuration for SNAP is 640K bytes of RAM and a hard disk.

SNAP provides primitive PML function libraries for third-party schematic design packages. Custom macro function libraries can be defined in schematic or equation form.

After the completion of a design, the software compiles the design for syntax and completeness. Complete simulation can be carried out using the different simulation tools available.

The programming data is generated in JEDEC format. Using the Device Programmer Interface (DPI) module of SNAP, the JEDEC fusemap is sent from the host computer to the device programmer.

SLICE

SLICE, which supports Signetics PLD line, is easy to understand and simple to use. Select a PLD, assign input and output pins and enter the desired equations in either Boolean or state form. SLICE then checks the equations for errors. It automatically generates a JEDEC-format fuse map for downloading to a PLD programmer.

Fully menu driven, SLICE incorporates a fuse table editor for making quick modifications to the design and a test vector editor for input of test vectors.

A built-in Boolean equation extractor allows existing PLDs to be used as the basis for a new design. the extractor reads JEDEC information from a PLD and creates a file containing the corresponding Boolean equations. The result can then be used to consolidate several PLD designs into a single, denser part.

And SLICE is upward compatible with Signetics extensive design suite, SNAP.

AMAZE

The AMAZE PLD Design Software development system also supports the PLHS501. AMAZE provides the following capabilities for the PLHS501:

- Schematic entry netlist conversion from third-party schematic software
- Boolean equation entry
- Logic and timing simulation
- Automatic test vector generation

AMAZE operates on an IBM PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.0 or higher. The minimum system configuration for AMAZE is 640K bytes of RAM and a hard disk.

AMAZE compiles the design after completion for syntax and completeness. Programming data is generated in JEDEC format.

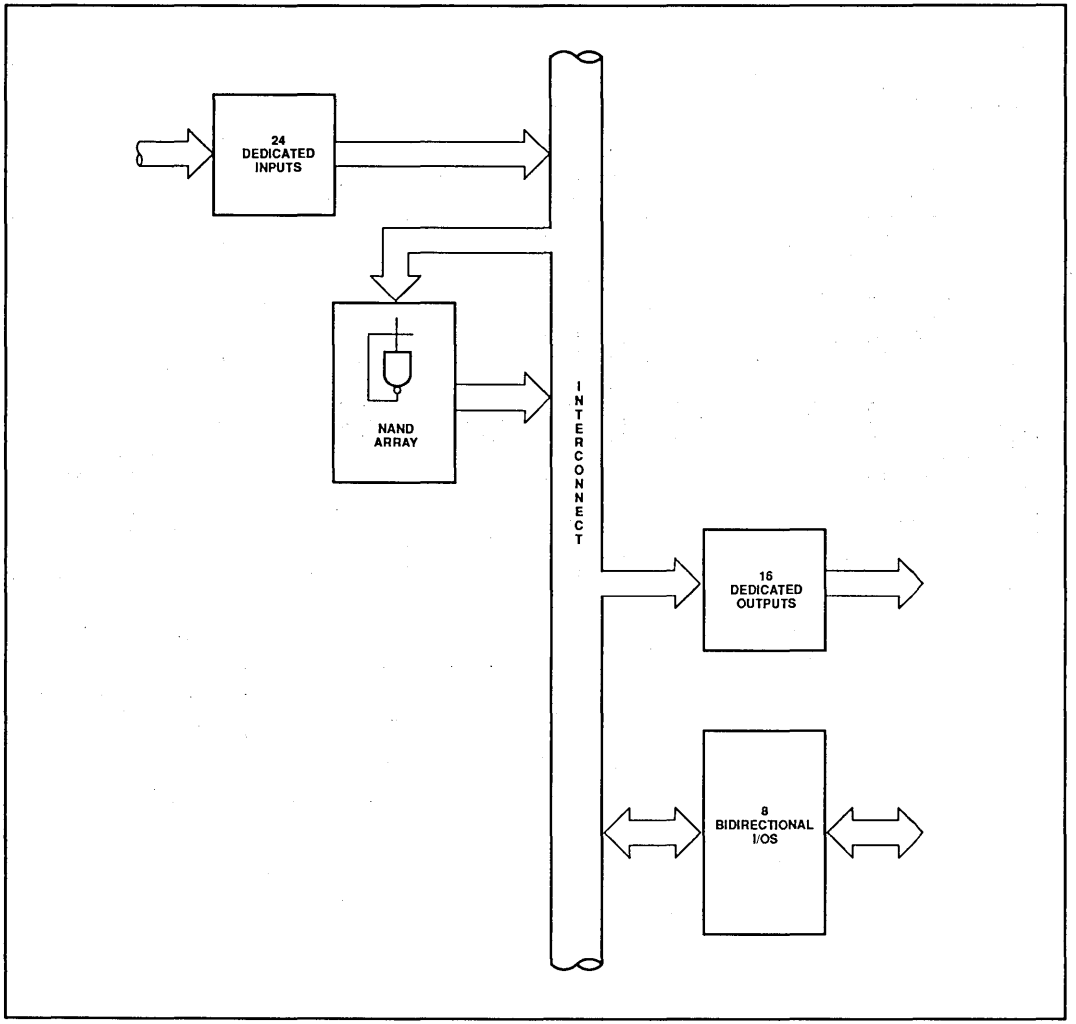
DESIGN SECURITY

The PLHS501 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

FutureNet is a trademark of FutureNet Corporation.
OrCAD/SDT is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

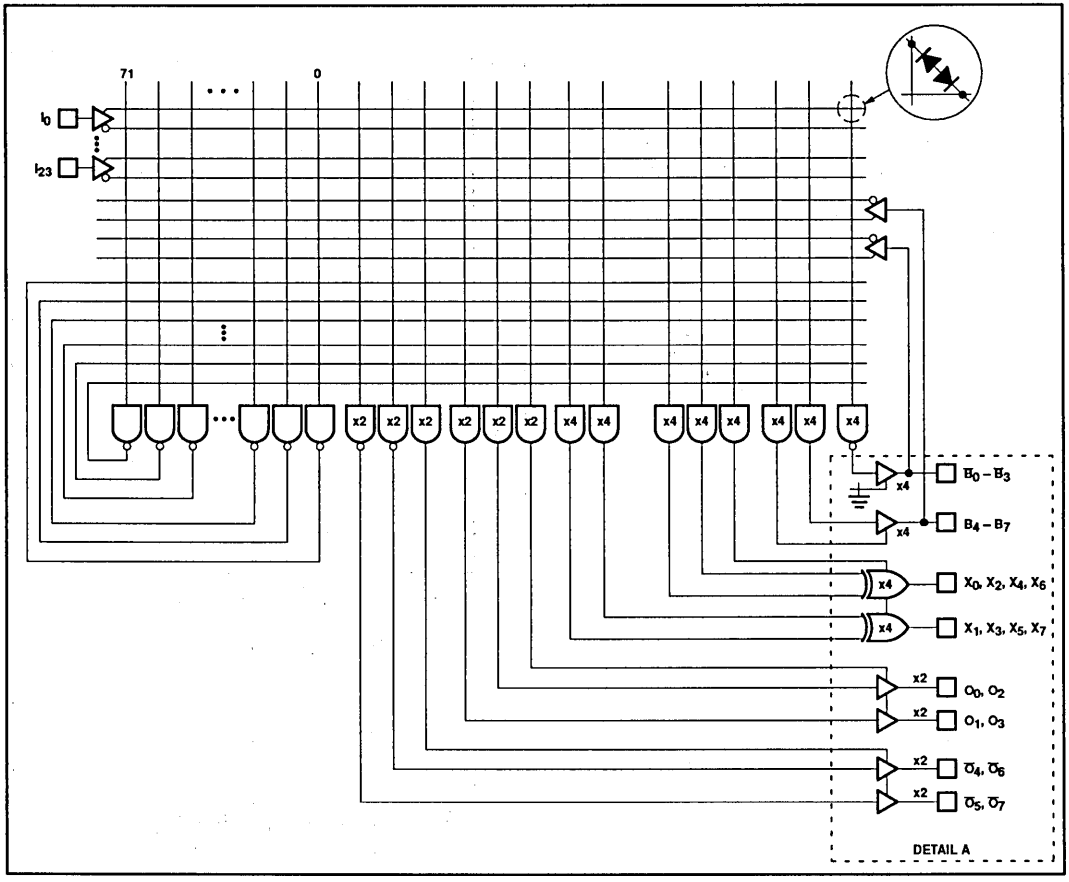
Programmable macro logic PLHS501

PLHS501 FUNCTIONAL BLOCK DIAGRAM

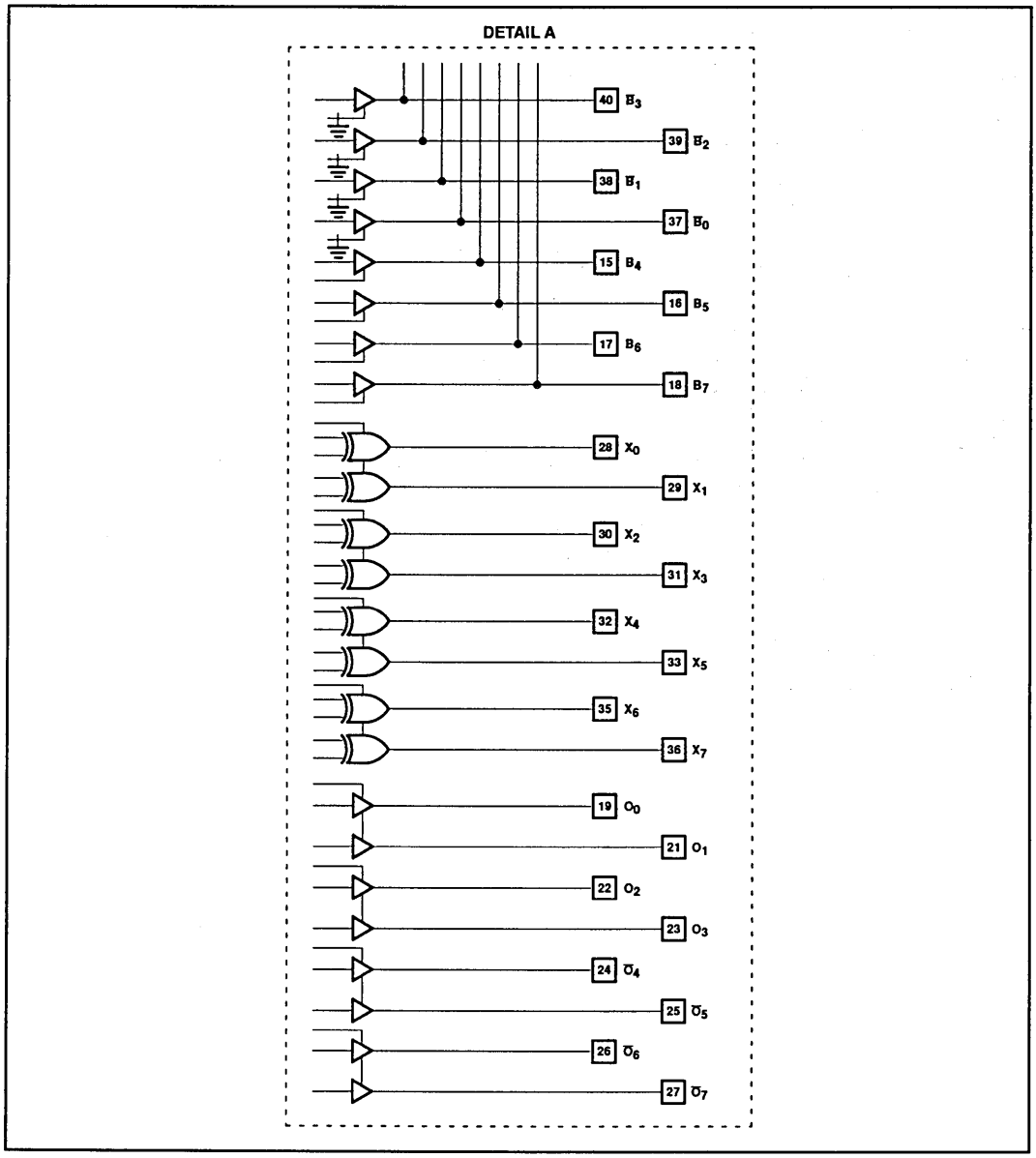


Programmable macro logic PLHS501

FUNCTIONAL DIAGRAM



Programmable macro logic PLHS501



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
52-Pin Plastic Leaded Chip Carrier	PLHS501A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All product terms are enabled.
2. All bidirectional (B) pins are outputs.
3. All outputs are enabled.
4. All outputs are Active-High **except** $\overline{B}_0 - \overline{B}_3$ (fusible I/O) and $\overline{O}_4 - \overline{O}_7$ which are Active-Low.

Programmable macro logic — PLHS501

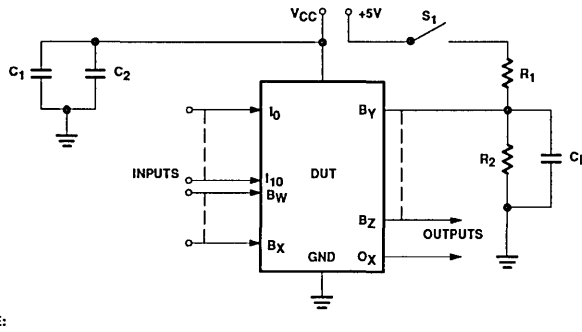
DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[2, 3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage						
V_{OL}	Low[2, 4]	V_{CC} = MIN I_{OL} = 10mA			0.45	V
V_{OH}	High[2, 5]	I_{OH} = –2mA	2.4			V
Input current						
I_{IL}	Low	V_{CC} = MAX V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = 5.5V			40	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[9]	V_{CC} = MAX V_{OUT} = 5.5V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		225	295	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5V V_{IN} = 2.0V		8		pF
C_B	I/O	V_{OUT} = 2.0V		15		pF

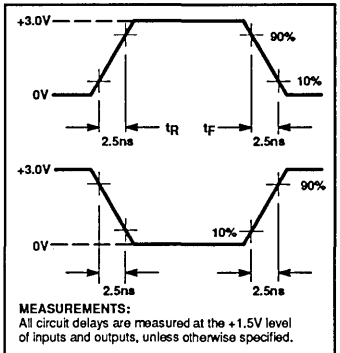
NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. For Pins 15 – 19, 21 – 27 and 37 – 40, V_{OL} is measured with Pins 5 and 41 = 8,75V, Pin 43 = 0V and Pins 42 and 44 = 4.5V. For Pins 28 – 33 and 35 – 36, V_{OL} is measured under same conditions EXCEPT Pin 44 = 0V.
5. V_{OH} is measured with Pins 5 and 41 = 8.75V, Pins 42 and 43 = 4.5V and Pin 44 = 0V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with all dedicated inputs at 0V and bidirectional and output pins open.
8. Measured at $V_T = V_{OL} + 0.5V$.
9. Leakage values are a combination of input and output leakage.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

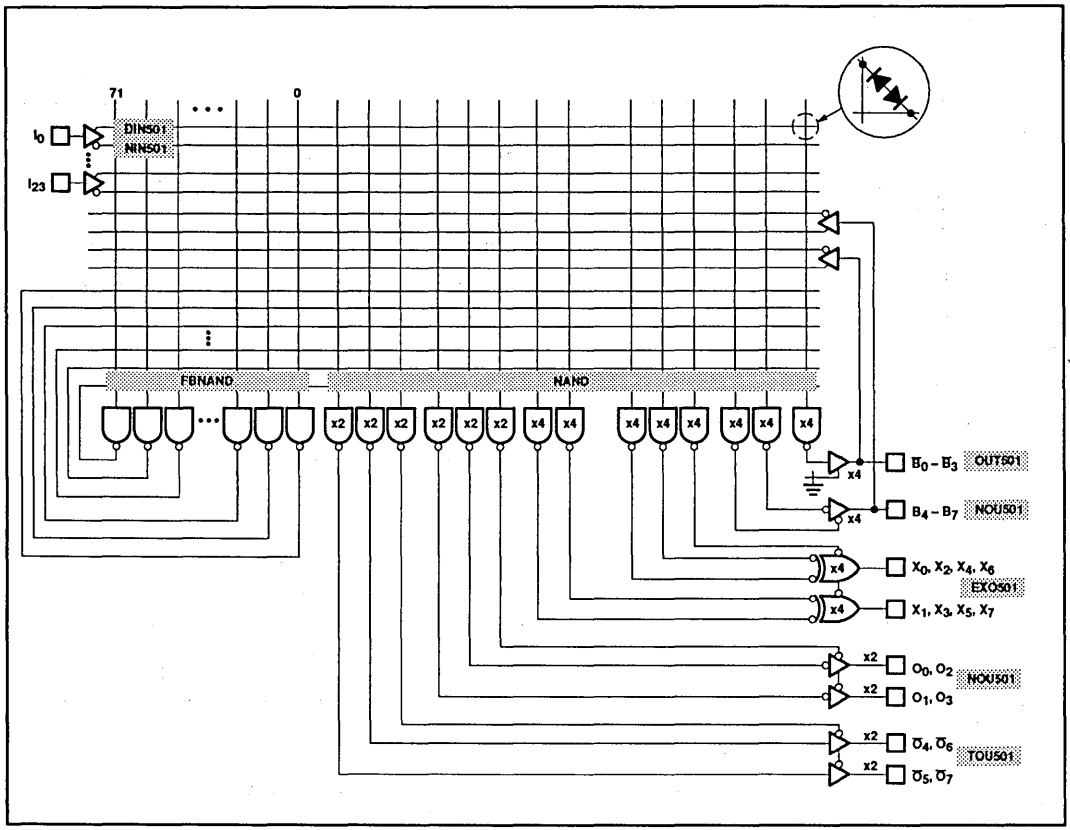
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

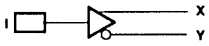
SNAP RESOURCE SUMMARY DESIGNATIONS



MACRO CELL SPECIFICATIONS[1] T_{amb} = 0°C to +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω

(SNAP Resource Summary Designations in Parantheses)

Input Buffer
(DIN501 [Non–Inverting], NIN501 [Inverting])

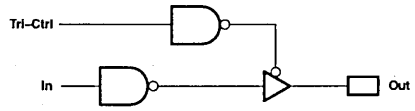


SYMBOL	LIMITS MIN	TYP	MAX	UNIT
Δt_{HL}	0.05	0.1	0.15	ns/p–term
Δt_{LH}	–0.02	–0.05	–0.08	ns/p–term

SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT	NOTES
t_{PHL}	X	I	4.5	5.5	6.5	ns	With 0 p–terms load
t_{PLH}	X	I	5	6	7.5	ns	
t_{PHL}	Y	I	2.5	3	3.5	ns	With 0 p–terms load
t_{PLH}	Y	I	4	4	4.5	ns	

Input Pins: 1 – 7, 9 – 14, 41 – 45, 48 – 52.
Bidirectional Pins: 15 – 18, 37 – 40.
Maximum internal fan–out: 16 p–terms on X or Y.

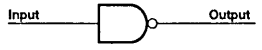
NAND Output Buffer with 3-State Control
(TOU501)



SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT
t_{PHL}	Out	In	8.5	14.0	17.5	ns
t_{PLH}	Out	In	8.5	14.0	16	ns
t_{OE}[2]	Out	Tri–Ctrl	8.5	15	18.5	ns
t_{OD}[2]	Out	Tri–Ctrl	8.5	12.5	17.0	ns

Output Pins: 24 – 27.

Internal Foldback NAND
(FBNAND)



SYMBOL	LIMITS MIN	TYP	MAX	UNIT
Δt_{PHL}	0.05	0.1	0.15	ns/p–term
Δt_{PLH}	–0.0	–0.05	–0.1	ns/p–term

SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT	NOTES
t_{PHL}	Out	Any	4.0	4.5	6.8	ns	With 0 p–terms load
t_{PLH}			5.5	6.5	8	ns	

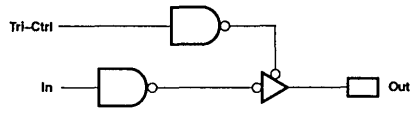
Maximum internal loading of 16 terms.

Notes are on following page.

MACRO CELL SPECIFICATIONS (Continued)

T_{amb} = 0°C to +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω
(SNAP Resource Summary Designations in Parantheses)

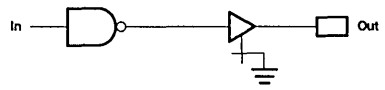
AND Output Buffer with 3-State Control (NOU501)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Output	In	8.0	11	13	ns
t_{PLH}	Output	In	8.0	11	13	ns
t_{OE}[2]	Out	Tri–Ctrl	8.5	15	18.5	ns
t_{OD}[2]	Out	Tri–Ctrl	8.5	12.5	17.0	ns

Bidirectional and Output Pins: 19, 21, 22, 23, 15 – 18.

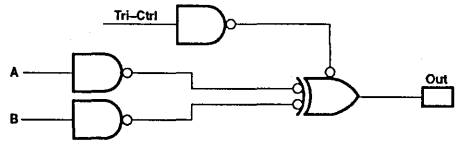
NAND Output Buffer (OUT501)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	8.5	14	17.5	ns
t_{PLH}	Out	In	8.5	14	16.0	ns

Bidirectional Pins: 37 – 40.

Ex–OR Output Buffer (EXO501)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	A or B	8.5	14	17.5	ns
t_{PLH}	Out	A or B	8.5	14	16.0	ns
t_{OE}[2]	Out	Tri–Ctrl	8.5	15	18.5	ns
t_{OD}[2]	Out	Tri–Ctrl	8.5	12.5	17.0	ns

Ex–OR Output Pins: 28 – 33.

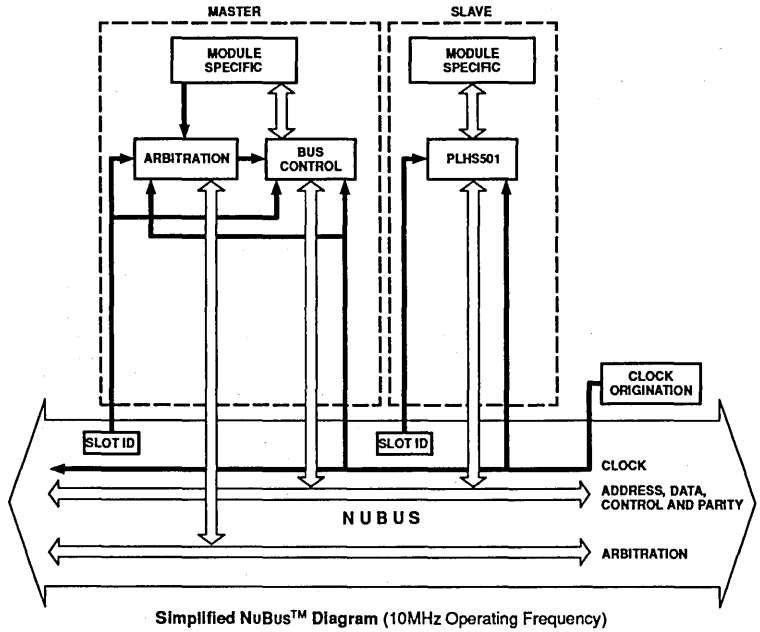
NOTES:

1. Limits are guaranteed with internal feedback buffers simultaneously switching cumulative maximum of eight outputs.
2. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

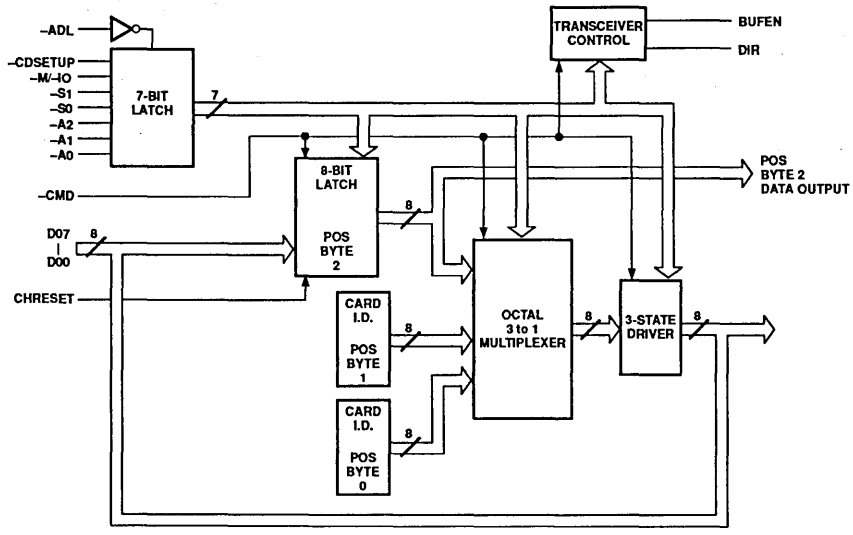
PLHS501 GATE AND SPEED ESTIMATE TABLE

FUNCTION	INTERNAL NAND EQUVALENT	TYPICAL t_{PD}	f_{MAX}	COMMENTS
Gates				
NANDs	1	6.5ns		For 1 to 32 input variables
ANDs	1	6.5ns		For 1 to 32 input variables
NORs	1	6.5ns		For 1 to 32 input variables
ORs	1	6.5ns		For 1 to 32 input variables
Decoders				
3-to-8	8	11ns		Inverted inputs available
4-to-16	16	11ns		Inverted inputs available
5-to-32	32	11ns		Inverted inputs available (24 chip outputs only)
Encoders				
8-to-3	15	11ns		Inverted inputs, 2 logic levels
16-to-4	32	11ns		Inverted inputs, 2 logic levels
32-to-5	41	11ns		Inverted inputs, 2 logic levels, factored solution.
Multiplexers				
4-to-1	5	11ns		Inverted inputs available
8-to-1	9	11ns		
16-to-1	17	11ns		
27-to-1	28	11ns		Can address only 27 external inputs - more if internal
Flip-Flops				
D-type Flip-Flop	6		30MHz	With asynchronous S-R
T-type Flip-Flop	6		30MHz	With asynchronous S-R
J-K-type Flip-Flop	10		30MHz	With asynchronous S-R
Adders				
8-bit	45	15.5ns		Full carry-lookahead (four levels of logic)
Barrel Shifters				
8-bit	72	11ns		2 levels of logic
Latches				
D-latch	3			2 levels of logic with one shared gate

APPLICATIONS



Simplified NuBus™ Diagram (10MHz Operating Frequency)



Block Diagram of Basic POS Implementation in PLHS501

NuBus is a trademark of Texas Instruments, Inc.

Philips Components–Signetics

Document No.	853–1405
ECN No.	98514
Date of Issue	January 9, 1990
Status	Product Specification
Programmable Logic Devices	

PLHS502

Programmable macro logic

PML™

FEATURES

- Programmable Macro Logic
- Full connectivity
- Delay per internal NAND function = 6.5ns
- Clock frequency = 40MHz
  Operating frequency = 33MHz
- SNAP development system eases design
  - Supports third-party schematic entry formats
  - Macro library
  - Versatile netlist format for design portability
  - Logic, timing, and fault simulation
- SLICE development system:
  - Easy to learn and use
  - State or Boolean equation entry
  - Fuse table editor
  - Test vector editor
  - Boolean equation extractor
  - JEDEC fusemap compiler
  - Upgradeable to SNAP
- TTL compatible
- Security fuse allows protection of proprietary designs
- Testable in unprogrammed state

STRUCTURE

- NAND gate based architecture
  - 64 foldback NAND terms
- 80 additional logic terms
- 160 inputs per logic term
- 20 dedicated inputs
- 4 programmable input/clock inputs
- 8 independent clocks
  - 4 from input/clock pins
  - 4 from NAND array
- 8 bidirectional I/Os
- 16 dedicated outputs
  - 8 Active-High outputs
  - 4 outputs with programmable polarity
  - 4 3-State outputs with programmable polarity and independent 3-State control
- 16 buried flip-flops
  - 8 D type
  - 8 S-R type
- Power-on preset ot logic '1' of all flip-flops

DESCRIPTION

The Signetics PML family of PLDs provides the capability to create fast and cost effective solutions for a number of microprocessor interface and control applications. PML incorporates the unique feature of a programmable NAND structure as the basis of its architecture.

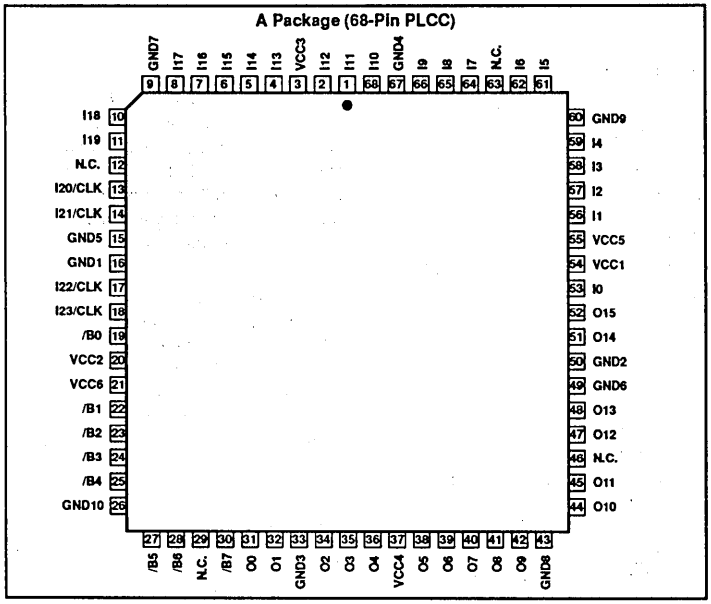
The PLHS502 is a high-density Bipolar Programmable Macro Logic Device. The folded NAND array combined with embedded I/O flip-flops allows for both timing control, wide decoders, multiplexers, and system input and output bus latches to be combined onto one device.

APPLICATIONS

- VRAM controllers
- DRAM/SRAM controllers
- Multiple state machines
- Timing control
- Error detection/correction

Programmable macro logic | PLHS502

PIN CONFIGURATION



ARCHITECTURE

The core of the PLHS502 is a programmable fuse array of 64 NAND gates and 16 buried flip-flops. The output of each gate and flip-flop folds back upon itself and all other NAND gates and flip-flops. In this manner, full connectivity of all logic functions is achieved in the PLHS502. Any logic function can be created within the core of the device without wasting valuable I/O pins. Furthermore, a speed advantage is acquired by implementing multi-level logic within a fast internal core without incurring any delays from the I/O buffers.

Buried Flip-Flops

The 16 buried flip-flops can be connected to the input or output structures through the NAND array. Intricate state machine designs can be implemented within the core without any unnecessary delays from the input or output buffers. Each flip-flop can be realized as an input or output register with no constraints.

The Clock Array

There are a combination of 26 possible inputs to the 'Clock Array':

– 2 are directly from the input pins fed through an inverting buffer.
– 4 clock inputs with selectable polarity directly from 2 input pins.
– 4 inputs from 4 individual NAND terms.
– 16 inputs from the 'Q' outputs of the flip-flops.

The wide selection of clocking options offers the user the capacity to create custom and independent clock functions for the flip-flops. This together with the full connectivity of the device, offers the capability to implement a variety of synchronous and asynchronous state machines. Another possible application is implementing multi-phase designs such as pipe-lined processing.

DESIGN DEVELOPMENT TOOLS

SNAP

The SNAP Software Development System provides the necessary tools for designing with PML. SNAP provides the following:

- Schematic entry netlist generation from third-party schematic design packages such as OrCAD/SDT™ and FutureNet™.
- Macro library for standard TTL functions and user defined functions
- Boolean equation entry
- State equation entry
- Syntax and design entry checking
- Simulator includes logic simulation, fault simulation, and timing simulation.

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. A minimum of 640K bytes of RAM is required together with a hard disk.

SNAP provides primitive PML function libraries for third party schematic design packages. Custom macro function libraries can be defined in schematic or equation form.

After the completion of a design, the software compiles the design for syntax and completeness. Complete simulation can be carried out using the different simulation tools available.

The programming data is generated in JEDEC format. Using the Device Programmer Interface (DPI) module of SNAP, the JEDEC fusemap is sent from the host computer to the device programmer.

SLICE

SLICE, which supports Signetics PLD line, is easy to understand and simple to use. Select a PLD, assign input and output pins and enter the desired equations in either Boolean or state form. SLICE then checks the equations for errors. It automatically generates a JEDEC-format fuse map for downloading to a PLD programmer.

Fully menu driven, SLICE incorporates a fuse table editor for making quick modifications to the design and a test vector editor for input of test vectors.

A built-in Boolean equation extractor allows existing PLDs to be used as the basis for a new design. the extractor reads JEDEC information from a PLD and creates a file containing the corresponding Boolean equations. The result can then be used to consolidate several PLD designs into a single, denser part.

And SLICE is upward compatible with Signetics extensive design suite, SNAP.

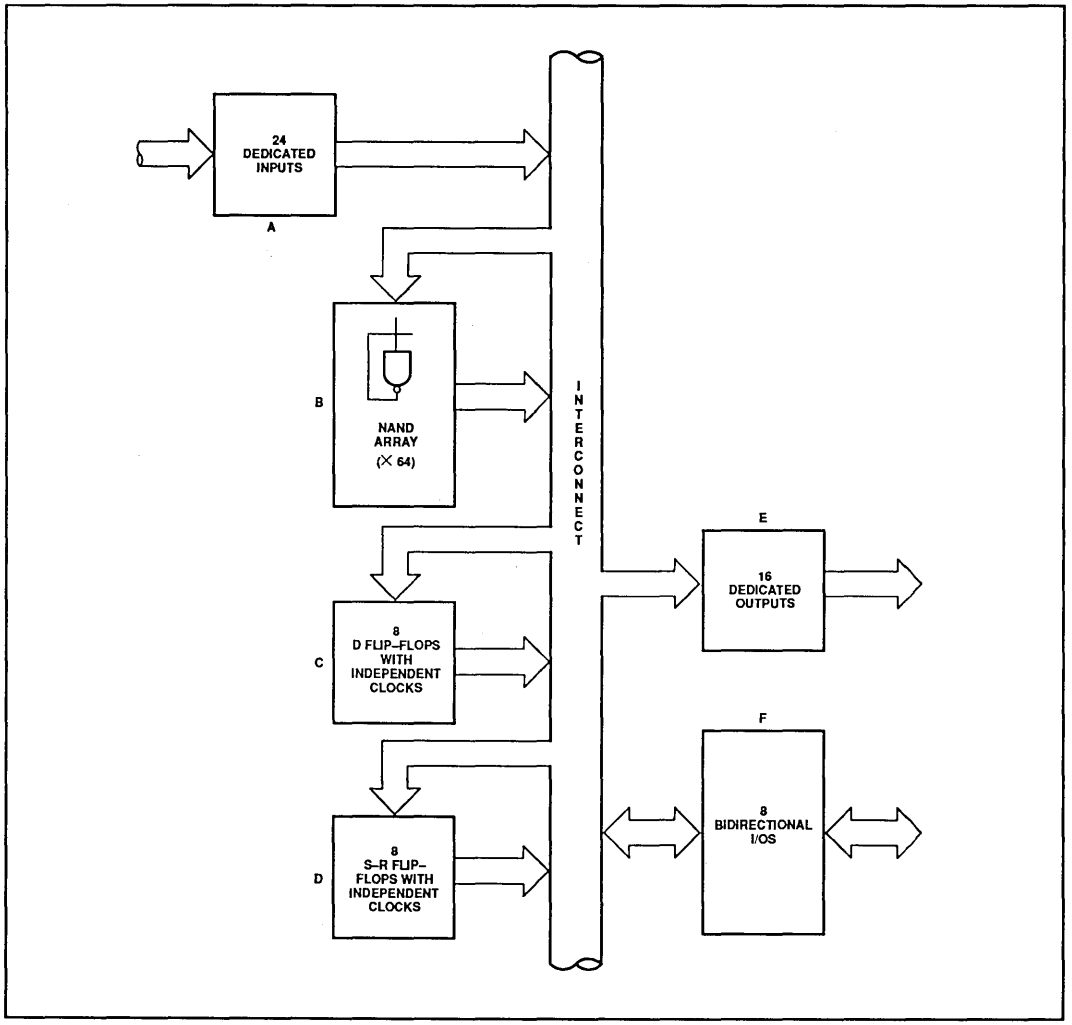
DESIGN SECURITY

The PLHS502 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

FutureNet is a trademark of FutureNet Corporation.
OrCAD/SDT is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

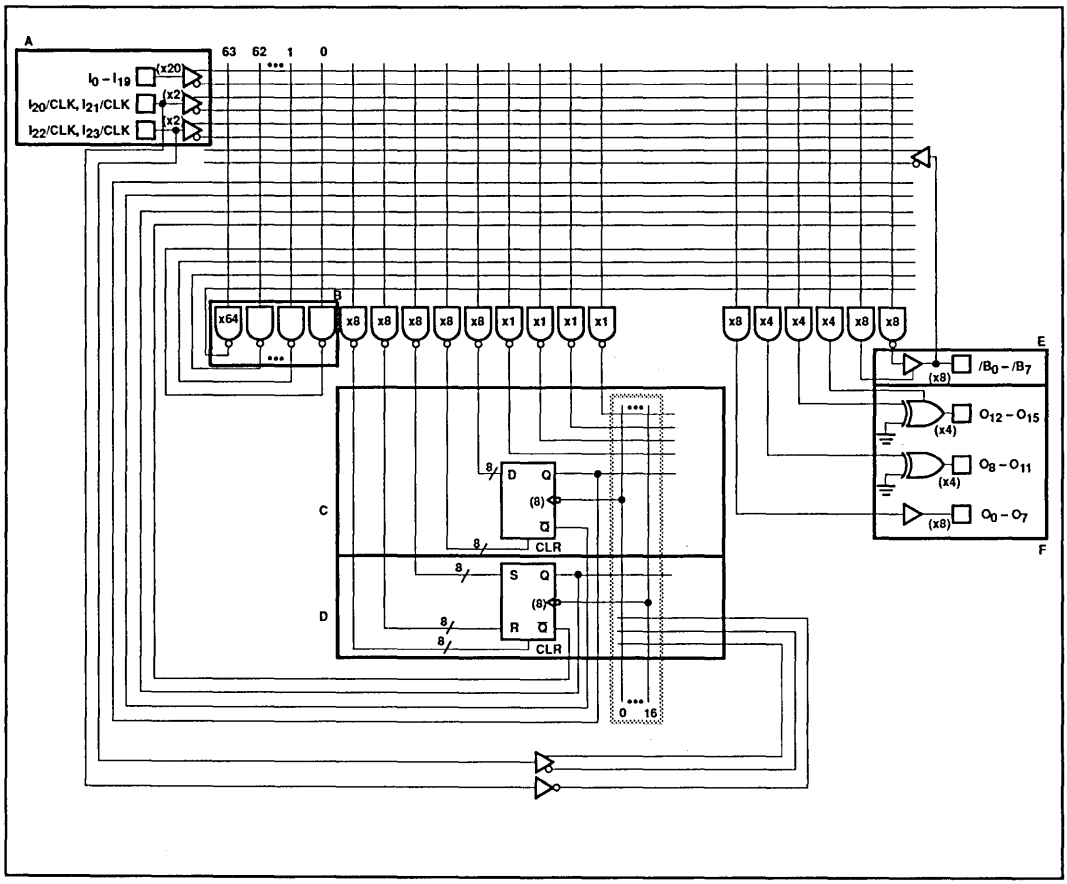
Programmable macro logic PLHS502

PLHS502 FUNCTIONAL BLOCK DIAGRAM

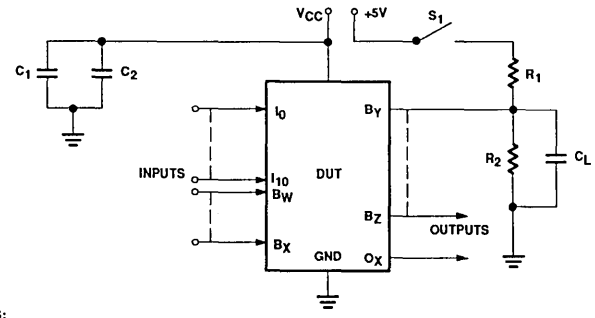


Programmable macro logic PLHS502

FUNCTIONAL DIAGRAM

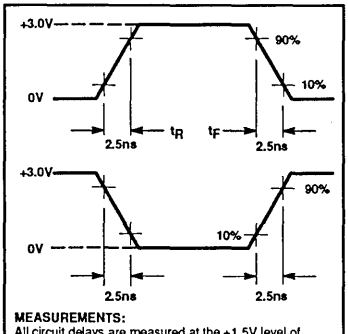


TEST LOAD CIRCUITS



NOTES:
C_1 and C_2 are to bypass V_{CC} to GND.
T_{amb} = 0°C to +75°C, 4.75 ≤ V_{CC} ≤ 5.25V, R_1 = 470Ω, R_2 = 1000Ω, C_L = 30pF

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

Programmable macro logic — PLHS502

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
68-Pin Plastic Leaded Chip Carrier	PLHS502A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All bidirectional (B) pins are outputs.
2. All outputs are enabled.
3. All outputs are Active-Low **except** $O_0 - O_7$, which are Active-High.

DC ELECTRICAL CHARACTERISTICS

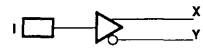
$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[2]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage						
		V_{CC} = MIN				
V_{OL}	Low[2]	I_{OL} = 10mA			0.45	V
V_{OH}	High[2]	I_{OH} = –2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = 5.5V			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = 5.5V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX			400	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.

MACRO CELL A.C. SPECIFICATIONS (SNAP Resource Summary Designations in Parentheses)

Input Buffer (DIN502, CDIN502, BDIN502 NIN502, CNIN502, BNIN502)

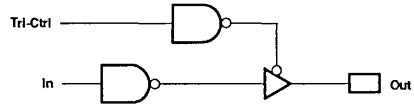


SYMBOL	MIN	TYP	MAX	UNIT
Δt_{HL}	0.05	0.1	0.15	ns/p-term
Δt_{LH}	–0.02	–0.05	–0.08	ns/p-term

SYMBOL	PARAMETER		LIMITS			UNIT	NOTES
	To (Output)	From (Input)	MIN	TYP	MAX		
t_{HL} t_{LH}	X X	I I	4 4.5	5 5.5	6 6.5	ns ns	With 0 p-term load
t_{HL} t_{LH}	Y Y	I I	2 3.5	3 3.5	3.5 4	ns ns	With 0 p-term load

Input Pins: 1, 2, 4–8, 10, 11, 13, 14, 17, 18, 53, 56–59, 61, 62, 64–66, 68.
Bidirectional Pins: 19, 22–25, 27, 28, 30.
Limit of 16 NAND terms for Input Buffer (X and Y) and Internal Foldback NAND (Y).

NAND Output Buffer with 3-State Control (TOU502)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL} t_{PLH}	Out Out	In In	8.5 9	9 10	13.5 14	ns ns
t_{OE}[1] t_{OD}[1]	Out Out	Tri-Ctrl Tri-Ctrl	10 8	12 10	18.5 16	ns ns

Bidirectional and Output Pins: 19, 22–25, 27, 28, 30.

Internal Foldback NAND (FBNAND)

SYMBOL	MIN	TYP	MAX	UNIT
Δt_{PHL}	0.05	0.1	0.15	ns/p-term
Δt_{PLH}	–0.02	–0.05	–0.08	ns/p-term

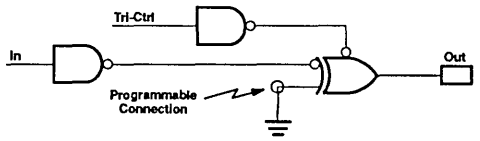
SYMBOL	PARAMETER		LIMITS			UNIT	NOTES
	To (Output)	From (Input)	MIN	TYP	MAX		
t_{PHL}	Y	X	4.0	5.0	5.5	ns	With 0 p-term load
t_{PLH}	Y	X	5.5	6.5	8	ns	With 0 p-term load

Limit of 16 NAND terms for Input Buffer (X and Y) and Internal Foldback NAND (Y).

Note on following page.

MACRO CELL A.C. SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

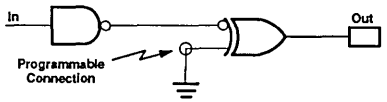
3-State Output with Programmable Polarity (TEXO502)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	9	10	19	ns
t_{PLH}	Out	In	10	12	19	ns
t_{OE}[1]	Out	Tri–Ctrl	10	12	18.5	ns
t_{OD}[1]	Out	Tri–Ctrl	8	10	16	ns

Output Pins: 47, 48, 51, 52.

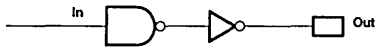
Output with Programmable Polarity (EXO502)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	9	10	19	ns
t_{PLH}	Out	In	10	12	19	ns

Output Pins: 41, 42, 44, 45.

Output Buffer (NOU502)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	8	9	14	ns
t_{PLH}	Out	In	9	10	13.5	ns

Output Pins: 31, 32, 34–36, 38–40.

NOTE:

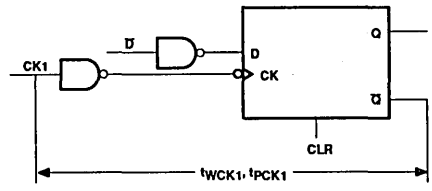
1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

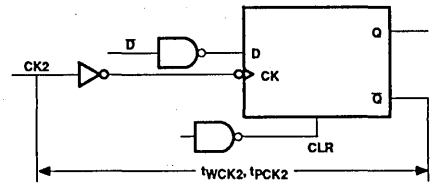
D FLIP-FLOP (SNAP Resource Summary Designation = DFF502)

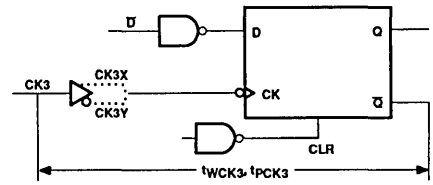
INPUTS			OUTPUTS	
CLR	CK	$\overline{D}$	Q	$\overline{Q}$
H	X	X	L	H
L	L	X	Q_0	$\overline{Q}_0$
L	↑	H	L	H
L	↑	L	H	L

NOTE:
Q_0, $\overline{Q}_0$ represent previous stable condition of Q, $\overline{Q}$.

SYMBOL	LIMITS			UNIT
	MIN	TYP	MAX	
f_{CK1}	33	37	40	MHz
f_{CK2}	37	40	43.5	MHz
f_{CK3X}	33	37	40	MHz
f_{CK3Y}	37	40	43.5	MHz
$t_{W\,CK1}$	15	10	8	ns
$t_{P\,CK1}$	30	27	25	ns
$t_{W\,CK2}$, $t_{W\,CK3Y}$	10	9	8	ns
$t_{P\,CK2}$, $t_{P\,CK3Y}$	27	25	23	ns
$t_{W\,CK3X}$	10	9	8	ns
$t_{P\,CK3X}$	30	27	25	ns
t_{SETUP} $\overline{D}$	7	5.5	3	ns
t_{HOLD} $\overline{D}$	8.5	4.5	1	ns
$t_{W\,CLR}$ High	10	9	8	ns







SYMBOL	LIMITS			NOTES
	MIN	TYP	MAX	
Δt_{SETUP}	–0.02	–0.05	–0.1	ns/FO of CK1,2,3

SYMBOL	FROM (INPUT)	TO (OUTPUT)	LIMITS			UNIT
			MIN	TYP	MAX	
t_{PLH}	CK1	Q, $\overline{Q}$	16.5	20	25	ns
t_{PHL}	CK1	Q, $\overline{Q}$	17.5	20.5	24.5	ns
t_{PLH}	CK2	Q, $\overline{Q}$	12	14	16.5	ns
t_{PHL}	CK2	Q, $\overline{Q}$	13	15	16	ns
t_{PLH}	CK3X	Q, $\overline{Q}$	14	16	19.5	ns
t_{PHL}	CK3X	Q, $\overline{Q}$	15	16	19	ns
t_{PLH}	CK3Y	Q, $\overline{Q}$	12	14	16.5	ns
t_{PHL}	CK3Y	Q, $\overline{Q}$	13	15	16	ns
t_{PLH}	CLR	Q, $\overline{Q}$	11	15	20	ns
t_{PHL}	CLR	Q, $\overline{Q}$	12	15	19.5	ns

NOTES:

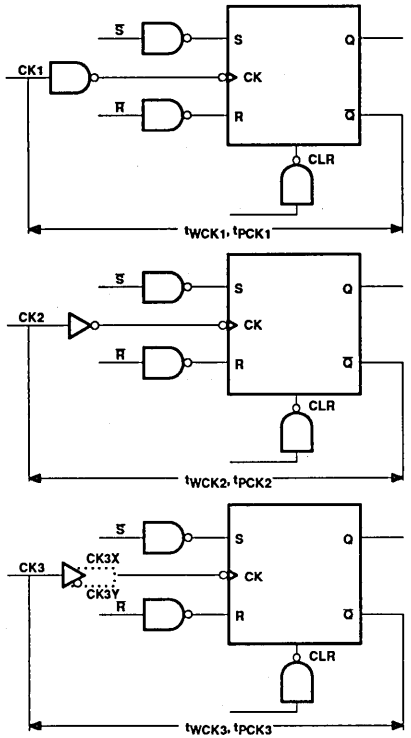
1. Setup and Hold times are with reference to rising edge of CK1, CK2, and CK3.
2. Limit of 16 Logic terms load on Q and $\overline{Q}$.
3. t_P = period; t_W = positive pulse width.

SYMBOL	LIMITS			NOTES
	MIN	TYP	MAX	
Δt_{PHLK}	0.05	0.1	0.15	ns/FO of CK1,2,3
Δt_{PLHK}	0.05	0.1	0.15	ns/FO of CK1,2,3
Δt_{PLHQ}	–0.02	–0.5	–0.08	ns/p-term load on Q, $\overline{Q}$
Δt_{PHLQ}	0.05	0.1	0.15	ns/p-term load on Q, $\overline{Q}$

S-R FLIP-FLOP (SNAP Resource Summary Designation = RSF502)

INPUTS				OUTPUTS	
CLR	CK	$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
H	X	X	X	L	H
L	L	X	X	Q_0	$\overline{Q}_0$
L	↑	H	L	L	H
L	↑	L	H	H	L
L	↑	H	H	Q_0	$\overline{Q}_0$
L	↑	L	L	Not allowed	

NOTE:
Q_0, $\overline{Q}_0$ represent previous stable condition of Q, $\overline{Q}$.



SYMBOL	LIMITS MIN	TYP	MAX	UNIT
f_{CK1}	33	37	40	MHz
f_{CK2}	37	40	43.5	MHz
f_{CK3X}	33	37	40	MHz
f_{CK3Y}	37	40	43.5	MHz
$t_{W\,CK1}$	15	10	8	ns
$t_{P\,CK1}$	30	27	25	ns
$t_{W\,CK2}$	10	9	8	ns
$t_{P\,CK2}$	27	25	23	ns
$t_{W\,CK3X}$	10	9	8	ns
$t_{P\,CK3X}$	30	27	25	ns
$t_{W\,CK3Y}$	10	9	8	ns
$t_{P\,CK3Y}$	27	25	23	ns
t_{SETUP} $\overline{S}$, $\overline{R}$	7	5.5	3	ns
t_{HOLD} $\overline{S}$, $\overline{R}$	8.5	4.5	1	ns
$t_{W\,CLR}$ High	10	9	8	ns

SYMBOL	LIMITS MIN	TYP	MAX	NOTES
Δt_{SETUP}	–0.02	–0.05	–0.1	ns/FO of CK1,2,3

SYMBOL	FROM (INPUT)	TO (OUTPUT)	LIMITS MIN	TYP	MAX	UNIT
t_{PLH}	CK1	Q, $\overline{Q}$	16.5	20	25	ns
t_{PHL}	CK1	Q, $\overline{Q}$	17.5	20.5	24.5	ns
t_{PLH}	CK2	Q, $\overline{Q}$	12	14	16.5	ns
t_{PHL}	CK2	Q, $\overline{Q}$	13	15	16	ns
t_{PLH}	CK3X	Q, $\overline{Q}$	14	16	19.5	ns
t_{PHL}	CK3X	Q, $\overline{Q}$	15	17	19	ns
t_{PLH}	CK3Y	Q, $\overline{Q}$	12	14	16.5	ns
t_{PHL}	CK3Y	Q, $\overline{Q}$	13	15	16	ns
t_{PLH}	CLR	$\overline{Q}$	11	15	20	ns
t_{PHL}	CLR	Q	12	15	19.5	ns

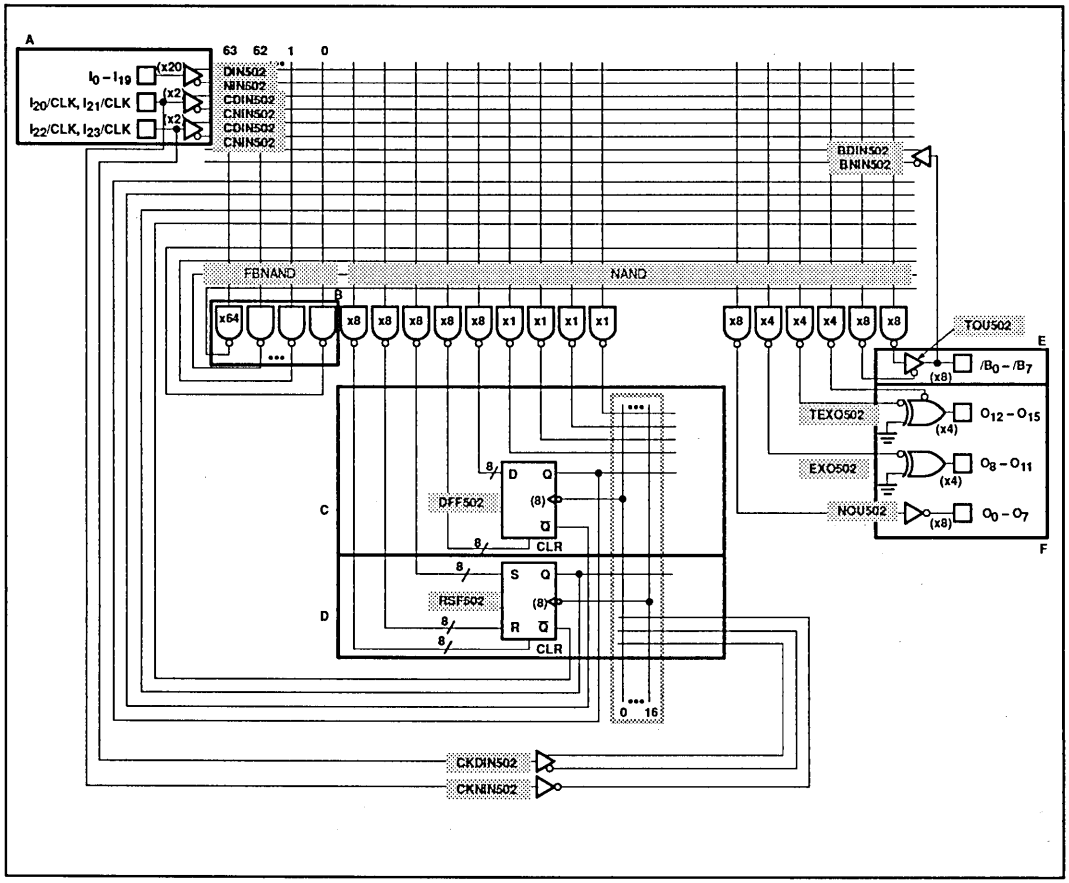
NOTES:

1. Setup and Hold times are with reference to rising edge of CK1, CK2, and CK3.
2. Limit of 16 Logic terms load on Q and $\overline{Q}$.
3. t_P = period; t_W = positive pulse width.

SYMBOL	LIMITS MIN	TYP	MAX	NOTES
Δt_{PHLK}	0.05	0.1	0.15	ns/FO of CK1,2,3
Δt_{PLHK}	0.05	0.1	0.15	ns/FO of CK1,2,3
Δt_{PLHQ}	–0.02	–0.5	–0.08	ns/p-term load on Q, $\overline{Q}$
Δt_{PHLQ}	0.05	0.1	0.15	ns/p-term load on Q, $\overline{Q}$

Programmable macro logic PLHS502

SNAP RESOURCE SUMMARY DESIGNATIONS



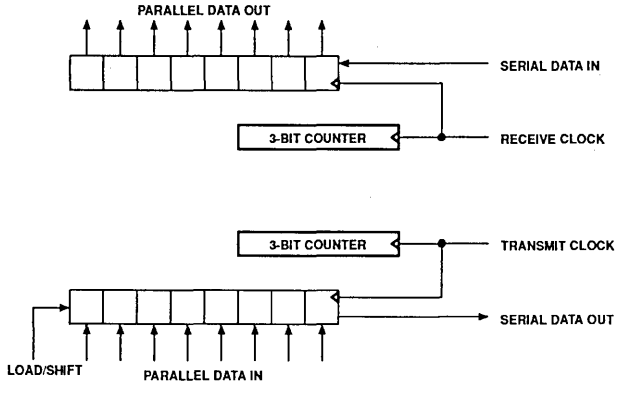
Programmable macro logic | PLHS502

PLHS502 GATE AND SPEED ESTIMATE TABLE

FUNCTION	INTERNAL NAND EQUVALENT	TYPICAL t_{PD}	f_{MAX}	COMMENTS
Gates				
NANDs	1	6.5ns		For 1 to 32-pin input variables
ANDs	1	6.5ns		Additional internal inputs can be used as needed
NORs	1	6.5ns		Additional internal inputs can be used as needed
ORs	1	6.5ns		Additional internal inputs can be used as needed
Macro Flip-Flops				
D-Type Flip-Flop	N/A		40MHz	Total budget = 8
SR-Type Flip-Flop	N/A		40MHz	Total budget = 8
Gate Implemented Flip-Flops				
D-Type Flip-Flop	6		35MHz	With asynchronous S-R
T-Type Flip-Flop	6		35MHz	With asynchronous S-R
J-K-Type Flip-Flop	10		35MHz	With asynchronous S-R
Transparent-D Latch	4		N/A	With asynchronous S-R
S-R Latch	2		N/A	With asynchronous S-R
Decoders				
3-to-8	8	11.5ns		Inverted inputs available
4-to-16	16	11.5ns		Inverted inputs available
5-to-32	32	11.5ns		Inverted inputs available (24 chip outputs only)
Encoders				
8-to-3	15	11.5ns		Inverted inputs, 2 logic levels
16-to-4	32	11.5ns		Inverted inputs, 2 logic levels
32-to-5	41	11.5ns		Inverted inputs, 2 logic levels
Multiplexers				
4-to-1	5	11.5ns		Inverted inputs available
8-to-1	9	11.5ns		Inverted inputs available
16-to-1	17	11.5ns		Inverted inputs available
27-to-1	28	11.5ns		Can address only 27 external inputs - more if internal only. This disallows clock inputs to flip-flop.

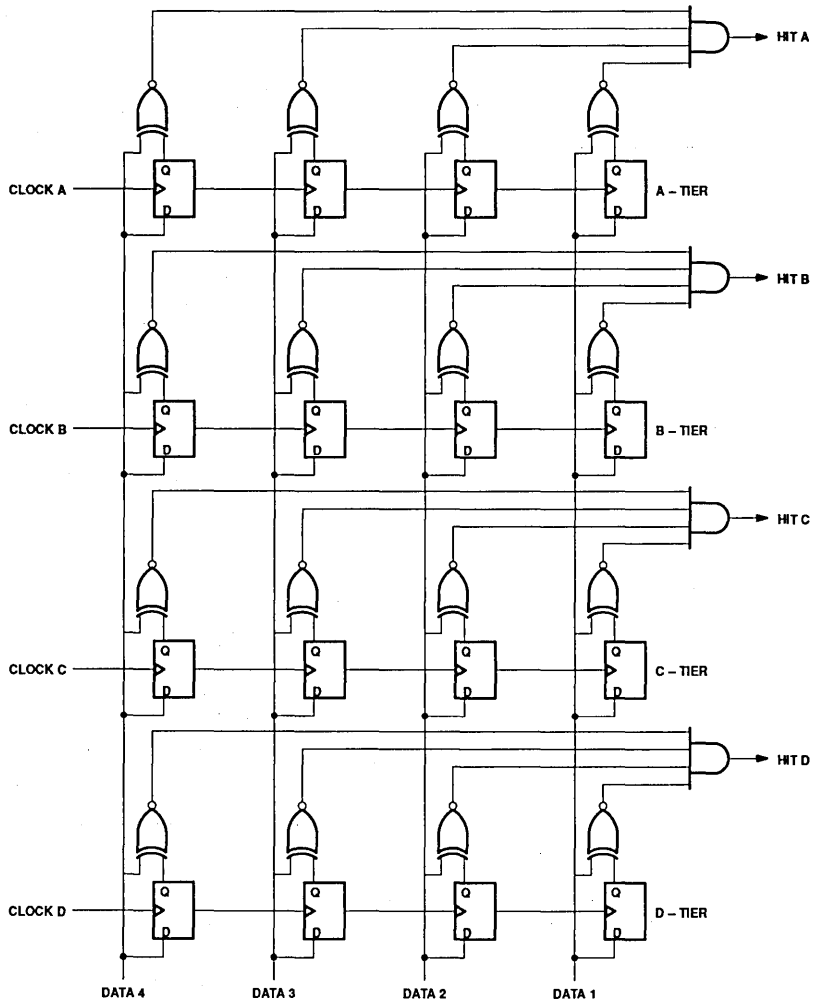
PLHS502 Rough Resource Budget = 64 NANDs, 8 D, 8 SR, 24 inputs, 16 outputs, 8 bidirectionals.

APPLICATIONS



40MHz Synchronous Receiver/Transmitter

APPLICATIONS (Continued)



4 × 4 Content Addressable Memory (Address valid to Hit Time = 20ns)

Philips Components–Signetics

Document No.	853–1476
ECN No.	00483
Date of Issue	September 21, 1990
Status	Product Specification
Programmable Logic Devices	

PLHS601
Programmable macro logic
PML

FEATURES
- Delay per internal NAND function = 4.5ns (typ)
- SNAP development system
  - Supports third-party schematic entry formats
  - TTL macro library
  - Versatile netlist format for design portability
  - Logic, timing, and fault simulation
  - Test vector generation
- SLICE development system:
  - Easy to learn and use
  - State or Boolean equation entry
  - Fuse table editor
  - Test vector editor
  - Boolean equation extractor
  - JEDEC fusemap compiler
  - Upgradeable to SNAP
- Full connectivity, no place and route restrictions
- Security fuse for copy protection
- 68-pin PLCC package

STRUCTURE
- 86 foldback terms
  - 78 foldback NAND terms
  - 8 foldback Exclusive-OR terms
- 166 input wide logic terms
- 64 additional logic terms
- 28 dedicated inputs
- 12 bidirectional pins with Active-High output and independent 3-State control
- 12 dedicated Active-High outputs with independent 3-state control

DESCRIPTION
The Philips Components–Signetics Programmable Macro Logic (PML) family of PLDs provides the capability to create fast and cost effective solutions for general purpose logic integration, microprocessor bus interface and control applications. PML incorporates wide-input NAND gates as the core of its architecture. With this architecture, multiple levels of logic can be realized within the device without wasting valuable I/O pins. Any combinatorial logic function can be implemented with no more than two gate levels. The un-committed NAND gates allow efficient implementation of flip-flops, latches, registers, and counters of any type, all with predictable delays. Furthermore, full connectivity among the different macros within the device eliminates the route and place restrictions associated with high density programmable gate arrays.

The PLHS601 is a high-density bipolar PML device. The PLHS601 is a powerful solution to eliminate wait states and create cost-effective microprocessor support circuitry.

APPLICATIONS
- Low-end gate array replacement
- High speed wide address decoders
- Wide multiplexers and decoders
- Bus arbitration functions
- General purpose logic integration and microprocessor support logic
- PAL® and glue logic replacement

ARCHITECTURE
The core of the PLHS601 is a programmable fuse array of 78 folded NAND gates and 8 folded Exclusive-OR gates. The output of each gate folds back upon itself and all other gates. In this manner, full connectivity of all logic functions is achieved. Any logic functions can be implemented within the core of the device without wasting valuable I/O pins. Furthermore, a speed advantage is achieved by creating multiple levels of logic within the folded core without incurring any delays from the I/O buffers.

Programmable macro logic PLHS601

DESIGN DEVELOPMENT TOOLS

SNAP

The SNAP Software Development System provides the necessary tools for designing with PML. SNAP provides the following:

- Schematic entry netlist generation from third-party schematic design packages such as OrCAD/SDT III™ and FutureNet™.
- Macro library for standard TTL functions and user defined functions
- Boolean equation entry
- State equation entry
- Syntax and design entry checking
- Simulator includes logic simulation, fault simulation, automatic test vector generation, and timing simulation.
- Resource summary

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. The minimum system configuration for SNAP is 640K bytes of RAM and a hard disk.

SNAP provides primitive PML function libraries for third-party schematic design packages. Custom macro function libraries can be defined in schematic or equation form.

After the completion of a design, the software compiles the design for syntax and completeness. Complete simulation can be carried out using the different simulation tools available.

The programming data is generated in JEDEC format. Using the Device Programmer Interface (DPI) module of SNAP, the JEDEC fusemap is sent from the host computer to the device programmer.

SLICE

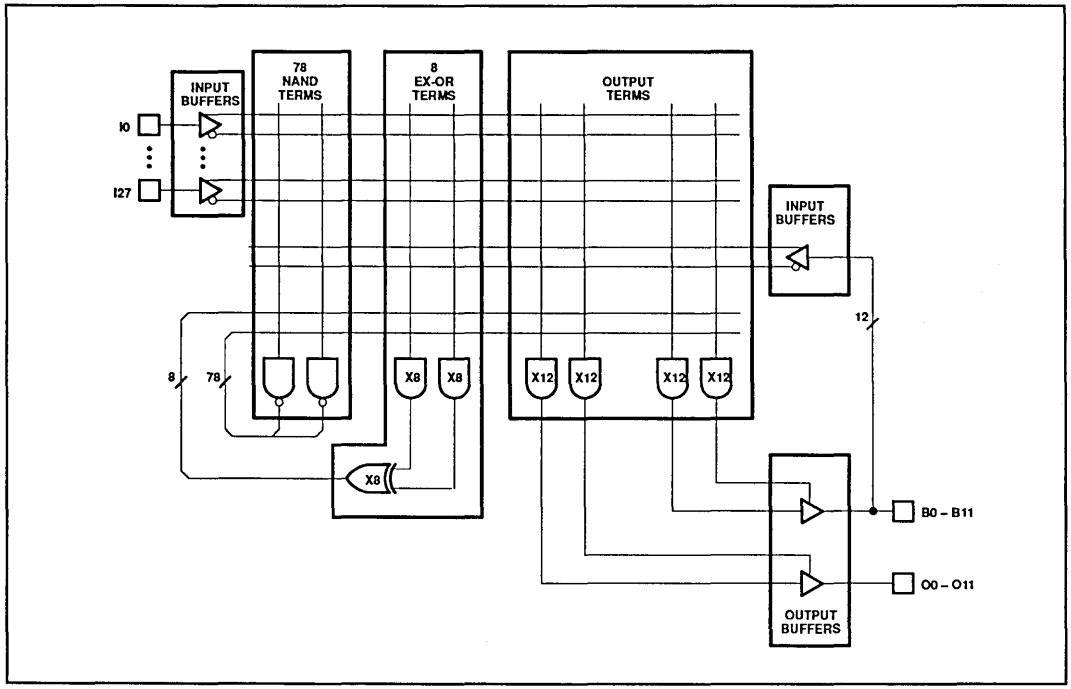
SLICE, which supports Signetics PLD line, is easy to understand and simple to use. Select a PLD, assign input and output pins and enter the desired equations in either Boolean or state form. SLICE then checks the equations for errors. It automatically generates a JEDEC-format fuse map for downloading to a PLD programmer.

Fully menu driven, SLICE incorporates a fuse table editor for making quick modifications to the design and a test vector editor for input of test vectors.

A built-in Boolean equation extractor allows existing PLDs to be used as the basis for a new design. the extractor reads JEDEC information from a PLD and creates a file containing the corresponding Boolean equations. The result can then be used to consolidate several PLD designs into a single, denser part.

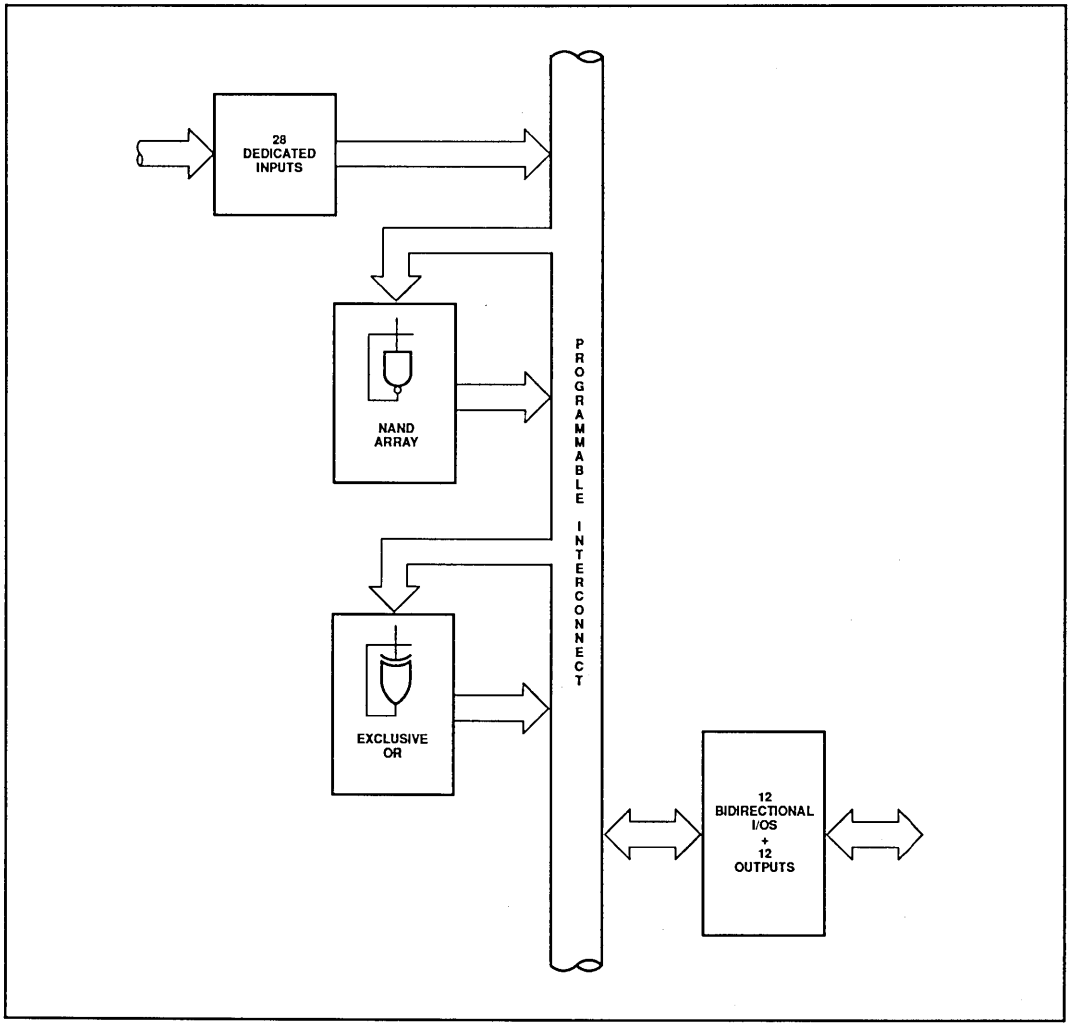
And SLICE is upward compatible with Signetics extensive design suite, SNAP.

FUNCTIONAL DIAGRAM



FutureNet is a trademark of FutureNet Corporation.
OrCAD/SDT is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

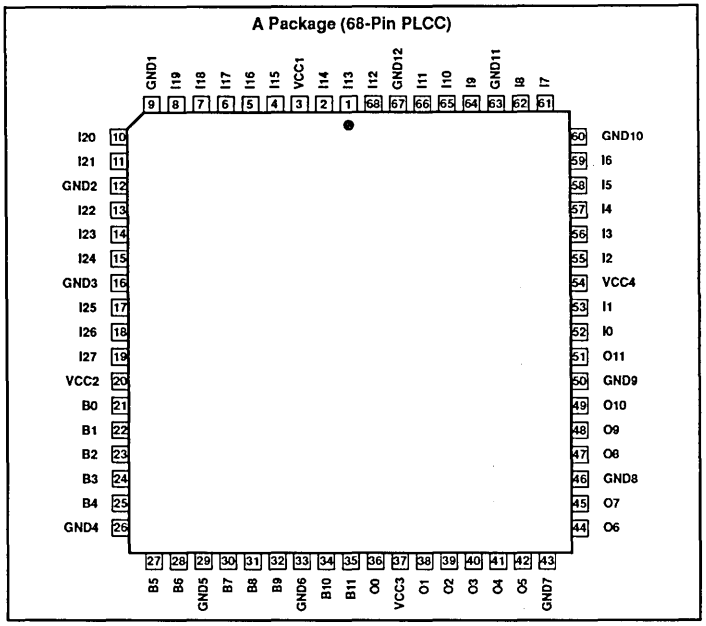
PLHS601 FUNCTIONAL BLOCK DIAGRAM



Programmable macro logic — PLHS601

PIN CONFIGURATION

A Package (68-Pin PLCC)



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
68-Pin Plastic Leaded Chip Carrier	PLHS601A

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All product terms are enabled.
2. All bidirectional (B) pins are outputs.
3. All outputs are enabled.
4. All outputs are Active-HIGH.

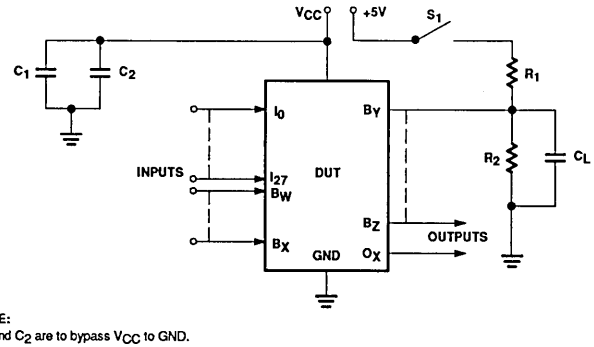
DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[2, 3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage						
		V_{CC} = MIN				
V_{OL}	Low[2]	I_{OL} = 10mA			0.45	V
V_{OH}	High[2]	I_{OH} = −2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			−100	μA
I_{IH}	High	V_{IN} = 5.5V			40	μA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[6]	V_{OUT} = 5.5V			80	μA
		V_{OUT} = 0.45V			−140	
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX			340	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_{OUT} = 2.0V		15		pF

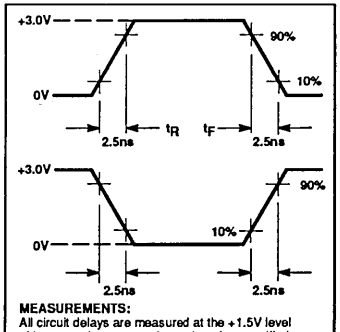
NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short circuit should not exceed 1 second.
5. I_{CC} is measured with all dedicated inputs at 0V and bidirectional and output pins open.
6. Leakage values are a combination of input and output leakages.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



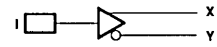
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

MACRO CELL SPECIFICATIONS

T_{amb} = 0°C to +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω
(SNAP Resource Summary Designations in Parentheses)

Input Buffer (DIN601, NIN601)

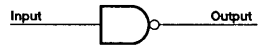


SYMBOL	LIMITS MIN	TYP	MAX	UNIT
Δt_{HL}	0.05	0.1	0.15	ns/p-term
Δt_{LH}	–0.02	–0.05	–0.08	ns/p-term

SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT	NOTES
t_{PHL} t_{PLH}	X X	I I	4 4.5	5 5.5	7 7.5	ns ns	With 0 p-terms load
t_{PHL} t_{PLH}	Y Y	I I	2 3.5	3 3.5	3.5 4	ns ns	With 0 p-terms load

Input Pins: 1 – 2, 4 – 8, 10 – 11, 13 – 15, 17 – 19, 52 – 53, 55 – 59, 61 – 62, 64 – 66, 68.
Bidirectional Pins: 21 – 25, 27 – 28, 30 – 32, 34 – 35.
Maximum internal fan-out: 16 p-terms on X or Y.

Internal Foldback NAND (FBNAND)



SYMBOL	LIMITS MIN	TYP	MAX	UNIT
Δt_{PHL}	0.05	0.07	0.1	ns/p-term
Δt_{PLH}	–0.02	–0.05	–0.08	ns/p-term

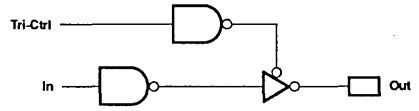
SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT	NOTES
t_{PHL} t_{PLH}	Out	Any	2.0 3.7	3.7 4.5	4.4 6	ns ns	With 0 p-terms load

Maximum internal loading of 16 terms.

MACRO CELL SPECIFICATIONS (Continued)

T_{amb} = 0°C to +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω
(SNAP Resource Summary Designations in Parentheses)

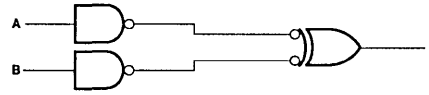
Non-Inverting Output Buffer with 3-State Control (NOU601)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Output	In	8	9	11.5	ns
t_{PLH}	Output	In	9	10	13.5	ns
t_{OE}[1]	Out	Tri-Ctrl	10	11.5	13.5	ns
t_{OD}[1]	Out	Tri-Ctrl	8	9.5	11.5	ns

Bidirectional and Output Pins: 19, 21, 22, 23, 15 – 18.

Internal Ex-OR Feedback Terms (EXO601)



SYMBOL	LIMITS			UNIT
	MIN	TYP	MAX	
Δt_{PHL}	0.05	0.07	0.1	ns/p-term
Δt_{PLH}	–0.02	–0.05	–0.08	ns/p-term

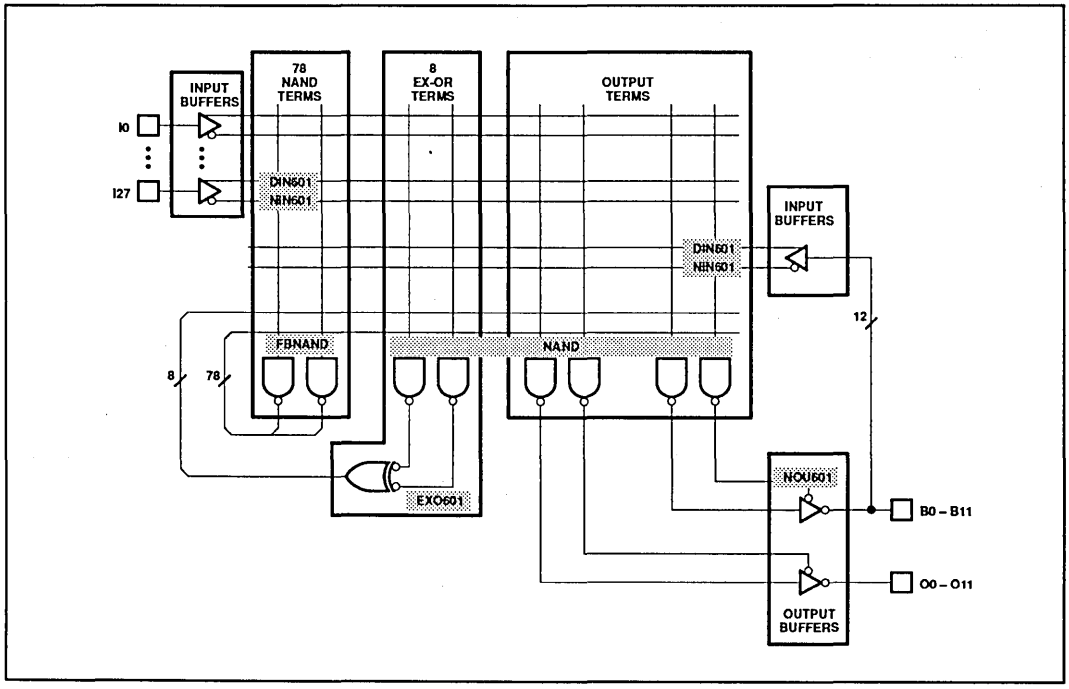
SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	A or B	4	5	6.6	ns
t_{PLH}	Out	A or B	5.5	6.5	8	ns

NOTE:

1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

Programmable macro logic PLHS601

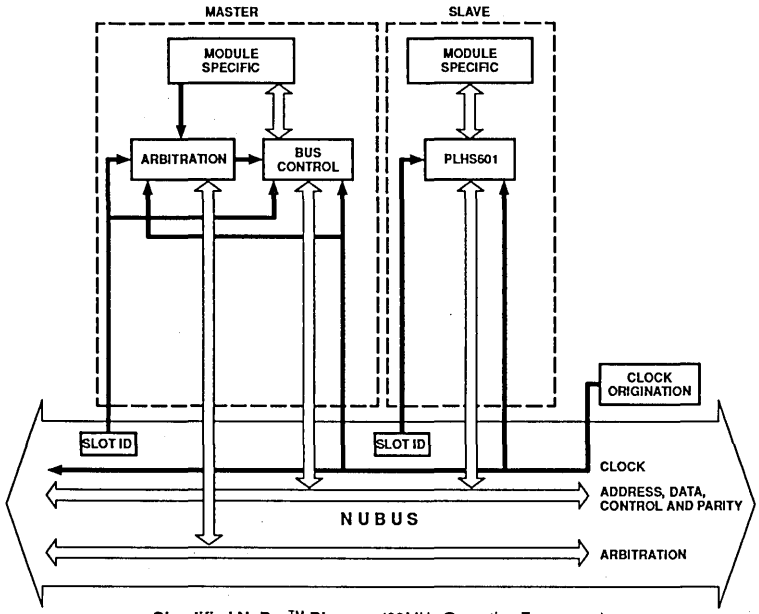
SNAP RESOURCE SUMMARY DESIGNATIONS



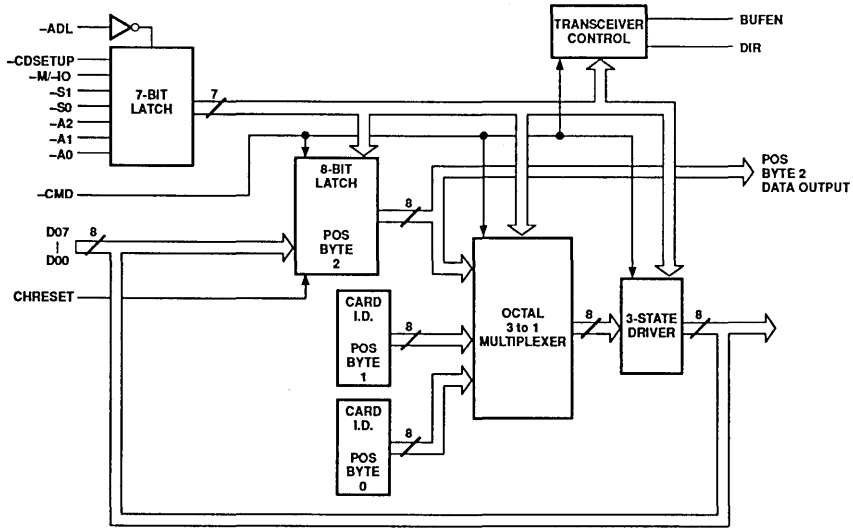
PLHS601 GATE AND SPEED ESTIMATE TABLE

FUNCTION	INTERNAL NAND EQUVALENT	TYPICAL t_{PD}	f_{MAX}	COMMENTS
Gates				
NANDs	1	4.5ns		For 1 to 40 input variables
ANDs	1	4.5ns		For 1 to 40 input variables
NORs	1	4.5ns		For 1 to 40 input variables
ORs	1	4.5ns		For 1 to 40 input variables
Decoders				
3-to-8	8	8.2ns		Inverted inputs available
4-to-16	16	8.2ns		Inverted inputs available
5-to-32	32	8.2ns		Inverted inputs available (24 chip outputs only)
Encoders				
8-to-3	15	8.2ns		Inverted inputs, 2 logic levels
16-to-4	32	8.2ns		Inverted inputs, 2 logic levels
32-to-5	41	8.2ns		Inverted inputs, 2 logic levels, factored solution.
Multiplexers				
4-to-1	5	8.2ns		Inverted inputs available
8-to-1	9	8.2ns		
16-to-1	17	8.2ns		
32-to-1	33	8.2ns		
Flip–Flops				
D-type Flip-Flop	6		45MHz	With asynchronous S-R
T-type Flip-Flop	6		45MHz	With asynchronous S-R
J-K-type Flip-Flop	10		45MHz	With asynchronous S-R
Adders				
8-bit	45	11.7ns		Full carry-lookahead (four levels of logic)
Barrel Shifters				
8-bit	72	8.2ns		2 levels of logic
Latches				
D-latch	3			2 levels of logic with one shared gate

APPLICATIONS



Simplified NuBus™ Diagram (20MHz Operating Frequency)



Block Diagram of Basic POS Implementation in PLHS601

NuBus is a trademark of Texas Instruments, Inc.

Programmable macro logic PLHS601

PROGRAMMING

Refer to the following charts for qualified manufacturers of programmers and software tools:

PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	UNISITE 40/48 V3.1 Pinsite – V3.1	O1C0BB* (with adaptor) O1C0BC (with pinsite)
STREBOR DATA COMMUNICATIONS 1008 N. NOB HILL AMERICAN FORK, UT 84003 (801)756–3605	PLP–S1A Programmer MP68CC Adaptor	

* Needs a 40-pin DIP to 68-pin PLCC adaptor that is available from Emulation Technology.
Part Number: AS–68–40–01P–6

EMULATION TECHNOLOGY, INC.
2368B Walsh Avenue, Blvd. D
Santa Clara, California 95051
Telephone No. (408) 982–0660
Fax. No. (408) 982–0664

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
PHILIPS COMPONENTS–SIGNETICS 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SNAP SOFTWARE REV. 1.4 AND LATER

Document No.	853–1475
ECN No.	00481
Date of Issue	September 20, 1990
Status	Product Specification
Programmable Logic Devices	

PML2552
Programmable macro logic
PML

FEATURES
- Full connectivity
- Erasable version and one time programmable version available
- Scan test
- Power down mode
- Power on reset
- 100% testable
- SNAP development system
  - Supports third-party schematic entry formats
  - TTL Macro library
  - Versatile netlist format for design portability
  - Logic, timing, and fault simulation
- SLICE development system:
  - Easy to learn and use
  - State or Boolean equation entry
  - Fuse table editor
  - Test vector editor
  - Boolean equation extractor
  - JEDEC fusemap compiler
  - Upgradeable to SNAP
- Power dissipation (TTL) = 630mW
- Power dissipation (CMOS) = 525mW
- Power dissipation (Power-Down mode) = 52mW
- Security fuse for copy protection
- Reprogrammable

PROPAGATION DELAYS
- Delay per internal NAND gate = 12ns (typ)
- 50MHz flip-flop toggle rate

APPLICATIONS
- Low-end gate array replacement
- Instrumentation
- Bus arbitration functions
- Wide multiplexers and decoders
- Multiple independent state machines
- General purpose logic integration and microprocessor support logic
- PAL® and glue logic replacement

DESCRIPTION
The Philips Components–Signetics PML family of PLDs provides "instant gate array" capabilities for general purpose logic integration applications. The PML2552 is the first high density CMOS–PML product. Fabricated with the Philips Components–Signetics high-performance EPROM process, it is an ideal way to reduce NRE costs, inventory problems and quality concerns. The PML2552 incorporates the PML folded NAND array architecture which provides 100% connectivity to eliminate routing restrictions. What distinguishes the PML2552 from the "classic" PLD architectures is its flexibility and the potent flip-flop building blocks. The device utilizes a folded NAND architecture, which enables the designer to implement multiple levels of logic on a single chip. The PML2552 eliminates the NRE costs, risks, and hard to use design tools associated with semicustom and full custom approaches. It allows the system designer to manage reliable functionality, in less time and space plus a faster time to market. The PML2552 is ideal in todays instrumentation, industrial control, EISA, NuBus™, bus interface and dense state machine applications in conjunction with the state-of-the-art CMOS processors. It is capable of replacing large amounts of TTL, SSI and MSI logic and literally allows the designer to build a system on the chip.

The SNAP development software gives easy access to the density and flexibility of the PML2552 through a variety of design entry formats, including schematic, logic equations, and state equations in any combination.

PAL is a registered trademark of Advanced Micro Devices, Inc.
NuBus is a trademark of Texas Instruments, Inc.

Programmable macro logic PML2552

FUNCTIONAL BLOCK DIAGRAM

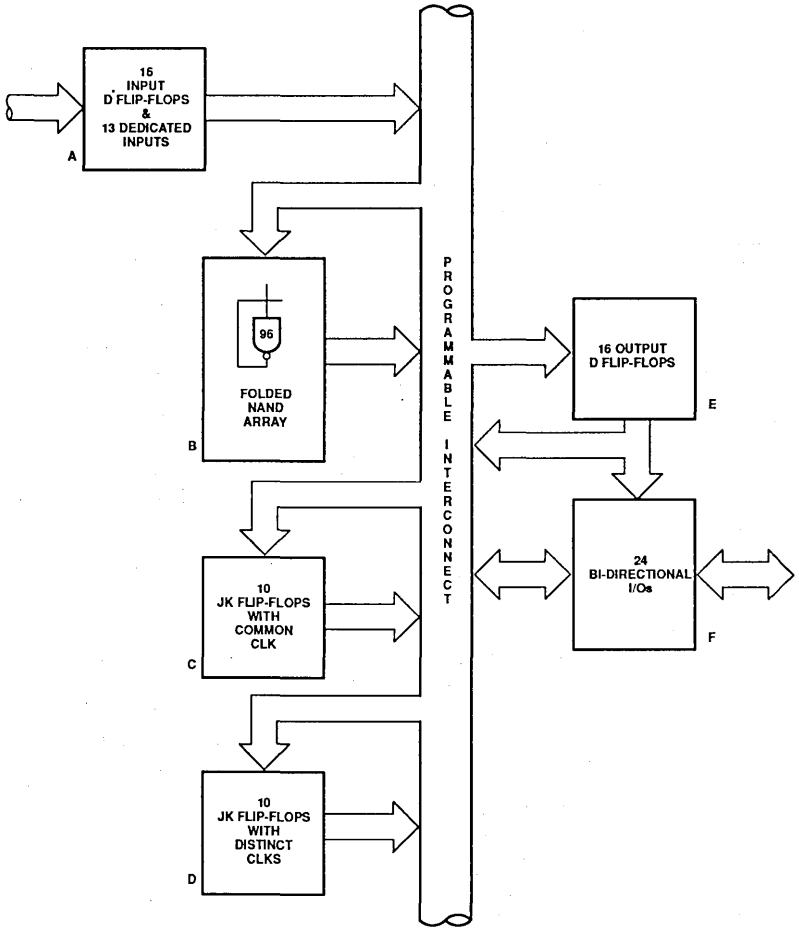
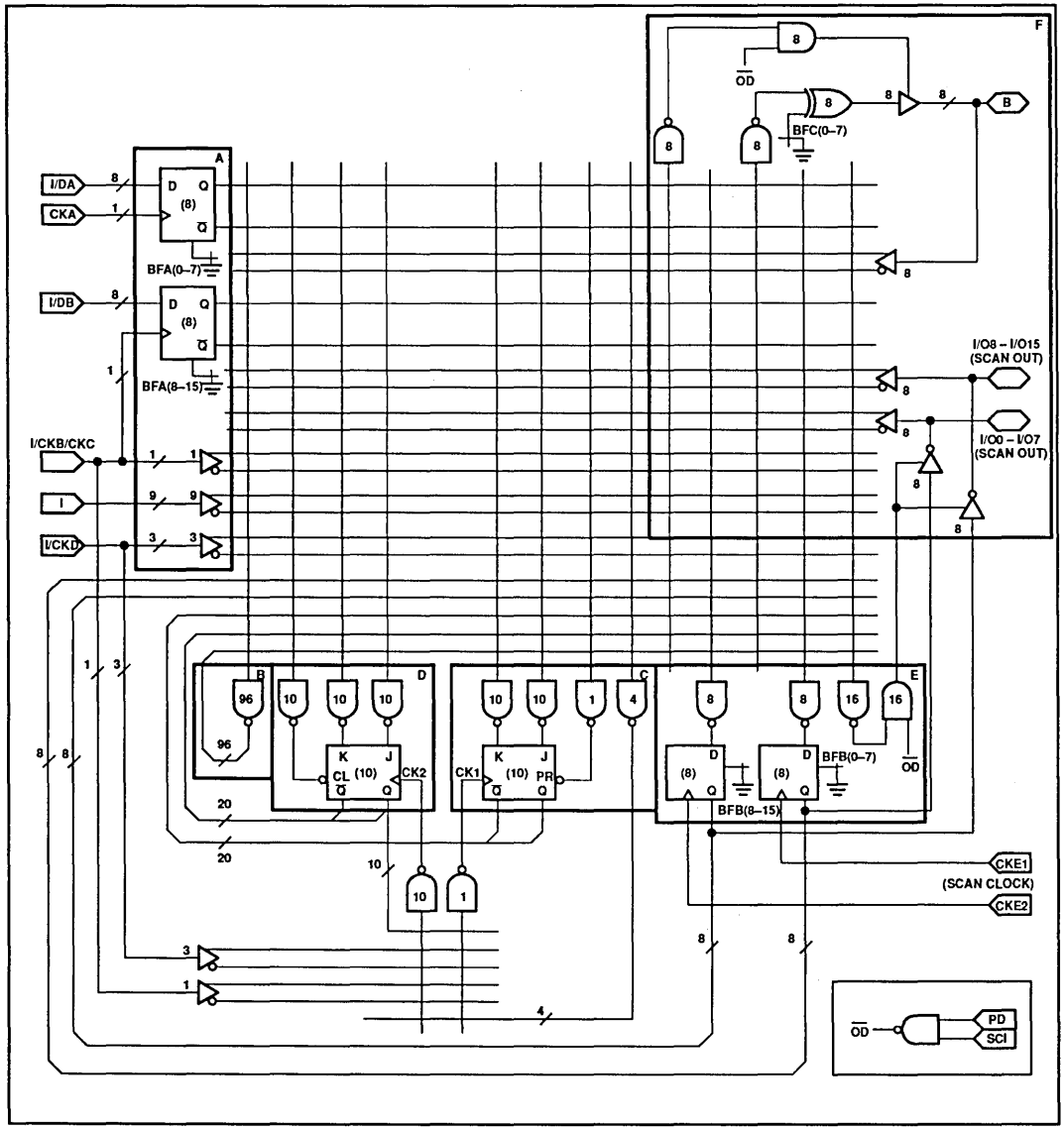


Figure 1

Programmable macro logic PML2552

LOGIC DIAGRAM



Programmable macro logic PML2552

STRUCTURE

- 112 possible foldback NAND gates:
  - 96 internal NAND
  - 16 from the I/O macros
- 114 additional logic terms
- 53 possible inputs (with programmable polarity)
  - 29 dedicated inputs
  - 24 bidirectional I/Os
- 24 bidirectional pins
- 52 flip-flops
- 24 possible outputs with individual Output Enable control (8 with programmable polarity)
- Multiple independent clocks
- 20 Buried JK-type flip-flops with foldback (JKFFs):
  - 10 JKFFs with one shared preset signal and one shared clocked signal originating from the clock array.
  - 10 JKFFs with 10 independent clock signals originating from the clock array and 10 independent clear signals
- 258 inputs per NAND gate
- Bypassable Input D-type flip-flop (DFFs)/Combinatorial Inputs:
  - 16 DFFs/combinatorial inputs
  - DFFs clocked in two groups of eight
  - DFFs not bypassed in unprogrammed state
  - Independent bypass fuse on each DFF
- Inputs/bypassable D-type flip-flop outputs/foldback NAND gates:
  - 16 output DFFs/combinatorial inputs/outputs with individual Output Enable control
  - DFFs clocked in two groups of eight
  - DFFs not bypassed in unprogrammed state
  - Independent bypass fuse on each DFF
  - The DFF can be used as an internal DFF or an internal foldback NAND gate.
- Combinatorial inputs:
  - 9 dedicated inputs to the NAND array
  - 3 inputs optional to NAND array and/or clock array
  - 1 input optional to NAND array and/or clock array, and/or clock of Input D Flip-Flops (Group B)
- Separate clock array:
  - Separate clock array for JKFFs clock inputs
  - 4 inputs to clock array originated from NAND array
  - 4 inputs (with programmable polarity) directly from input pins
  - 10 inputs from Q outputs of JKFFs with clear
- Dedicated clocks:
  - One dedicated clock for input DFFs (Group A)
  - Two dedicated clocks for output DFFs
- Scan test feature:
  - Scan chain is implemented through the 20 buried JKFFs and 16 output DFFs
  - Pins SCI, SCM, and CKE1 are used to operate the scan test
- Power down mode
  - Dedicated pin (PD) freezes the circuit when brought to logic "1". The circuit remains in the same state prior to the logic "0" to logic "1" transition of the "PD" pin.
  - When in the power down mode, the SCI pin acts as the 3-State pin for the 24 outputs.
- Power on reset:
  - All flip-flops (16 input DFFs, 20 buried JKFFs, and 16 output DFFs) are reset to logic "0" after V_{CC} power on.

ARCHITECTURE

The core of the PML2552 is a programmable NAND array of 96 NAND gates and 20 buried JKFFs. The output of each NAND gate folds back upon itself and all other NAND gates and flip-flops. The 'Q' and '$\overline{Q}$' output of each flip-flop also folds back in the same manner. Thus, total connectivity of all logic functions is achieved in the PML2552. Any logic function can be created within the core without wasting valuable I/O pins. Furthermore, a speed advantage is acquired by implementing multi-level logic within a fast internal core without incurring any delays from the I/O buffers. Figure 1 shows the functional block diagram of the PML2552.

Macro Cells

There are 16 bypassable DFFs on the input to the NAND array. These flip-flops are split in two banks of 8 (Bank A and Bank B). Each bank of flip-flops has a common clock. In the unprogrammed state of the device the flip-flops are active. In order to bypass any DFF, its respective bypass fuse (BFA_X) must be programmed.

The 16 I/O pins (IO_0 - IO_{15}) and their respective D flip-flop macros can be used in any one of the following configurations:

1. As combinatorial input(s).
   Each of the 16 3-State outputs can be individually disabled by the associated NAND term and the pin is used as an inverting or non-inverting input.
2. As registered DFF outputs.
   These DFFs are split into two banks of 8, and each bank is clocked separately. The bypass fuse BFB_X (see PML2552 Logic Diagram) is used to bypass any one of these DFFs. The flip-flops are all active in an unprogrammed device.
3. As combinatorial outputs.
   By programming the bypass (BFB_X) fuse of any one of the DFFs, the flip-flop(s) is bypassed. The I/O pin can then be used as a combinatorial output.
4. As Internal foldback DFFs or foldback NAND gates.
   When the I/O pin is used as an input, the output macro can be used as an internal DFF or a foldback NAND term. If the bypass fuse is programmed, the macro will act as a foldback NAND term. Otherwise it will act as an internal DFF.

The 8 bidirectional pins (B0-B7) can be used as either combinatorial inputs or outputs with programmable polarity. The Exclusive-OR polarity gates are non-inverting in the unprogrammed state.

The NAND signal labeled '$\overline{OD}$' (Output Disable) shown on the PML2552 logic diagram is used for the Power Down mode operation. This signal disables the outputs when the device enters the Power Down mode and SCI is high.

Clock Array

The 20 buried JKFFs can be clocked through the 'Clock Array'. The Clock Array consists of 11 NAND terms. Ten of these terms are connected to the clock inputs of the Bank A flip-flops that can be clocked individually. One NAND gate is connected to Bank B flip-flops that have a common clock. There are 18 inputs to the clock array. Four come directly from the input pins (with programmable polarity), 4 inputs are from 4 NAND gates connected directly to the folded NAND array. 10 inputs are from the Q outputs of the JKFFs with clear.

SCAN TEST FEATURE

With the rise in the ratio of devices on a chip to the number of I/O pins, Design For Testability is becoming an essential factor in logic design methodology. The PML2552 incorporates a variable length scan test feature which permits access to the internal flip-flop nodes without requiring a separate external I/O pin for each node accessed. Figure 2 (Scan Mode Operation) shows how a scan chain is implemented through the 20 buried JKFFs and 16 output DFFs. Two dedicated pins, SCI (Scan In) and SCM (Scan Mode), are used to operate the scan test. The SCM pin is used to put the circuit in scan mode. When this pin is brought to a logic "1", the circuit enters the scan mode.

In this mode it is possible to shift an arbitrary test pattern into the flip-flops. The SCI pin is used to input the pattern. The inverted outputs of flip-flops D0 - D15 are observable on pins I/O0 - I/O15.

The following are features and characteristics of the device when in Scan Mode:

1. CKE1 is the common scan-clock for all the flip-flops when in scan mode. CKE1 overrides all clock resources of normal operational mode.
2. The Preset (PR) and Clear (CL) functions of the flip-flops are disabled.
3. Scan overrides the bypass fuse of the flip-flops. This means that all the bypassable DFFs remain intact during scan operation even though they may have been bypassed during normal operation.
4. To observe the SCAN data, the output buffers must be enabled by the Output Enable (tri-ctrl) terms.
5. The outputs of the flip-flops are complemented on pins I/O0 - I/O15.
6. All external inputs to flip-flops in the scan chain are disabled when the device enters the scan mode.
7. Blowing the security fuse does not disable the Scan Test feature.

SCAN MODE OPERATION

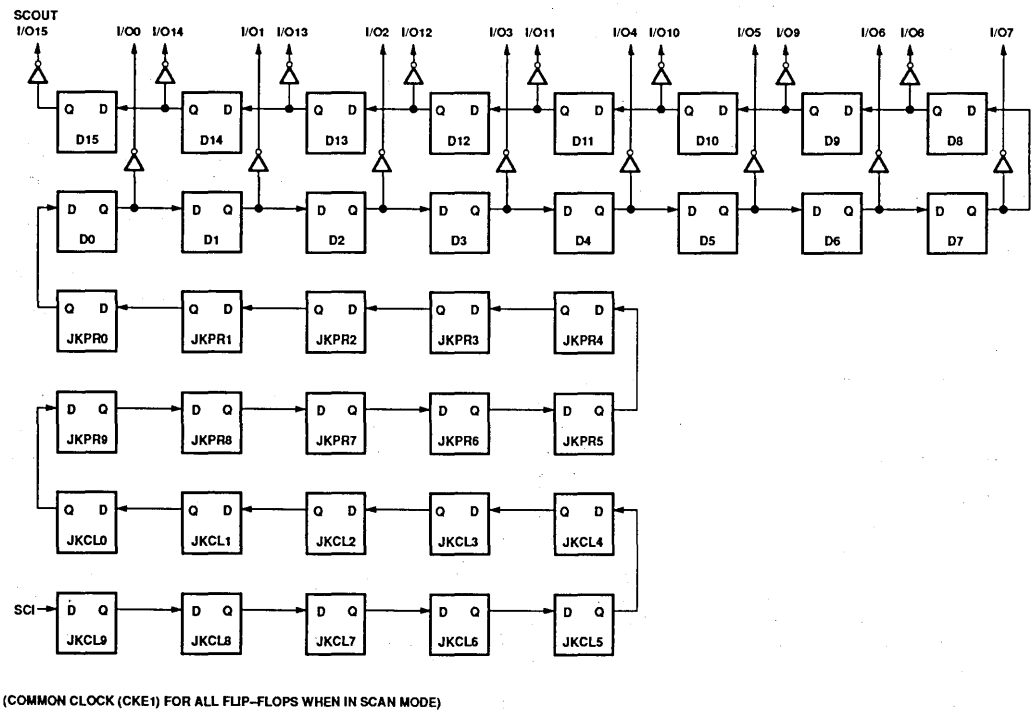


Figure 2

Scan Test Strategy

The scan test pattern is design dependent and the user must make considerations for Design For Testability (DFT) during the initial stages of the design. A typical test sequence is to pre-load (i.e., enter a state); revert to normal operation (i.e., activate the next state transition); go back to scan mode to check the result. Note that the scan test feature available in the PML2552 is a variable length scan chain. The DATA entered at SCI (JKCL9) can be accessed anywhere between 21 clock cycles (at I/O0) to 36 clock cycles (at I/O15). For the strategy discussed here, DATA is read out after 36 clocks at I/O15 (i.e., D15). The following operation sequence suggests a possible scan test method.

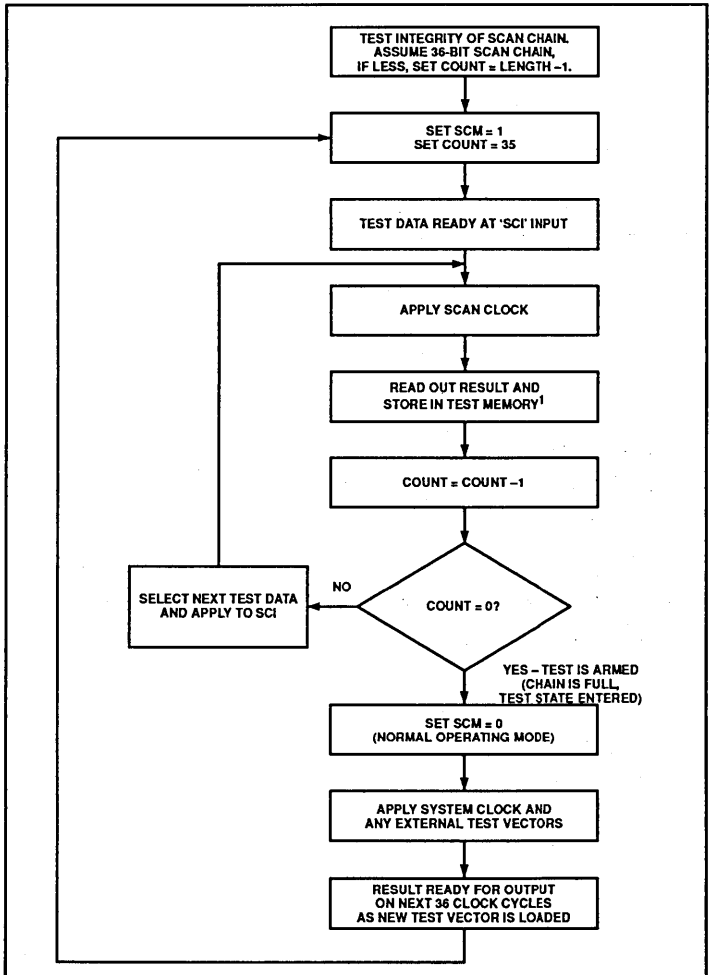
A conservative test policy demands proof that the test facility is working. Thus, to prove Scan Chain holds and maintains correct data:

a. Fill chain with several patterns (for example, all ones and all zeros).

b. Retrieve same patterns.

The user is responsible for managing an external test memory buffer for applied vectors and results, as part of the test equipment.

1. Parallel readout of I/O0 - I/O15 is possible, but assume only I/O15 is used for this strategy.
2. The first DATA entered at SCI (or JKCL9) will be the content of D15 after 36 clocks. This DATA will be inverted at the output pin I/O15 (i.e., SCOUT). The last DATA entering the scan chain will be the content of JKCL9. Thus, the scan chain resembles a first-in-first-out shift register with inverted outputs (I/O0 - I/O15).
3. 'Test Data' is read in at the SCI input and read out of the SCOUT output pin (I/O15). To enter 'Test Data':
   a. Put device in Scan Mode by applying the scan control signals (SCM=1).
   b. Clock device with scan clock (CKE1).
   c. Apply consecutive serial test vectors.
   d. Read back results as new 'Test Data' (States) are applied. The first 36 outputs read at SCOUT (I/O15) are random ('old') data (e.g., remnant of Step 1).
   e. Apply 36 'Test Data' until the chain is full.
4. To apply 'Test Data' (States), exit Scan Mode and apply on system clock together with any other possible test vectors.
5. To read result of the state transition, re-enter scan and apply the scan clock (CKE1). The result of the state transition in JKCL9 will be available at SCOUT (I/O15) after 36 clocks. The results can be stored in a user defined test memory buffer in inverted logic representation.
6. As the results are being read and stored, new 'Test Data' can be entered via SCI.
7. Repeat for all test patterns of interest.
8. Figure 3 (FLOW_CHART) depicts a flow chart version of the test sequence.



NOTE:
1. The first 36 outputs are random ('OLD') data.

Figure 3. FLOW_CHART

A Simple Example

Assume the last three cells of the scan chain (JKCL9, JKCL8, JKCL7 in Figure 4 contain a 3-bit up counter. Our test vector will be a single clock applied to the counter. Suppose we wish to first check the *State 5* (i.e., 101) to *State 6* (i.e., 110) transition, then the *State 3* (i.e., 011) to *State 4* (i.e., 100) transition. Assume the scan chain has been pre-verified and we may begin.

Enter scan mode (set SCM=1)I apply 36 bits in sequence so that the value *101* (i.e., State 5) resides in the last three cells. Exit scan mode (set SCM=0) and apply a single clock to the counter. Now the value *110* (i.e., State 6) resides in the last three cells. Re-enter scan mode (set SCM=1) and read back 36 bits from position I/O15. Note that the outputs are complemented and are also read back in the reverse order. Therefore the value for *STATE 6* read at I/O15 will be 100 which is the complement of *STATE 6* (110) read in the reverse order.

As this is being read back, apply a new state, serially equal to the value *011* (i.e., State 3). This state should be loaded on the last three clock cycles during which *STATE 6* is being read back at I/O15. After *STATE 3* has been loaded (and *STATE 6* read back), exit scan mode and apply a single clock which will invoke the *STATE 3* (i.e., 011) to *STATE 4* (i.e., 100) transition. Re-enter scan mode and read back 36 bits at I/O15. The last three bits should contain *110* which is the complement of *State 4* read in the reverse order. Figure 4 (SCAN_EXAMPLE) shows a flow diagram of this example. Note that the States will always be complemented and read back in the reverse at I/O15. Other sequences may be applied in the same manner.

A possible alternative to this example is to read back the output states at I/O0 (D0) instead of I/O15 (JKCL9). This will allow the outputs to be read back after 21 clock cycles rather than the 36 used in the above example.

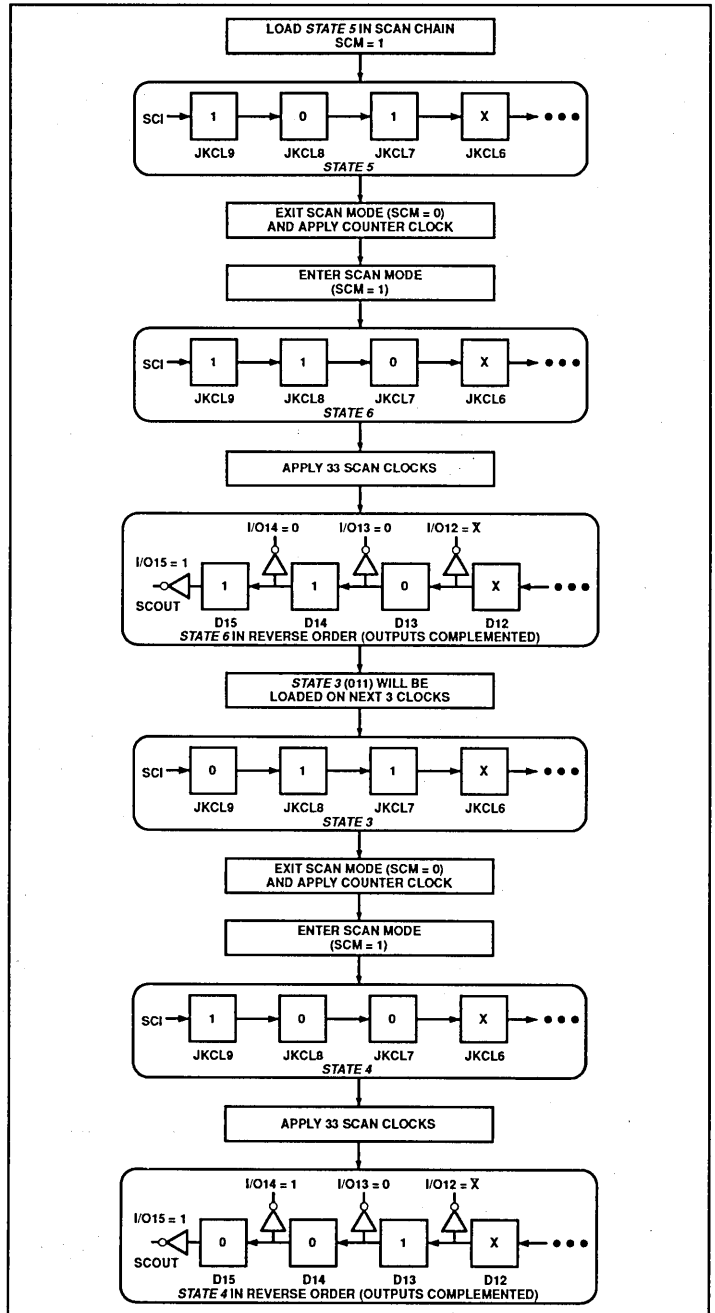


Figure 4. SCAN_EXAMPLE

POWER DOWN

The PML2552 offers the user controlled capability of putting the device to "sleep" where power dissipation is reduced to very low levels. When brought to a logic "1", the PD pin freezes the circuit while reducing the power. All data is retained. This not only includes that of the registers, but also the state of each foldback gate. For those cases where it is desirable to 3-State the outputs, that can be accomplished by raising the SCI pin to a logic "1".

There is one point that should be noted while the circuit is in its power-down mode. The switching of any external clock pin will cause a disruption of the data. All clocks must be frozen before the circuit goes into power-down and stay that way until it powered back up. Clocks that are internally generated and feed the clock array are automatically stopped by the power-down circuitry. Any other input can toggle without any loss of data.

NOTE:

1. During power down, external clocks (CKA, CKB/CKC, CKE1, CKE2) should not change.
2. SCM must be "0" as in normal operation mode.
3. External clock recovery time (low-to-high) is 60ns (high-speed) and 70ns (standard) after the device is powered up.
4. Power Down Timing Diagrams on pages 17 and 18 are for combinatorial operation only.

DEVELOPMENT TOOLS

The PM2552 is supported by the Signetics SNAP software development package and a multitude of hardware and software development tools. These include industry standard PLD programmers and CAD software.

SNAP

Features

- Schematic entry using DASH™ 4.0 or above or OrCAD™ SDT III
- State Equation Entry
- Boolean Equation Entry
- Allows design entry in any combination of above formats
- Simulator
  - Logic and fault simulation
  - Timing model generation for device timing simulation
  - Synthetic logic analyzer format
- Macro library for standard TTL and user defined functions
- Device independent netlist generation
- JEDEC fuse map generated from netlist

SNAP (Synthesis, Netlist, Analysis and Program) is a versatile development tool that speeds the design and testing of PML. SNAP combines a user-friendly environment and powerful modules that make designing with PML simple. The SNAP environment gives the user the freedom to design independent of the device architecture.

The flexibility in the variations of design entry methodologies allows design entry in the most appropriate terms. SNAP merges the inputs, regardless of the type, into a high-level netlist for simulation or compilation into a JEDEC fuse map. The JEDEC fuse map can then be transferred from the host computer to the device programer.

SNAP's simulator uses a synthetic logic analyzer format to display and set the nodes of the design. The SNAP simulator provides complete timing information, setup and hold-time checking, plus toggle and fault grading analysis.

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. A minimum of 640K bytes of RAM is required together with a hard disk.

SLICE

SLICE, which supports Signetics PLD line, is easy to understand and simple to use. Select a PLD, assign input and output pins and enter the desired equations in either Boolean or state form. SLICE then checks the equations for errors. It automatically generates a JEDEC-format fuse map for downloading to a PLD programmer.

Fully menu driven, SLICE incorporates a fuse table editor for making quick modifications to the design and a test vector editor for input of test vectors.

A built-in Boolean equation extractor allows existing PLDs to be used as the basis for a new design. the extractor reads JEDEC information from a PLD and creates a file containing the corresponding Boolean equations. The result can then be used to consolidate several PLD designs into a single, denser part.

And SLICE is upward compatible with Signetics extensive design suite, SNAP.

DESIGN SECURITY

The PML2552 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary design implemented in the device cannot be copied or retrieved.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

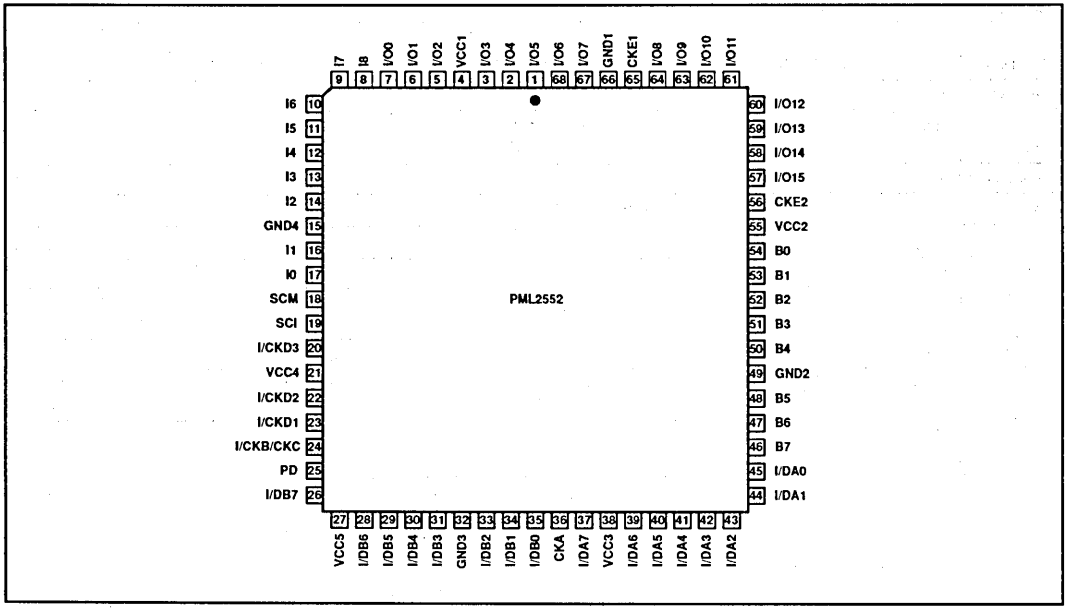
THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DASH is a trademark of Data I/O Corporation.
OrCAD is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

Programmable macro logic PML2552

PIN CONFIGURATION



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
68-pin Plastic Leaded Chip Carrier	PML2552A
68-pin "J" Leaded Ceramic Cerquad Package	PML2552KA

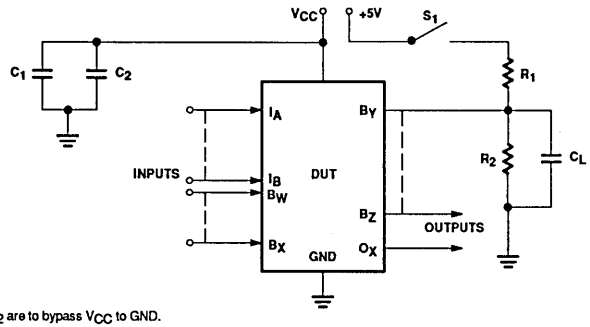
DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS		LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage							
V_{IL}	Low	V_{CC} = MIN		–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX		2.0		V_{CC} + 0.3	V
Output voltage							
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 5mA				0.45	V
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA		2.4			V
Input current							
I_{IL}	Low	V_{IN} = GND				–10	µA
I_{IH}	High	V_{IN} = V_{CC}				10	µA
Output current							
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC}				10	µA
		V_{OUT} = GND				–10	µA
I_{OH}	Output High	V_{CC} = MIN, V_{OUT} = 2.4V				–2	mA
I_{OL}	Output Low	V_{CC} = MIN, V_{OUT} = 0.45V				5	mA
I_{OS}	Short-circuit[5]	V_{OUT} = GND				–100	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX, No load	CMOS input[2]		60	100[6]	mA
		f = 1MHz	TTL input[3]		65	120[6]	mA
I_{SB}	Standby V_{CC} supply current	V_{CC} = MAX, No load	CMOS input		1.0	10	mA
		PD = V_{IH}	TTL input		1.5	10	mA
Capacitance							
C_{IN}	Input	V_{CC} = 5V, T_{amb} = +25°C, V_{IN} = 2.0V			8		pF
C_B	I/O	V_{CC} = 5V, T_{amb} = +25°C, V_{IO} = 2.0V			16		pF

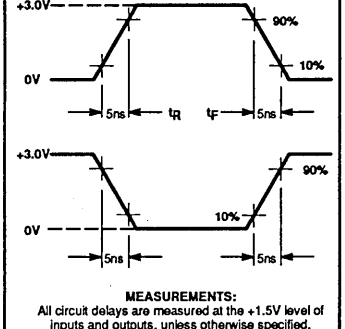
NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. CMOS inputs: V_{IL} = GND, V_{IH} = V_{CC}.
3. TTL inputs: V_{IL} = 0.45V, V_{IH} = 2.4V.
4. All voltage values are with respect to network ground terminal.
5. Duration of short-circuit should not exceed one second. Test one at a time.
6. ΔI_{CC} vs. Frequency = 4mA/MHz max.

TEST LOAD CIRCUITS



NOTES:
C_1 and C_2 are to bypass V_{CC} to GND.
Test Load R_1 = 750Ω, R_2 = 442Ω, C_L = 30pF (C_L = 5pF for Output Disable)
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

VOLTAGE WAVEFORMS



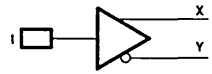
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

MACRO CELL AC SPECIFICATIONS

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V
(SNAP Resource Summary Designations in Parentheses)

Input Buffer
(DIN552, NIN552, BDIN55, BNIN552 CDIN552, CNIN552, CKDIN552, CKNIN552)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	X	I	3	6	10	ns
t_{PLH}	X	I	3	6	10	ns
t_{PHL}	Y	I	3	6	10	ns
t_{PLH}	Y	I	3	6	10	ns

Input Pins: 8–14, 16, 17, 20, 22–24.
Bidirectional Pins: 1–3, 5–7, 46–48, 50–54, 57–64, 67, 68.

Internal NAND of Main Array
(FBNAND, NAND)

SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Y	X	6	12	20	ns
t_{PLH}	Y	X	6	12	20	ns

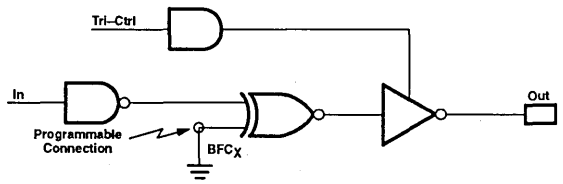
Internal NAND of Clock Array
(NAND)

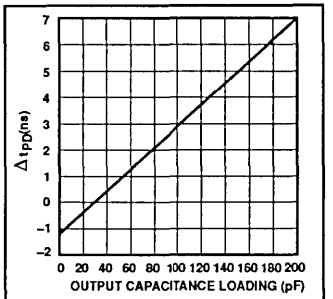
SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Y	X	3	6	10	ns
t_{PLH}	Y	X	3	6	10	ns

MACRO CELL AC SPECIFICATIONS (Continued)

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V
(SNAP Resource Summary Designations in Parentheses)

3-State Output with Programmable Polarity (TOUT552 + EXOR552)



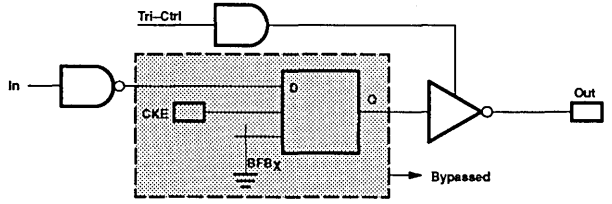


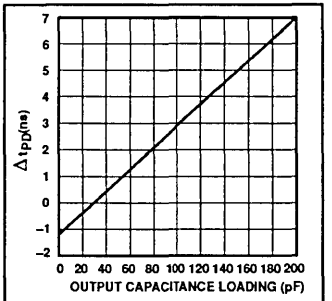
ΔtPD vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	9	18	30	ns
t_{PLH}	Out	In	9	18	30	ns
t_{OE}[4]	Out	Tri-Ctrl	3	6	10	ns
t_{OD}[4]	Out	Tri-Ctrl	3	6	10	ns

Bidirectional Pins: 46–48, 50–54.

I/O Output Buffer with 3-State Control, DFF Bypassed (TOUT552 + NAND)





ΔtPD vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	9	18	30	ns
t_{PLH}	Out	In	9	18	30	ns
t_{OE}[4]	Out	Tri-Ctrl	3	6	10	ns
t_{OD}[4]	Out	Tri-Ctrl	3	6	10	ns

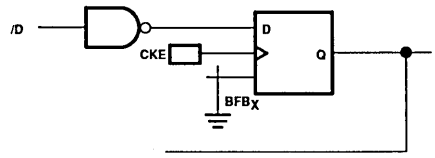
I/O Pins: 1–3, 5–7, 57–64, 67, 68.

Notes on page 461.

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

D FLIP-FLOP

Output DFF Used Internally (ODFF552)



SYMBOL	PARAMETER	LIMITS			UNIT
		MIN	TYP	MAX	
f_{CKE}	Flip-flop toggle rate			50	MHz
$t_{W\,CKE}$ High	Clock HIGH	10			ns
$t_{W\,CKE}$ Low	Clock LOW	10			ns
t_{SETUP} /D	/D setup time to CKE	15			ns
t_{HOLD} /D	/D hold time to CKE	4			ns

SYMBOL	PARAMETER		LIMITS			UNIT
	From (Input)	To (Output)	MIN	TYP	MAX	
t_{PLH}	CKE ↑	Q	6	12	20	ns
t_{PHL}	CKE ↑	Q	6	12	20	ns

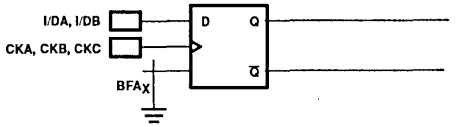
MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

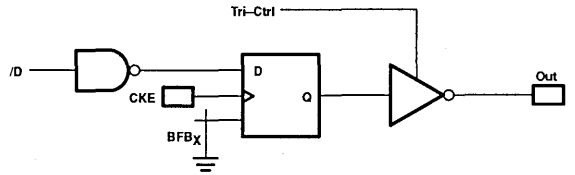
D FLIP-FLOP (Continued)

Input and Output (IDFF552 & ODFF552)

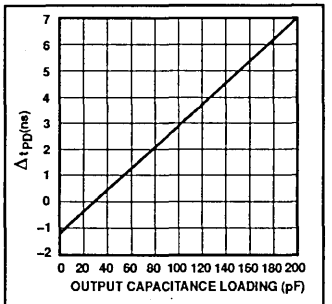
INPUTS		OUTPUTS	
CK	D	Q	$\overline{Q}$
L	X	Q_0	$\overline{Q}_0$
↑	H	H	L
↑	L	L	H

NOTE:
Q_0, $\overline{Q}_0$ represent previous stable condition of Q, $\overline{Q}$.





SYMBOL	LIMITS			UNIT
	MIN	TYP	MAX	
$f_{CKA, CKB, CKC}$			50	MHz
$t_{W\ CKA, CKB, CKC}$ High	10			ns
$t_{W\ CKA, CKB, CKC}$ Low	10			ns
t_{SETUP} I/DA, I/DB	5			ns
t_{HOLD} I/DA, I/DB	10			ns
f_{CKE}			50	MHz
$t_{W\ CKE}$ High	10			ns
$t_{W\ CKE}$ Low	10			ns
t_{SETUP} /D	15			ns
t_{HOLD} /D	4			ns



Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS			UNIT
	From (Input)	To (Output)	HIGH–SPEED			
			MIN	TYP	MAX	
t_{PLH}	CKA, CKB/CKC ↑	Q, $\overline{Q}$	3	6	10	ns
t_{PHL}	CKA, CKB/CKC ↑	Q, $\overline{Q}$	3	6	10	ns
t_{PLH}	CKE ↑	Out	9	18	30	ns
t_{PHL}	CKE ↑	Out	9	18	30	ns

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

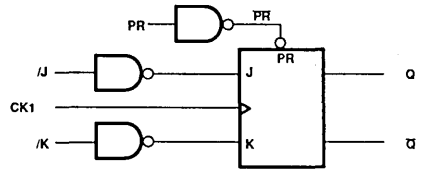
JK FLIP-FLOPS

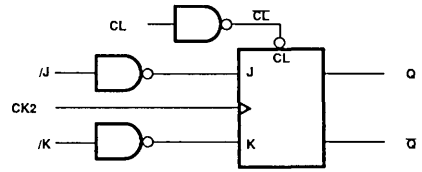
(JKPR552)

INPUTS				OUTPUTS	
$\overline{PR}$	CK	J	K	Q	$\overline{Q}$
L	X	X	X	H	L
H	↑	L	L	Q_0	$\overline{Q}_0$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	TOGGLE	
H	L	X	X	Q_0	$\overline{Q}_0$

(JKCL552)

INPUTS				OUTPUTS	
$\overline{CL}$	CK	J	K	Q	$\overline{Q}$
L	X	X	X	L	H
H	↑	L	L	Q_0	$\overline{Q}_0$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	TOGGLE	
H	L	X	X	Q_0	$\overline{Q}_0$





SYMBOL	PARAMETER	LIMITS			UNIT
		MIN	TYP	MAX	
f_{CK1}	CK1 toggle frequency			50	MHz
f_{CK2}	CK2 toggle frequency			50	MHz
$t_{W\,CK1}$ High	CK1 clock HIGH	10			ns
$t_{W\,CK1}$ Low	CK1 clock LOW	10			ns
$t_{W\,CK2}$ High	CK2 clock HIGH	10			ns
$t_{W\,CK2}$ Low	CK2 clock LOW	10			ns
t_{SETUP} /J, /K	/J, /K setup time to CK1, CK2	30			ns
t_{HOLD} /J, /K	/J, /K hold time to CK1, CK2	0			ns
t_W PR Low	Preset Low period	10			ns
t_W CL Low	Clear Low period	10			ns

SYMBOL	PARAMETER		LIMITS			UNIT
	From (Input)	To (Output)	MIN	TYP	MAX	
t_{PLH} t_{PHL}	CK1,2 CK1,2	Q, $\overline{Q}$ Q, $\overline{Q}$	1.5 1.5	3 3	5 5	ns ns
t_{PLH} t_{PHL}	PR PR	Q, $\overline{Q}$ Q, $\overline{Q}$	7.5 7.5	15 15	25 25	ns ns
t_{PLH} t_{PHL}	CL CL	Q, $\overline{Q}$ Q, $\overline{Q}$	7.5 7.5	15 15	25 25	ns ns

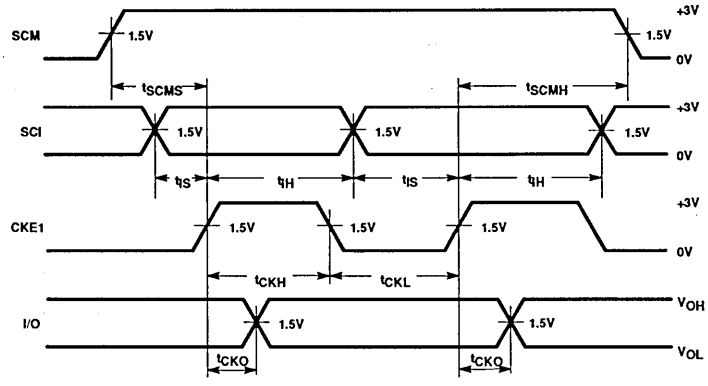
AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$, $V_{PP} = V_{CC}$,
$R_1 = 750\Omega$, $R_2 = 442\Omega$, $C_L = 5pF$ for Output Disable) (See Test Load Circuit Diagram)

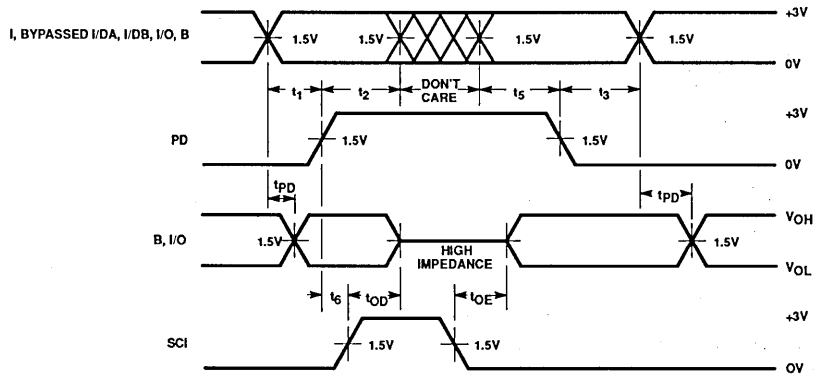
SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
Scan mode operation[1]				
t_{SCMS}	Scan Mode (SCM) Setup time	15		ns
t_{SCMH}	Scan Mode (SCM) Hold time	25		ns
t_{IS}	Data Input (SCI) Setup time	5		ns
t_{IH}	Data Input (SCI) Hold time	5		ns
t_{CKO}	Clock to Output (I/O) delay		30	ns
t_{CKH}	Clock High	10		ns
t_{CKL}	Clock Low	10		ns
Power down, power up[2]				
t_1	Input (I, bypassed I/DA, I/DB, I/O, B) setup time before power down	40		ns
t_2	Input hold time	30		ns
t_3	Power Up recovery time		60	ns
t_4	Output hold time	0		ns
t_5	Input setup time before Power Up	20		ns
t_{OE}	SCI to Output Enable time[3]		40	ns
t_{OD}	SCI to Output Disable time[3]		40	ns
t_6	Power Down setup time	10		ns
t_7	Power Up to Output valid		70	ns
Power-on reset				
t_{PPR1}	Power-on reset output register (Q = 0) to output (I/O) delay		10	ns
t_{PPR2}	Power-on reset input register (Q = 0), buried JK Flip-Flop (Q = 0) to output (B, bypassed I/O) delay		40	ns

NOTE:
1. SCM recovery time is 50ns after SCM operation. 50ns after SCM operation, normal operations can be resumed.
2. Timings are measured without foldbacks.
3. Transition is measured at steady state High level (–500mV) or steady state Low level (+500mV) on the output from 1.5V level on the input with specified test load ($R_1 = 750\Omega$, $R_2 = 442\Omega$, $C_L = 5pF$). This parameter is sampled and not 100% tested.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TIMING DIAGRAMS

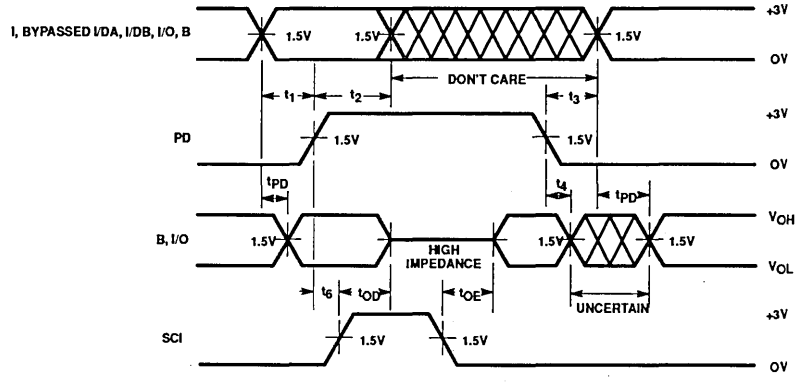


Scan Mode Operation

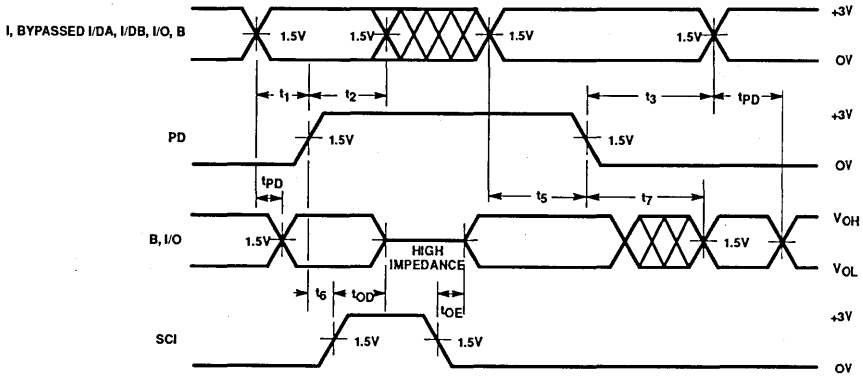


Power Down, Power Up
Input (old) Ready Before Power Up

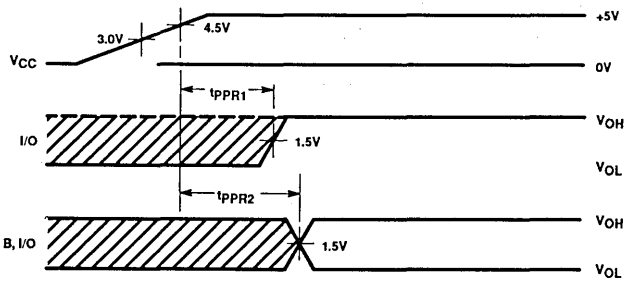
TIMING DIAGRAMS (Continued)



Power Down, Power Up
Input (new) Ready After Power Up



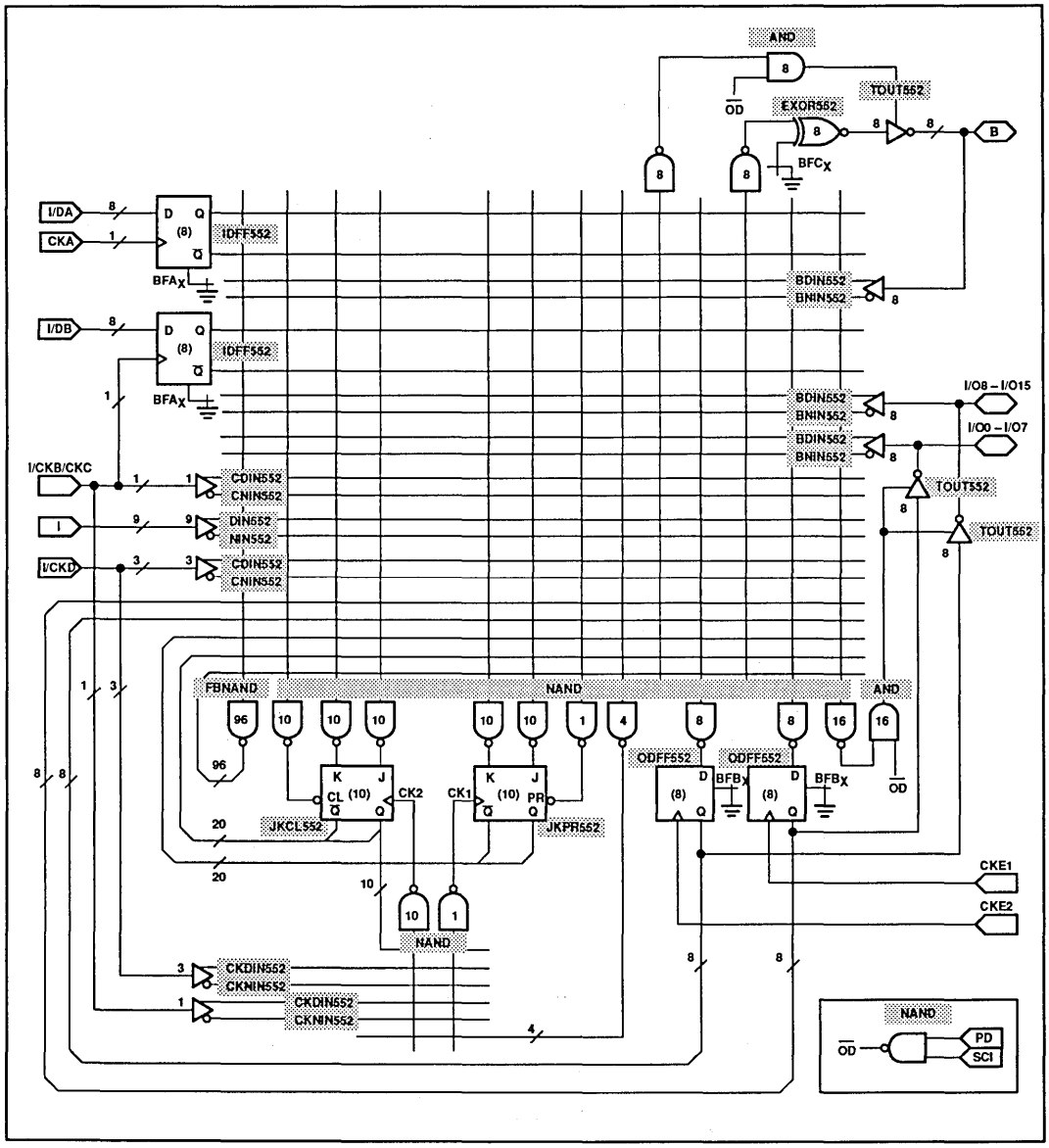
Power Down, Power Up
Input (new) Ready Before Power Up



Power-On Reset

Programmable macro logic PML2552

SNAP RESOURCE SUMMARY DESIGNATIONS



ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PML2552 device is such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PML2552 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PML2552 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the PML2552 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000μW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12,000μW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

PROGRAMMING

Refer to the following charts for qualified manufacturers of programmers and software tools:

PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	UNISITE 40/48 V2.8 Pinsite – V2.0	15908C* (with adaptor) 15908D (with pinsite)
STREBOR DATA COMMUNICATIONS 1008 N. NOB HILL AMERICAN FORK, UT 84003	PLP–S1A Programmer MP68CC Adaptor	

* Needs a 40-pin DIP to 68-pin PLCC adaptor that is available from Emulation Technology.
Part Number: AS–68–40–04P–6

EMULATION TECHNOLOGY, INC.
2368B Walsh Avenue, Blvd. D
Santa Clara, California 95051
Telephone No. (408) 982–0660
Fax. No. (408) 982–0664

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
PHILIPS COMPONENTS–SIGNETICS 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SNAP SOFTWARE REV. 1.4 AND LATER

Section 7
Military Selection Guide

INDEX

Military Selection Guide

Programmable Logic Devices

Part Number	Device Description	Package Description	M38510/	JB**	JS**	Standard MIL-Drawing	MIL-Drawing Status**
PLHS18P8/BRA	PAL	20DIP3				5962-8752801RX	A
PLHS18P8/BSA	PAL	20FLAT		NA	NA	5962-8752801SX	A
PLS159A	PLS	20DIP3		NA	NA	PLANNED	NA
PLS173/BLA	PLA	24DIP3		NA		5962-8850402LA	A
PLS179/BLA	PLS	24DIP3		NA	NA	5962-8850701LA	A
82S100/BXA	PLA	28DIP6	50202	A	NA		
82S100/BYA	PLA	28FLAT	50202	NA	NA		
82S100/B3A	PLA	28LLCC	50202	NA	NA		
82S101/BXA	PLA	28DIP6	50201	A	NA		
82S101/BYA	PLA	28FLAT	50201	NA	NA		
82S101/B3A	PLA	28LLCC	50201	NA	NA		
82S105/BXA	PLS	28DIP6		NA	NA	5962-8670901XA	A
82S105/BYA	PLS	28FLAT		NA	NA	5962-8670901YA	A
82S105/B3A	PLS	28LLCC		NA	NA	5962-86709013A	A
82S153A/BRA	PLA	20DIP3		NA	NA	5962-8768201RA	A
82S153A/BSA	PLA	20FLAT		NA	NA	5962-8768201SA	A
82S153A/B2A	PLA	20LLCC		NA	NA	5962-87682012A	A
PLC42VA12/BLA	PLS	24DIP		NA	NA	PLANNED	NA
PLC42VA12/BYA	PLS	24FLAT		NA	NA	PLANNED	NA
PLC42VA12/B3A	PLS	28LLCC		NA	NA	PLANNED	NA
PML2552/BUA	PML	68LLCC		NA	NA	PLANNED	NA
PLC18V8Z/BRA	PAL	20DIP		NA	NA	PLANNED	NA
PLC18V8Z/BSA	PAL	20FLAT		NA	NA	PLANNED	NA
PLC18V8Z/B2A	PAL	20LLCC		NA	NA	PLANNED	NA

* Not available as a Class B standard product. See M38510 and/or Military Drawing columns for availability
** A = available, NA = not available, IP = in process, call for availability.

Section 8
Development Software

INDEX

SNAP 16
Synthesis Netlist Analysis Program

Programmable Logic Development Software

FEATURES

- Schematic entry available using Data I/O SNAP-DASH™ OrCAD SDT III™
- State equation entry
- Boolean equation entry
- Netlist entry
- Simulation waveform entry
- Capability to design in one or any combination of formats
- Device independent, netlist based design platform
- Boolean equation extractor
- Fuse table editor
- Philips LESIM 5-State gate array simulator as well as Signetics SIGSIM:
  - Logic and fault simulation
  - Model extraction and timing simulation
  - Synthetic logic analyzer format
  - Stimuli entry in waveform format
- Freezing of selected Critical paths
- Capability to create user defined macros
- Full documentation of design and simulation results in waveform format
- JEDEC fusemap compiler and device programmer interface

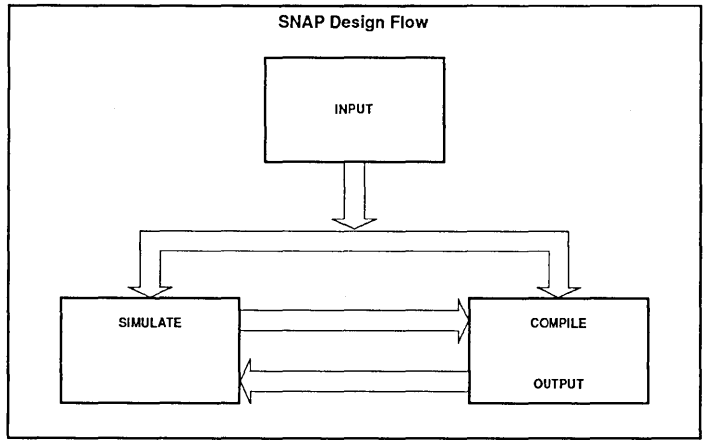
GENERAL DESCRIPTION

SNAP PLD development software. Simple-to-use tools for demanding designs.

Get ready for greater design productivity. SNAP, the complete logic synthesis, simulation and layout package for Signetics full line of PLDs, saves one commodity in preciously short supply: design time. **Fully equipped with every tool you need to turn out PLD designs quickly,** SNAP eliminates the "learning curve" that can keep you from being immediately productive. Regardless of whether you're a PLD novice or seasoned pro, SNAP allows you to produce optimized designs within a matter of hours.

For rapid design you need flexibility and SNAP provides lots of it. Enter your design in the most convenient way possible — using **any combination of schematics, waveforms, Boolean equations, state equations or netlists.** SNAP merges the inputs and generates a dense, high-speed design that can be simulated in SNAP's powerful simulator and then downloaded to a PLD programmer.

With SNAP, you produce your design in a netlist-based, device-independent environment. No need to commit to a particular part from the start of the design process: With SNAP, you can change the target PLD at will. If you find that your design needs a larger device or can fit into a smaller, less expensive one, simply select a new part and resimulate. SNAP allows you to take advantage of the most appropriate PLD for the job without wasting time.



SNAP-DASH is a trademark of Data I/O Corp.
OrCAD STD III is a trademark of OrCAD, Inc.

Programmable logic development software SNAP 16

SNAP'S UNRIVALED SIMULATION FACILITY

Simulation is a key part of the SNAP design process. **SNAP incorporates Philips 5-State gate array simulator,** a simulator so unsurpassed in its accuracy and diagnostic ability that it is a standard tool used by the company's own chip designers. You can examine any of your design's internal nodes and apply SNAP's virtual logic analyzer to display the precise timing at that node. Then change the stimulus and put the design through its paces with SNAP's built-in waveform editor. Make changes using the SNAP fuse table editor and resimulate. When you finally program a PLD, chances are that it will run perfectly the first time.

Since testability represents an ever-important measure of the success of a PLD design, **SNAP includes a powerful fault simulator** that simplifies the task of maximizing fault coverage. The tool rapidly generates a report detailing undetected and potentially undetectable faults, coverage efficiency, and other useful data. With it, you get the most thorough fault coverage possible in a limited test period.

Would you like to know how many potential faults your test vectors can detect?
Just look at the output of the SNAP FAULT SIMULATOR...

```
FAULT LIST:
   TOTAL NUMBER OF SIGNALS                =     6
   NUMBER OF NAMED SIGNALS                =     6
   NUMBER OF CIRCUIT FAULTS               =    12
   NUMBER OF INSERTED FAULTS              =    10
   NUMBER OF COLLAPSED FAULTS             =     2

FAULT DETECTION:
   NUMBER OF HARD DETECTED FAULTS         =    12
   NUMBER OF POTENTIALLY DETECTED FAULTS  =     0
   NUMBER OF UNDETECTED FAULTS            =     0

FAULT COVERAGE:
   HARD DETECTION FAULT COVERAGE          =  100.0%
   POTENTIAL DETECTION FAULT COVERAGE     =  0.0  %
   TOTAL DETECTION FAULT COVERAGE         =  100.0%

HARD DETECTION FAULT COVERAGE VERSUS PATTERN# :
PATTERN#   %      0         20        40        60        80        100
-----------------+---------+---------+---------+---------+---------+
       1  58.3    .****************************
       2  75.0    .************************************
       3  83.3    .*******************************************
       4  100.0   .**************************************************
       5  100.0   .**************************************************
-----------------+---------+---------+---------+---------+---------+
```

Programmable logic development software SNAP 16

Designers who need to consolidate the designs of existing logic devices will draw considerable benefit from SNAP's unique **Boolean equation extractor.** It take the design data from existing PLDs and converts it to the actual, corresponding Boolean equations, which can then be used as an input to SNAP. It eliminates the need to find and re-enter design data, often a time-consuming process.

And for added convenience, **SNAP features the powerful logic optimizer, Espresso Minimizer.** Espresso automatically removes all unnecessary gates from your design, assuring that it will be the fastest and densest possible. Espresso allows you to pack more in — or fit it into a smaller PLD. The result can be substantial cost and power savings.

FROM THIS:

```
L0000
11111111111111011101111011111111111111111
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
11111111111111011111111111111111111011111
```

TO THIS:

```
@LOGIC EQUATION
S.D     =/(ASN*RWN*/DREQN1+S*/DREQN1);
```

AUTOMATICALLY • IMMEDIATELY • EFFORTLESSLY

FULL SUPPORT NOW — AND INTO THE FUTURE

SNAP supports Signetics broad line of PLDs, which includes high-speed PAL®-type devices, programmable logic arrays, programmable logic sequencers, and sophisticated programmable macro logic. It is fully compatible with SLICE, Signetics entry-level design package. And as Signetics introduces new PLDs in the future, SNAP will support those too, in a timely manner. You can standardize on SNAP for your future development, with confidence.

Menu-driven and supported by clear, concise documentation, SNAP is a pleasure to use. But if problems do arise, Signetics network of field applications engineer5s stand ready to help. Specially trained and backed by a comps of factory experts, Signetics FAEs are stationed in all major cities in the U.S. and overseas. Wherever you are, chances are that support is nearby.

PRODUCT SUPPORT

SLICE supports the Signetics line of PLDs, which ranges from high-speed PAL®-type devices to complex Programmable Macro Logic circuits. It will also support new Signetics PLDs as they are introduced. The devices currently supported are:

Programmable Logic Arrays:

PLUS153
PLUS173

Programmable Macro Logic:

PLHS501	PLHS601
PLHS502	PML2552

Programmable Logic Sequencers:

PLC42VA12	PLUS105
PLC415	PLUS405

Programmable Array Logic

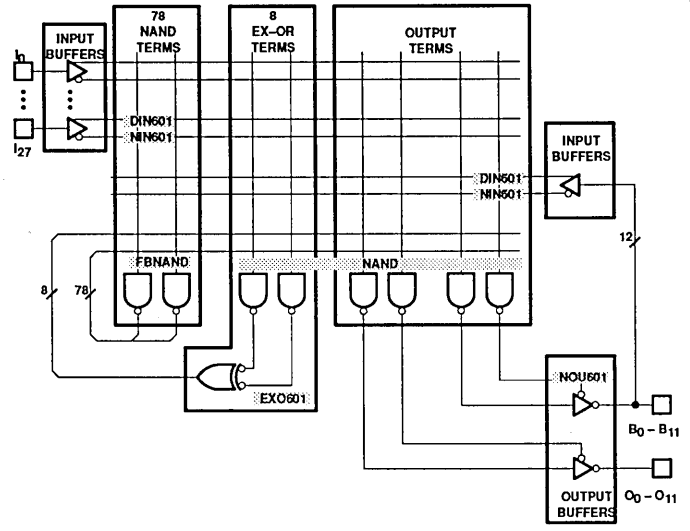
PLUS20L8	PHD16N8
PLUS20R8	PLUS16L8
PLUS20R6	PLUS16R8
PLUS20R4	PLUS16R6
10H20EV8	PLUS16R4
10020EV8	PLQ22V10
PHD48N22	PLC18V8Z

PAL is a registered trademark of AMD/MMI, Inc.

Programmable logic development software

SNAP 16

Did you use your device to its full capability?



For the answer,
Just look at the output of SNAP:

```
Resources Summary
   Cell name   used/total      %
=========================================
      DIN601    23 /  40     57%
      NIN601     5 /  40     12%
      FBNAND     1 /  78      1%
        NAND    48 /  64     75%
      EXO601     0 /   8      0%
      NOU601    24 /  24    100%
Please hit any key to continue...
```

PLHS601 Resources

TRY IT — YOU'LL LIKE IT

Pop the enclosed SNAP demo disk into your computer and see how easy PLD design can be. The demo, like SNAP itself, runs on almost any IBM® PC or compatible having DOS 2.1 or higher, 640K RAM and a hard disk.

IBM is a registered trademark of International Business Machines Corp.

SLICE 10

Signetics Logic Integration Computer Environment

Programmable Logic Development Software

FEATURES

- Easy to learn and use
- Supports Signetics PLD line
- State or Boolean equation entry
- Fuse table editor
- Test vector editor
- Boolean equation extractor
- JEDEC fusemap compiler
- Interfaces to standard PLD programmers
- Upgradeable to SNAP

PRODUCT SUPPORT

SLICE supports the Signetics line of PLDs, which ranges from high-speed PAL®-type devices to complex Programmable Macro Logic circuits. It will also support new Signetics PLDs as they are introduced. The devices currently supported are:

Programmable Logic Arrays:
PLUS153
PLUS173

Programmable Macro Logic:

PLHS501	PLHS601
PLHS502	PML2552

Programmable Logic Sequencers:

PLC42VA12	PLUS105
PLC415	PLUS405

Programmable Array Logic

PLUS20L8	PHD16N8
PLUS20R8	PLUS16L8
PLUS20R6	PLUS16R8
PLUS20R4	PLUS16R6
10H20EV8	PLUS16R4
10020EV8	PLQ22V10
PHD48N22	PLC18V8Z

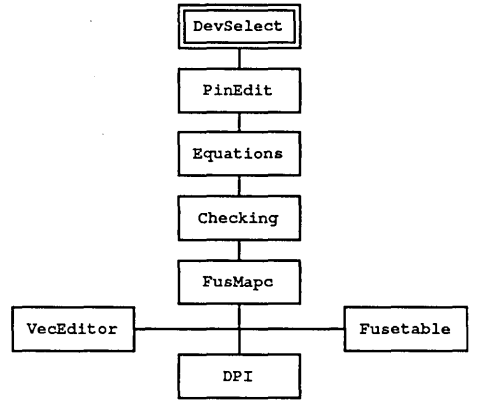
GENERAL DESCRIPTION

Sit down at your PC, install the SLICE software, and **you'll be programming PLDs within the hour.** SLICE (Signetics Logic Integration Computer Environment) provides all the functions you need for speedy PLD development without the tedious learning curve that accompanies other PLD design tools. **It allows first-time users to immediately produce a working PLD design—in the very first session.**

SLICE, which supports Signetics PLD line, is **easy to understand and simple to use.** Select a PLD, assign input and output pins and enter the desired equations in either Boolean or state form. SLICE then checks the equations for errors. It automatically generates a JEDEC-format fuse map for downloading to a PLD programmer.

As easy as 1, 2, 3:

1. **Select device and enter pins.**
2. **Enter equation.**
3. **Generate fuse map and download to device.**



PAL is a trademark of AMD/MMI.

Programmable logic development software — SLICE 10

Fully menu driven, SLICE incorporates a **fuse table editor** for making quick modifications to the design and a test vector editor for input of test vectors.

A built-in Boolean equation extractor allows existing PLDs to be used as the basis for a new design. The extractor reads JEDEC information from a PLD and creates a file containing the corresponding Boolean equations. The result can then be used to consolidate several PLD designs into a single, denser part.

And SLICE is **upward compatible with Signetics extensive design suite, SNAP.** SNAP, in fact, is a superset of SLICE. Among other enhancements, it provides full logic and fault simulation capabilities and design entry using the popular Data I/O DASH and OrCAD STD III schematic capture tools. Compatibility means users can move up to SNAP for more complex design, while making full use of files created under SLICE.

FROM THIS:

```
L0000
11111111111111011101111011111111111111111
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000000000000000000
11111111111111011111111111111111111011111
```

TO THIS:

```
@LOGIC EQUATION
S.D     =/(ASN*RWN*/DREQN1+S*/DREQN1);
```

AUTOMATICALLY • IMMEDIATELY • EFFORTLESSLY

ORDERING INFORMATION

To order your **FREE** SLICE 10 package, contact your local Philips Components–Signetics sales or representative office.

Interpreting the SLICE fusetable

Programmable Logic Devices

INTERPRETING THE SLICE FUSETABLE

A PLD fusetable or program table is a representation of how the device is actually programmed. It may be used to hand code the device, or more importantly, check that the design software implemented the design efficiently. Sometimes the way equations are written affects their implementation. It is recommended if a change in the program table is desired, that it NOT be modified directly. The original equations should be edited, compiled and a new JEDEC fuse table generated.

SLICE contains a module called 'Fusetable' which is a program table editor. It reads in the JEDEC fuse table (.JED file) and displays the program table on the screen. The program tables displayed are similar in appearance to the program tables printed on device data sheets and in the Signetics PLD Data Manual. SLICE fusetable representations differ slightly from those in data sheets. Data sheet program tables contain boxes, headers and labeling surrounding each section while SLICE contains a box in the upper right of the screen describing the cursor location. The characters describing the different fusing configurations are the same.

We will start out first by looking at a simple PHD16N8 program table. Next, different device program tables will be presented from devices of varying levels of complexity up to the PLHS501. The concepts used in all program tables are similar and once the user is familiar with the character representations used, any program table should be easy to interpret. Three areas of connections to especially note are the input/feedback buffers to product terms, OR gate inputs, and JK type flip-flop inputs. Each of these areas either use different or have unique definitions of the fusing representation characters.

The PHD16N8 is a very simple and useful high-speed device. It contains only 16 product terms (AND gates), 8 of which are connected through 3-State inverters to pins. The remaining 8 product terms control the 3-State function of the output buffers. It contains no OR gates. Each product term may receive inputs from up to 16 sources. There are 10 direct input pins and 6 feedbacks from bidirectional pins. Each possible input source goes through a buffer which has an inverting and a non-inverting output. Four characters are therefore required to show how each input source is connected. A SLICE representation of the program table is shown below with four product terms programmed.

```
----------------
--------HH------
----------------
---------L------
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
```

There are 16 lines of 16 characters. Each line represents a product term. Each character represents the way an input source is connected to the product term. By moving the cursor around the program table, a box in the upper right of the screen will display the product term number and input designator. In the PHD16N8 program table, product terms controlling the output buffer 3-State function are arranged in the program table immediately above the product term for the output function. The four characters used to denote connections between an input or feedback buffer and a product term are H, L, –, and 0.

A zero (0) is the default or virgin state designator for the connection between an input buffer and product term. It indicates that both the non-inverting and inverting outputs from the buffer are connected to a product term's input. This has the effect of holding the product term's input and thus it's output LOW. A zero is usually not found on a row alone. To hold a product term low, it is better to connect all inputs of a product term to all buffer outputs. This is because the two paths through a buffer do not have equal delays. If only one buffer was used to hold a product term LOW, any signal on the input of the buffer may cause a glitch on the output of the product term.

A dash (–) is the opposite of a zero. It denotes neither the non-inverting or inverting output of a buffer is connected to the product term. If all of the inputs to a product term are dashed, then an internal pull-up guarantees that the output of the product term will be HIGH. This is used frequently for product terms controlling 3-State output buffers. In the PHD16N8 example above, the first and third product terms control the 3-State output buffers of the second and fourth product terms. They are all dashed so the two outputs are constantly enabled. The other 3-State control product terms are all zeroed so the unused outputs are in a high-impedance mode.

The H and L characters denote respectively a connection between only the non-inverting and inverting buffer outputs and the product term's input. Reading the H and L characters on a specific product term's row almost makes the device appear to be a collection of many comparators. For example, the second product term of the PHD16N8 program table above is:

```
--------HH------
```

The output of the product term will only be HIGH when the inputs match that pattern. In other words, the output will only be HIGH when I0 is HIGH AND I2 is HIGH. The fourth product term is:

```
---------L------
```

The output of this product term will be HIGH only when I0 is LOW. Note that the PHD16N8 has a 3-State inverting buffer following the product term. Therefore, the device pin will be LOW when the product term is HIGH.

A FIXED-OR architecture PLD such as the PLQ22V10 is represented by the following program table segment.

```
HHHHHHHHHH
DDDBBBBBBB
----------------------
-------------------L--
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
----------------------
-------------------LHH
-------------------HLH
-------------------HLL
-------------------LLL
```

Interpreting the SLICE fusetable

At the top of the program table are two lines which are shorter than the regular product term lines. These lines indicate the functions of each output macrocell (OMC). When the cursor is moved over the top row, the upper right box of SLICE will show POLARITY. Each character indicates how a particular output's polarity is programmed, either H or L indicating Active-High or Active-Low. Output F9 is on the left and F0 is on the right. The next line will show CONTROL indicating whether an OMC is set for D-type flip-flop or combinatorial output. Valid characters in this line are D for flip-flop and B for bidirectional combinatorial output.

The OR terms are not shown by SLICE because they are fixed. If the cursor is moved over a product term the indicator in the upper right of the editor will show with which output pin the product term is associated. The 3-State control product term for a specific output occurs before the associated group of ORed product terms. For example, if the cursor is moved over the first product term, the indicator will show OE9-P23 which means that this product term is the output enable control for output F9. Moving the cursor down to the second product term shows P0-F9 which indicates this term is one which feeds into the OR gate of output F9.

The next program table segment is from a PLUS153. This segment is from the top portion of the program table.

```
------LH----------A.AAAAAAAA
------HL----------A.AAAAAAAA
------HH----------.AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
```

A new section has been added to what we've seen previously. The new section indicates the programming of the OR array and is represented on the right portion by "A" and dot (.) characters. In any PLA architecture device the OR array is connectable to any product term. This enables product terms to be 'shared' between outputs. For example, if an output called OUT1 had an equation of "OUT1 = IN1 * IN2 + IN3" and OUT2 had an equation of "OUT2 = IN1 * IN2" only two product terms would be needed for a PLA while three would be needed for a FIXED-OR architecture device. One product term (IN1 * IN2) is common to both outputs and in a PLA may be connected to both OUT1 and OUT2. The FIXED-OR device would have to duplicate this product term – one for each output. In addition, a bipolar PLA device may even be reprogrammed if the changes are small. This can be achieved by disconnecting the unwanted product term from the OR array, adding the desired product term to the end of the program table and connecting it to the desired output.

The OR section of the PLUS153 program table consists of 10 columns, one for each output and 24 rows, one for each possible product term connection (the other 10 product terms control the 3-State output buffers). We noted in the preceding paragraphs for product terms that each input buffer had both inverting and non-inverting outputs and thus four characters were required to describe possible connections. The AND gates only have only one output and therefore only 2 characters are needed to represent a connection and lack thereof. This is denoted by an A and a dot (.) respectively. Also, this section is read vertically, NOT horizontally as the product term inputs. If a PLUS153 logic diagram is rotated 90 degrees, the relationship to the program table should be obvious.

A section of an adder example is shown below.

```
------LH----------A.AAAAAAAA
------HL----------A.AAAAAAAA
------HH----------.AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
```

In this example outputs B(9) and B(8) are used. The remaining outputs are not used. The first and second product terms are connected to output B(9) and the third is connected to B(8). Notice that these product terms are also connected to all unused outputs. This does not cause any problems because of the 3-State control fusing as shown by the following segment from the end of the same PLUS153 program table.

```
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
----------------
----------------
HHLLLLLLLL
```

FIXED-OR architecture devices have their 3-State output control product term distributed between sections of OR groupings. PLA devices have control product terms displayed at the end of their program table. The outputs B(9) and B(8) have their 3-State output buffer constantly enabled as shown by the dashed product terms. The other enabling product terms are all zeroed, forcing the unused B pins into their high-impedance 3-State mode.

Leaving product terms connected to unused B pins gives an advantage if additional output pins are added at a later time. If another output is added that requires a product term which is already used by an output, then it may be possible to reprogram the old device with the new pattern.

A minor section of the PLUS153 fusemap occurs as the last line as shown above. It consists of 10 characters which may be either H or L. These characters show the polarity of the outputs. An 'H' denotes no inversion or the output is 'Active-High'. An 'L' denotes an inversion or the output is 'Active-Low'. In the adder example, B(9) and B(8) are both 'Active-High'.

So, just to quickly review the PLUS153 adder example program table, three product terms were programmed. Two for the B(9) output which when read from the fusemap translates to B(9) = I0 * /I1 + /I0 * I1. This is an AND-OR implementation of the XOR function. Output B(8) used one product term of B(8) = I0 * I1. Both outputs are 'Active-High' and constantly enabled.

The following is a PLUS405 program table segment. However, the different sections have been separated by spaces for clarity.

Interpreting the SLICE fusetable

```
LH
LLLLLLLLLLLLLLL
AA ---------------- -----LLL -----HLL HLL-----
AA ---------------- -----HLL -----HHH HHH-----
AA ---------------- -----LHL -----HHL HHL-----
AA ---------------- -----HHL -----HLH HLH-----
AA ---------------- -----LLH -----LLL LLL-----
AA ---------------- -----HLH -----LLH LLH-----
AA ---------------- -----LHH -----LHL LHL-----
AA ---------------- -----HHH -----LHH LHH-----
00 0000000000000000 00000000 00000000 00000000
00 0000000000000000 00000000 00000000 00000000
|| |               | |      | |      | |      |
01 2              17 18    25 26    33 34    41
```

The first row indicates the fusing configuration for the CLK1/CLK2 and INIT/OE options. The second row displays the flip-flop PRESET/RESET option for each flip-flop. In this example, all flip-flops will be reset upon an INIT signal on pin 19.

The next row is a representation of a product term within the PLUS405 and it's connections to the OR arrays. Like the PLUS153 it consists of two main sections. The product term array (columns 0 - 25) and the OR array (26 - 41). Ignoring columns 0 and 1 for the moment, columns 2 through 25 display the fusing of input buffers connected to the product term. It is exactly like the combinatorial devices we have seen so far. Columns 2 through 17 show the dedicated input buffer connection fusing and 18 through 25 display the feedback from the internal buried (not connected to pins) flip-flops. In this example, the output of the first product term will be HIGH when the outputs of flip-flops P2, P1 and P0 are all LOW.

The OR array differs in it's representation as compared to a device without JK or SR flip-flops because of the two inputs required for JK flip-flops. As the input buffer to product term connections consists of two buffer output lines being fused to a single product term input, the PLUS405 OR array section consists of one product term output being fused to two OR gate inputs - one or the J and one for the K flip-flop input. Four characters are required to represent the two fuses. The same characters are used as in the product term section. The PLUS405's OR array section is still read vertically for a single output but it can also be read horizontally to determine the next state. In the above program table, columns 26 - 33 are inputs to P7 - P0 (the internal flip-flops). Columns 34 - 41 are inputs to F7 - F0 which are the output flip-flops.

A zero (0) is also the default virgin state condition of the OR array section. It represents a connection of the product term output to both the J and K inputs. Unlike a zero in the product term section, a zero may be found alone on the line. It is used to toggle the flip-flop upon a clock. Counters may be efficiently constructed using this feature.

A dash (–) indicates that the product term does not connect to either the J or K. The flip-flop will remain in it's current state while being clocked.

An "H" indicates the product term connects to only the J input of the flip-flop. If the product term is HIGH and after a clock occurs, the flip-flop output will be HIGH.

An "L" indicates the product term connects to only the K input of the flip-flop. If the product term is HIGH and after a clock occurs, the flip-flop output will be LOW.

Sometimes columns 18 - 25 are called the "present state" inputs because they come directly from the buried flip-flops. Columns 26 - 33 are call the "next state" outputs as they connect to the buried flip-flops inputs and control to which state the flip-flops will transition.

Looking at the above program table we can determine into which state this machine will go given a present state. Assuming that an INIT pulse occurred, all flip-flops will be LOW (this was read from the second line). This will make the very first product term (third line) active or HIGH. Upon the rising edge of a clock pulse, the state will change to HLL (P2-P0) and the output also will change to HLL (F7-F5). The second product term (line four) will become HIGH and force a jump to HHH upon the next clock. This will cause the product term of line 10 to become active and force a jump to LHH. You should be able to follow the state machine from here back to state LLL.

Although this state machine did not use any of the direct inputs, adding a pattern to a product term would be quite easy. It would cause this state machine to wait in a state until the particular input pattern occurred on the input pins before proceeding to the next state.

Columns 0 and 1 show the fusing of the two complement arrays. A complement array is actually a NOR gate whose input is connectable to any or all product term outputs and whose input is connectable to any or all of the product terms input. It can be used as an illegal state detector, forcing a jump to a known state, or as an "ELSE" jump generator

The complement array's column is a little different from what we've seen so far. This is because the complement array input spans all of the product terms output (like a regular OR gate) but it's output is fed back into the product term input's section. Two fuses are represented by each character in the complement array column. A zero (0) indicates both the output of the product term is connected to the input of the complement array and the output of the complement array is connected to the product term's input. This condition is not used for a product term that has other inputs programmed because it could cause oscillations.

A dash (–) indicates neither the output of the product term is connected to the complement array or the output of the complement array is connected to the input of the product term.

An "A" denotes connection only between the product term output and the input of the complement array. A dot (.) denotes a connection only between the complement array output and the input of the product term. Graphical representations of these connection options are listed in the PLUS405 data sheet.

A portion of a PLHS501 program table is shown below:

Interpreting the SLICE fusetable

```
HHHH
--------------------------------.................................................................
H-L-----------------------------.................................................................
--------------------------------..A..............................................................
--------------------------------AA.............................................................
--------------------------------.................................................................
```

The first line indicates the state of the fuses enabling outputs B3-B0. An "H" indicates the 3-State output buffer is enabled. An "L" indicates the specific B pin's output buffer is permanently in the high-impedance mode.

The remaining lines of the PLHS501 program table indicate how the inputs to each NAND gate are configured. In the above representation, the dashes and dots occur on different lines. While viewing the program table with SLICE, the 32 dashes and 72 dots occur on one line. Cursor control keys may be used to scroll the screen horizontally and vertically. The upper portion of the program table display inputs to NAND gate that connect to output buffers followed by the 72 foldback NAND gates. The upper right status box of SLICE's program table editor always displays the name of the NAND gate currently under the cursor as well as which input is at the cursor intersection.

Each NAND gate of the PLHS501 has as it's inputs 24 dedicated input buffer outputs, 8 bidirectional pins feedback buffer outputs, and the outputs of 72 foldback NAND gates! That adds up to a 104 input NAND gate and the PLHS501 has 120 of them! The dashed portion of the line represents the 24 input and 8 bidirectional pins feedback buffer connections. Valid entries are H, L, –, and 0. The meanings are exactly like the inputs to product terms described in the previous device examples.

The remaining portion of the line contains dots (.) or "A" characters. This represents no connection or a connection to a foldback NAND gate's output. The specific NAND gate's name will be listed in the status box of the editor when the cursor is under an "A" character.

In the above example, the first NAND gate does not have anything connected to it. The second has I0 (pin 41) and I2 (pin 43). The output of this NAND gate will go LOW only when I0 is HIGH and I2 is LOW. The third NAND gate doesn't have any pin buffer inputs, but it does have the output of a foldback NAND gate (FB2) as an input. The fourth NAND gate has two foldback NAND gate outputs as inputs. They are FB0 and FB1. The fourth NAND gate's output will only go active LOW when both outputs of FB0 and FB1 are HIGH.

The fusetable or program table for a PLUS153 is actually quite simple. It is divided into two main sections which represent the AND array and the OR array sections of the actual device. These sections contain characters which represent the fusing connections of the arrays.

In a PLUS153, there are 18 buffers feeding into the inputs of 42 AND gates. Eight of these buffers are from input only pins (I0-I7) while the remaining 10 are from the bidirectional pins (B0-B9). Each buffer has a non-inverting and inverting output. Therefore, there are four possible output connections from each buffer to the input of an AND gate.

Notice in the program table the H,L,– and 0 characters which are located left of the 'A' and dot (.) characters. These characters represent the four possible input buffer connections to product terms (AND gates) and are read horizontally. There are 18 columns in this section, one for each buffer and there are 42 rows, one for each product term.

A zero (0) is the default or virgin state designator for the connection between a buffer and product term of the PLUS153. It indicates that both the non-inverting and inverting outputs from the buffer are connected to a product term's input. This has the effect of holding the product term's input and thus it's output LOW. A zero is usually not found on a row alone. To hold a product term low, it is better to connect all inputs of a product term to all buffer outputs. This is because the two paths through a buffer do not have equal delays. If only one buffer was used to hold a product term LOW, any signal on the input of the buffer may cause a glitch on the output of the product term.

A dash (–) is the opposite of a zero. It denotes neither the non-inverting or inverting output of a buffer is connected to the product term. If all of the inputs to a product term are dashed, then an internal pull-up guarantees that the output of the product term will be HIGH. This used frequently for product terms controlling the PLUS153's 3-State output buffers.

The H and L characters denote respectively a connection between only the non-inverting and inverting buffer outputs and the product term's input. Reading the H and L characters on a specific product term's row almost makes the device appear to be many comparators. For example, if a product term had "————H–L————————" listed as it's inputs, then the output will only be HIGH when the inputs match that pattern. In other words, the output will only be HIGH when I0 is LOW AND I2 is HIGH.

The OR gate input connections are represented by the other half of the program table. There are 10 columns, one for each output and 24 rows, one for each possible product term connection (the other 10 product terms control the 3-State output buffers). We noted in the preceding paragraphs that each input buffer had both inverting and non-inverting outputs and thus four characters were required to describe possible connections. The AND gates only have only one output and therefore only 2 characters are needed the represent a connection and lack thereof. This is denoted by an A and a dot (.) respectively. Also, this section is read vertically, NOT horizontally as the product term inputs. If a PLUS153 logic diagram is rotated 90 degrees, the relationship to the program table should be obvious.

For example, a section of the adder example is shown below.

```
------LH----------A.AAAAAAAA
------HL----------A.AAAAAAAA
------HH----------.AAAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
```

In this example the outputs B(9) and B(8) were named SUM and CY. The remaining outputs were not used. The first and second product terms are connected to SUM and the third is connected to CY. Notice that these product terms are also connected to all unused outputs. This does not cause any problems because of the 3-State control fusing as shown by the following segment from the end of the program table.

Interpreting the SLICE fusetable

```
00000000000000000..AAAAAAAA
00000000000000000..AAAAAAAA
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
----------------
----------------
HHLLLLLLLL
```

The outputs SUM and CY have their 3-State output buffer constantly enabled as shown by the dashed product terms. The other enabling product terms are all zeroed, forcing the unused B pins into their high-impedance 3-State mode.

Leaving product terms connected to unused B pins gives an advantage if additional output pins are added at a later time. The PLA architecture of the PLUS153 allows product terms to be shared between two or more outputs. If another output is added that requires a product term which is already used by an output, then it may be possible to reprogram the old device with the new pattern.

A minor section of the PLUS153 program table occurs as the last line as shown above. It consists of 10 characters which may be either H or L. These characters show the polarity of the outputs. An 'H' denotes no inversion or the output is 'Active-High'. An 'L' denotes an inversion or the output is 'Active-Low'. In the adder example, SUM and CY are both 'Active-High'.

So, just to quickly review the adder example program table, three product terms were programmed. Two for the SUM output which when read from the fusemap translates to SUM = I0 * /I1 + /I0 * I1. This is an AND-OR implementation of the XOR function. Output CY used one product term of CY = I0 * I1. Both outputs were 'active-high' and constantly enabled.

Third-party software support

Programmable Logic Devices

Data I/O Corporation
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA 98073–9746
Telephone: (206) 881-6444

PART NUMBER	TYPE	PACKAGE	# PINS	DEVICE FILE	ABEL REV.
10020EV8	FPLA	DIP	24	EC20EV8A	2.1
10H20EV8	FPLA	DIP	24	EC20EV8A	2.1
10H20EV8	FPLA	PLCC	28	EC20EV8A	2.1
CK2605	FPLA	DIP	20	F2605	2.0
CK2678	FPLA	DIP	20	F2678	3.1
PHD16N8	FPLA	DIP	20	P16N8	3.0
PHD16N8	FPLA	PLCC	20	P16N8	3.0
PHD48N22	PHD	PLCC	68	P48N22	4.0
PLC105	FPLS	DIP	28	F105	1.0
PLC153	FPLA	DIP	20	F153	1.0
PLC153	FPLA	PLCC	20	F153	1.0
PLC159	FPLA	DIP	20	F159	1.0
PLC16V8	FPLA	DIP	20	P16SV8	3.1
PLC16V8	FPLA	PLCC	20	P16SV8	3.1
PLC18V8Z	EPLD	DIP	20	P18V8Z	3.2
PLC18V8Z	EPLD	PLCC	20	P18V8Z	3.2
PLC20V8	FPLA	DIP	24	P20SV8	3.1
PLC20V8	FPLA	PLCC	28	P20SV8	3.1
PLC415	FPLA	DIP	28	F415	3.2
PLC415	FPLA	PLCC	28	F415	3.2
PLC42VA12	FPLA	DIP	24	F42VA12	3.2
PLC42VA12	FPLA	PLCC	28	F42VA12	3.2
PLC473	FPLA	DIP	24	F473	2.1
PLC473	FPLA	PLCC	28	F473	2.1
PLHS153	FPLA	DIP	20	F153	1.0
PLHS153	FPLA	PLCC	20	F153	1.0
PLHS16L8	FPLA	DIP	20	P16L8	1.0
PLHS16L8	FPLA	PLCC	20	P16L8	1.0
PLHS18P8	FPLA	DIP	20	P18P8	1.0
PLHS18P8	FPLA	PLCC	20	P18P8	1.0
PLHS473	FPLA	DIP	24	F473	2.1
PLHS473	FPLA	PLCC	28	F473	2.1
PLHS501	FPLA	PLCC	52	PML501	2.1
PLHS601	PLHS	PLCC	68	F601	4.0
PLHS601	PLHS	PLCC	68	F601	4.0
PLS100	FPLA	DIP	28	F100	1.0
PLS100	FPLA	PLCC	28	F100	1.0
PLS101	FPLA	DIP	28	F100	1.0
PLS101	FPLA	PLCC	28	F100	1.0
PLS102	FPGA	DIP	28	F103	1.0
PLS103	FPGA	DIP	28	F103	1.0
PLS103	FPGA	PLCC	28	F103	1.0
PLS104	FPLS	DIP	28	F105	1.0
PLS105	FPLS	DIP	28	F105	1.0
PLS105	FPLS	PLCC	28	F105	1.0
PLS150	FPGA	DIP	20	F151	2.1
PLS151	FPGA	DIP	20	F151	2.1
PLS151	FPGA	PLCC	20	F151	2.1
PLS152	FPLA	DIP	20	F153	1.0
PLS152	FPLA	PLCC	20	F153	1.0

Third-party software support

Data I/O Corporation (CONTINUED)

PART NUMBER	TYPE	PACKAGE	# PINS	DEVICE FILE	ABEL REV.
PLS153	FPLA	DIP	20	F153	1.0
PLS153	FPLA	PLCC	20	F153	1.0
PLS154	FPLS	DIP	20	F155	2.1
PLS155	FPLS	DIP	20	F155	2.1
PLS155	FPLS	PLCC	20	F155	2.1
PLS156	FPLS	DIP	20	F157	1.0
PLS157	FPLS	DIP	20	F157	1.0
PLS157	FPLS	PLCC	20	F157	1.0
PLS158	FPLS	DIP	20	F159	1.0
PLS159	FPLS	DIP	20	F159	1.0
PLS159	FPLS	PLCC	20	F159	1.0
PLS159A	FPLS	DIP	20	F159	1.0
PLS159A	FPLS	PLCC	20	F159	1.0
PLS161	FPLA	DIP	24	F161	1.0
PLS162	FPLA	DIP	24	F162	2.1
PLS163	FPLA	DIP	24	F163	2.0
PLS166	FPLS	DIP	24	F167	1.0
PLS167	FPLS	DIP	24	F167	1.0
PLS167	FPLS	PLCC	28	F167	1.0
PLS168	FPLS	DIP	24	F168	2.0
PLS168	FPLS	PLCC	28	F168	2.0
PLS173	FPLA	DIP	24	F173	2.0
PLS173	FPLA	PLCC	28	F173	2.0
PLS179	FPLS	DIP	24	F179	2.1
PLS179	FPLS	PLCC	28	F179	2.1
PLUS105–40	FPLS	DIP	28	F105	1.0
PLUS105–40	FPLS	PLCC	28	F105	1.0
PLUS153	FPLA	DIP	20	F153	1.0
PLUS153	FPLA	PLCC	20	F153	1.0
PLUS153B	FPLA	DIP	20	F153	1.0
PLUS16L8	FPLA	DIP	20	P16L8	1.0
PLUS16L8	FPLA	PLCC	20	P16L8	1.0
PLUS16R4	FPLA	DIP	20	P16R4	1.0
PLUS16R4	FPLA	PLCC	20	P16R4	1.0
PLUS16R6	FPLA	DIP	20	P16R6	1.0
PLUS16R6	FPLA	PLCC	20	P16R6	1.0
PLUS16R8	FPLA	DIP	20	P16R8	1.0
PLUS16R8	FPLA	PLCC	20	P16R8	1.0
PLUS173	FPLS	DIP	24	F173	2.0
PLUS173	FPLS	PLCC	28	F173	2.0
PLUS20L8	FPLA	DIP	24	P20L8	1.0
PLUS20L8	FPLA	PLCC	28	P20L8C	3.1
PLUS20R4	FPLA	DIP	24	P20R4	1.0
PLUS20R4	FPLA	PLCC	28	P20R4C	3.1
PLUS20R6	FPLA	DIP	24	P20R6	1.0
PLUS20R6	FPLA	PLCC	28	P20R6C	3.1
PLUS20R8	FPLA	DIP	24	P20R8	1.0
PLUS20R8	FPLA	PLCC	28	P20R8C	3.1
PLUS405	FPLS	DIP	28	F405	3.0
PLUS405	FPLS	PLCC	28	F405	3.0
PML2552	FPLA	PLCC	68	F2552	4.0

Third-party software support

Logical Devices, Inc.
1201 N.W. 65th Place
Ft. Laudedale, FL 33309
Telephone: (305) 974-0967

PART NUMBER	DEVICE MNEOMONIC	# PINS	# FUSES	# OF P-TERMS	CUPL VER. 4.0A REC. VER. LEVEL
10020EV8	P1020EV8	24	3616	80	3.0a
10H20EV8	P1020EV8	24	3616	80	3.0a
82S100	F100	28	1928	48	2.00a
82S101	F100	28	1928	48	2.00a
82S103	F103	28	297	9	2.00a
82S105	F105	28	3553	48	2.00a
82S105A	F105	28	3553	48	2.00a
82S151	F151	20	564	18	2.10a
82S153	F153	20	1842	42	2.15a
82S153A	F153	20	1842	42	2.15a
82S155	F155	20	2108	43	2.00a
82S157	F157	20	2108	43	2.00a
82S159	F159	20	2108	43	2.00a
82S161	F161	24	1544	48	2.00a
82S162	F162	24	165	5	2.00a
82S163	F163	24	225	9	2.00a
82S167	F167	24	3361	48	2.00a
82S167A	F167	24	3361	48	2.00a
82S168	F168	24	3553	48	2.10a
82S173	F173	24	2178	42	2.10a
82S179	F179	24	2452	43	3.0a
PHD16N8	P16N8	20	512	16	2.50a
PHD48N22	F48N22	68	7008	73	4.0a
PLC153	F153	20	1842	42	2.15a
PLC16V8	F16V8	20	2617	72	3.2b
PLC18V8Z	F18V8Z	20	2689	72	4.0a
PLC20V8	F20V8	24	3193	72	4.0a
PLC415	F415	28	5751	68	4.0a
PLC42VA12	F42VA12	24	8994	10	3.2b
PLC473	F473	24	1499	24	2.15a
PLHS153	F153	20	1842	42	2.15a
PLHS16L8A	P16L8	20	2048	64	1.01a
PLHS16L8B	P16L8	20	2048	64	1.01a
PLHS18P8A	P18P8	20	2600	72	2.01a
PLHS18P8B	P18P8	20	2600	72	2.01a
PLHS473	F473	24	1499	24	2.15a
PLHS501	F501	52	15780	112	3.2b
PLHS502	F502	64	23464	144	3.2a
PLS100	F100	28	1928	48	2.00a
PLS101	F100	28	1928	48	2.00a
PLS103	F103	28	297	9	2.00a
PLS105	F105	28	3553	48	2.00a
PLS105A	F105	28	3553	48	2.00a
PLS151	F151	20	564	16	2.10a
PLS152	F153	20	1842	42	2.15a
PLS153	F153	20	1842	42	2.15a
PLS153A	F153	20	1842	42	2.15a
PLS155	F155	20	2108	43	2.00a
PLS157	F157	20	2108	43	2.00a
PLS159	F159	20	2108	43	2.00a
PLS159A	F159	20	2108	43	2.00a
PLS161	F161	24	1544	44	2.00a
PLS162	F162	24	165	8	2.00a

Third-party software support

Logical Devices, Inc. (CONTINUED)

PART NUMBER	DEVICE MNEOMONIC	# PINS	# FUSES	# OF P-TERMS	CUPL VER. 4.0A REC. VER. LEVEL
PLS163	F163	24	225	9	2.00a
PLS167	F167	24	3361	48	2.00a
PLS167A	F167	24	3361	48	2.00a
PLS168	F168	24	3553	48	2.10a
PLS168A	F168	24	3553	48	2.10a
PLS173	F173	24	2178	42	2.10a
PLS179	F179	24	2452	43	3.0a
PLUS105	F105	28	3553	48	2.00a
PLUS153B/D	F153	20	1842	42	2.15a
PLUS16L8D/–7	F16L8	20	2048	64	1.01a
PLUS16R4D/–7	P16R4	20	2048	64	1.01a
PLUS16R6D/–7	P16R6	20	2048	64	1.01a
PLUS16R8D/–7	P16R8	20	2048	64	1.01a
PLUS173B/D	P173	24	2178	42	2.10a
PLUS20L8D/–7	P20L8	24	2560	64	2.00a
PLUS20R4D/–7	P20R4	24	2560	64	2.00a
PLUS20R6D/–7	P20R6	24	2560	64	2.00a
PLUS20R8D/–7	P20R8	24	2560	64	2.00a
PLUS405	F405	28	5410	64	3.0a

Third-party software support

Minc, Incorporated
6755 Earl Drive
Colorado Springs, CO 80918
Telephone: (719) 590-1155

PART NUMBER	TEMPLATE NAME	TECHNOLOGY	PACKAGE	TEMPERATURE
82S100B3A	A100	TTL	LCC	MIL
82S100BXA	A100	TTL	CDIP	MIL
82S100BYA	A100	TTL	SM	MIL
82S101B3A	A100	TTL	LCC	MIL
82S101BXA	A100	TTL	CDIP	MIL
82S101BYA	A100	TTL	SM	MIL
PLS100A	A100	TTL	PLCC	COM
PLS100F	A100	TTL	CDIP	COM
PLS100N	A100	TTL	DIP	COM
PLS101A	A100	TTL	PLCC	COM
PLS101F	A100	TTL	CDIP	COM
PLS101N	A100	TTL	DIP	COM
82S153AB2A	A153	TTL	LCC	MIL
82S153ABRA	A153	TTL	CDIP	MIL
82S153ABSA	A153	TTL	SM	MIL
PLC153–45A	A153	CMOS	PLCC	COM
PLC153–45FA	A153	CMOS	CDIP	COM
PLC153–45N	A153	CMOS	DIP	COM
PLC153–60A	A153	CMOS	PLCC	COM
PLC153–60FA	A153	CMOS	CDIP	COM
PLC153–60N	A153	CMOS	DIP	COM
PLHS153A	A153	TTL	PLCC	COM
PLHS153N	A153	TTL	DIP	COM
PLS153A	A153	TTL	PLCC	COM
PLS153AA	A153	TTL	PLCC	COM
PLS153AF	A153	TTL	CDIP	COM
PLS153AN	A153	TTL	DIP	COM
PLS153F	A153	TTL	CDIP	COM
PLS153N	A153	TTL	DIP	COM
PLUS153BA	A153	TTL	PLCC	COM
PLUS153BN	A153	TTL	DIP	COM
PLUS153DA	A153	TTL	PLCC	COM
PLUS153DN	A153	TTL	DIP	COM
PLS173	A173	TTL	PLCC	COM
PLS173BLA	A173	TTL	CDIP	MIL
PLS173F	A173	TTL	CDIP	COM
PLS173N	A173	TTL	DIP	COM
PLUS173BA	A173	TTL	PLCC	COM
PLUS173BN	A173	TTL	DIP	COM
PLUS173DA	A173	TTL	PLCC	COM
PLUS173DN	A173	TTL	DIP	COM
PLC473–60A	A473	CMOS	PLCC	COM
PLC473–60FA	A473	CMOS	CDIP	COM
PLC473–60N	A473	CMOS	DIP	COM
PLHS473A	A473	TTL	PLCC	COM
PLHS473F	A473	TTL	CDIP	COM
PLHS473N	A473	TTL	DIP	COM
10020EV8–4F	P10H20EV8	ECL	CDIP	COM
10H20EV8–4F	P10H20EV8	ECL	CDIP	COM

Third-party software support

Minc, Incorporated (CONTINUED)

PART NUMBER	TEMPLATE NAME	TECHNOLOGY	PACKAGE	TEMPERATURE
PLHS16L8AA	P16L8	CMOS	PLCC	COM
PLHS16L8AN	P16L8	CMOS	DIP	COM
PLHS16L8BA	P16L8	CMOS	PLCC	COM
PLHS16L8BN	P16L8	CMOS	DIP	COM
PLUS16L8–7A	P16L8	CMOS	PLCC	COM
PLUS16L8–7N	P16L8	CMOS	DIP	COM
PLUS16L8DA	P16L8	CMOS	PLCC	COM
PLUS16L8DN	P16L8	CMOS	DIP	COM
PHD16N8–5A	P16N8	TTL	PLCC	COM
PHD16N8–5N	P16N8	TTL	DIP	COM
PLUS16R4–7A	P16R4	CMOS	PLCC	COM
PLUS16R4–7N	P16R4	CMOS	DIP	COM
PLUS16R4DA	P16R4	CMOS	PLCC	COM
PLUS16R4DN	P16R4	CMOS	DIP	COM
PLUS16R6–7A	P16R6	CMOS	PLCC	COM
PLUS16R6–7N	P16R6	CMOS	DIP	COM
PLUS16R6DA	P16R6	CMOS	PLCC	COM
PLUS16R6DN	P16R6	CMOS	DIP	COM
PLUS16R8–7A	P16R8	CMOS	PLCC	COM
PLUS16R8–7N	P16R8	CMOS	DIP	COM
PLUS16R8DA	P16R8	CMOS	PLCC	COM
PLUS16R8DN	P16R8	CMOS	DIP	COM
PLC16V8H35A	P16V8S	CMOS	PLCC	COM
PLC16V8H35FA	P16V8S	CMOS	CDIP	COM
PLC16V8H35N	P16V8S	CMOS	DIP	COM
PLC16V8H45A	P16V8S	CMOS	PLCC	COM
PLC16V8H45FA	P16V8S	CMOS	CDIP	COM
PLC16V8H45N	P16V8S	CMOS	DIP	COM
PLC16V8Q35A	P16V8S	CMOS	PLCC	COM
PLC16V8Q35FA	P16V8S	CMOS	CDIP	COM
PLC16V8Q35N	P16V8S	CMOS	DIP	COM
PLC16V8Q45A	P16V8S	CMOS	PLCC	COM
PLC16V8Q45FA	P16V8S	CMOS	CDIP	COM
PLC16V8Q45N	P16V8S	CMOS	DIP	COM
PLHS18P8AA	P18P8	TTL	PLCC	COM
PLHS18P8AN	P18P8	TTL	DIP	COM
PLHS18P8BA	P18P8	TTL	PLCC	COM
PLHS18P8BN	P18P8	TTL	DIP	COM
PLC18V8Z35A	P18V8S	CMOS	PLCC	COM
PLC18V8Z35FA	P18V8S	CMOS	CDIP	COM
PLC18V8Z35N	P18V8S	CMOS	DIP	COM
PLC18V8ZIA	P18V8S	CMOS	PLCC	EXT
PLC18V8ZIFA	P18V8S	CMOS	CDIP	EXT
PLC18V8ZIN	P18V8S	CMOS	DIP	EXT
PLUS20L8–7A	P20L8	CMOS	PLCC	COM
PLUS20L8–7N	P20L8	CMOS	DIP	COM
PLUS20L8DA	P20L8	CMOS	PLCC	COM
PLUS20L8DN	P20L8	CMOS	DIP	COM
PLUS20R4–7A	P20R4	CMOS	PLCC	COM
PLUS20R4–7N	P20R4	CMOS	DIP	COM
PLUS20R4DA	P20R4	CMOS	PLCC	COM
PLUS20R4DN	P20R4	CMOS	DIP	COM

Third-party software support

Minc, Incorporated (CONTINUED)

PART NUMBER	TEMPLATE NAME	TECHNOLOGY	PACKAGE	TEMPERATURE
PLUS20R6–7A	P20R6	CMOS	PLCC	COM
PLUS20R6–7N	P20R6	CMOS	DIP	COM
PLUS20R6DA	P20R6	CMOS	PLCC	COM
PLUS20R6DN	P20R6	CMOS	DIP	COM
PLUS20R8–7A	P20R8	CMOS	PLCC	COM
PLUS20R8–7N	P20R8	CMOS	DIP	COM
PLUS20R8DA	P20R8	CMOS	PLCC	COM
PLUS20R8DN	P20R8	CMOS	DIP	COM
PLC20V8H35A	P20V8S	CMOS	PLCC	COM
PLC20V8H35FA	P20V8S	CMOS	CDIP	COM
PLC20V8H35N	P20V8S	CMOS	DIP	COM
PLC20V8H45A	P20V8S	CMOS	PLCC	COM
PLC20V8H45FA	P20V8S	CMOS	CDIP	COM
PLC20V8H45N	P20V8S	CMOS	DIP	COM
PLC20V8Q35A	P20V8S	CMOS	PLCC	COM
PLC20V8Q35FA	P20V8S	CMOS	CDIP	COM
PLC20V8Q35N	P20V8S	CMOS	DIP	COM
PLC20V8Q45A	P20V8S	CMOS	PLCC	COM
PLC20V8Q45FA	P20V8S	CMOS	CDIP	COM
PLC20V8Q45N	P20V8S	CMOS	DIP	COM
82S105B3A	S105	TTL	LCC	MIL
82S105BXA	S105	TTL	CDIP	MIL
82S105BYA	S105	TTL	SM	MIL
PLS105A	S105	TTL	PLCC	COM
PLS105AA	S105	TTL	PLCC	COM
PLS105AF	S105	TTL	CDIP	COM
PLS105AN	S105	TTL	DIP	COM
PLS105F	S105	TTL	CDIP	COM
PLS105N	S105	TTL	DIP	COM
PLS155A	S155	TTL	PLCC	COM
PLS155F	S155	TTL	CDIP	COM
PLS155N	S155	TTL	DIP	COM
PLS157A	S157	TTL	PLCC	COM
PLS157F	S157	TTL	CDIP	COM
PLS157N	S157	TTL	DIP	COM
PLS159AA	S159	TTL	PLCC	COM
PLS159AN	S159	TTL	DIP	COM
PLS167A	S167	TTL	PLCC	COM
PLS167AA	S167	TTL	PLCC	COM
PLS167AN	S167	TTL	DIP	COM
PLS167BLA	S167	TTL	CDIP	MIL
PLS167N	S167	TTL	DIP	COM
PLS168A	S168	TTL	PLCC	COM
PLS168AA	S168	TTL	PLCC	COM
PLS168AN	S168	TTL	DIP	COM
PLS168BLA	S168	TTL	CDIP	MIL
PLS168N	S168	TTL	DIP	COM
PLS179A	S179	TTL	PLCC	COM
PLS179N	S179	TTL	DIP	COM
PLUS405–37A	S405	TTL	PLCC	COM
PLUS405–37N	S405	TTL	DIP	COM
PLUS405–45A	S405	TTL	PLCC	COM
PLUS405–45N	S405	TTL	DIP	COM

Section 9
Application Notes

Programmable Logic Devices

INDEX

Programmable logic design and application notes

Programmable Logic Devices

Today's engineer is constantly striving to consolidate higher complexity and more feature-intensive circuits into designs without sacrificing flexibility. In a competitive marketplace, designs need to be brought to market quickly. The Signetics solution is to provide high-performance Programmable Logic Devices (PLDs) that can be quickly and easily integrated into system designs.

In using this manual, some familiarity with PLDs is helpful. In addition, we recommend the recently published text books, Programmable Logic Devices, by Geoff Bostock (McGraw-Hill, copyright 1988), and Programmable Designer's Guide by Roger Alford (Howard W. Sams & Co., copyright 1989).

This document provides complete, straightforward application examples. The first three sections describe Signetics PLDs. Sections four through eight provide application examples. Most applications are accompanied by one or two pages of text. Some also include a circuit or block diagram and an AMAZE design file listing to implement that application. To save time, the files are available on diskette or by accessing the Signetics toll-free bulletin board: (800) 451-6644.

ACKNOWLEDGEMENTS

In appreciation for their contributions:

Dimitrious Douros
Mike Gershowitz
Ali Gheissari
Jim Greene
Neil Kellet
Bob Kelley
Jerry Liebler
Bob Lundeberg
John McNally
Tony Parisi

AMAZE SOFTWARE

Design, simulation and device-programming support for all Signetics PLD families is provided by Signetics AMAZE PLD design software. AMAZE, which supports many of Signetics programmable products, includes Boolean logic and direct state-equation entry. Functional and AC timing simulation models and an automatic test vector generator are included in the AMAZE PLD design package. The software runs with MS-DOS 2.0 or newer operating systems and is available free of charge to qualified users. Schematic capture capability is available in conjunction with Data I/O's FutureNet Dash System and the OrCad Systems Corporation OrCAD/SDT™ schematic capture software packages.

The Signetics families of PLDs are also supported by Data I/O®'s design software package, ABEL™, and the P-CAD and CUPL™ design software.

Automatic Map and Zap Equation Entry (AMAZE) software designed by Signetics will interface with most commercial programmers.

AMAZE consists of five modules:

- BLAST (Boolean Logic and State Transfer entry program)
- PTE (Program Table Editor)
- PTP (PAL to PLD conversion program)
- DPI (Device Programmer Interface program)
- PLD SIM (PLD Simulator program)

The software modules allow expansion for future requirements. They are user friendly with both HELP and ERROR messages. Simulator programs provide applications assistance and automatic test vector generation.

Equipment requirements, products supported and details of product modules are contained in the AMAZE design software manual.

AMAZE is available at no charge by request to the Signetics factory.

Additional design support is usually available with the commercially available packages ABEL, CUPL PLDesigner, LOG/iC, etc. Programming support is always available through DATA I/O, STAG Microsystems, and several other programmers.

PRODUCT SECTION INTRODUCTION

Signetics manufactures a wide range of PLDs and Programmable Logic Sequencers. In the area of PLDs, there are two basic architectures: Programmable Logic Arrays (PLAs) and Programmable Array Logic (PAL®).

The PLA architecture consists of two interconnectable arrays with programmable connections between the input pins and a group of AND gates. Another programmable array exists between the AND gate outputs and the inputs to a group of OR-gates. Complete freedom of interconnection is possible with this arrangement.

PAL-type devices, on the other hand, provide programmable interconnection between the input pins and the AND gates, but the outputs of the AND gates are tied to specific OR gates, then finally routed to output pins. By eliminating the programmability between the AND-OR area, some speed savings are achieved at the expense of interconnect freedom.

Signetics Programmable Logic Sequencers combine the versatility of the PLA with flip-flops to achieve powerful state machines in a variety of user configurations.

This section is designed to familiarize design engineers with the Signetics lines of PLAs, PALs, and Sequencers available, and acquaint them with their general capabilities and features. Each architecture is briefly showcased in an initial rendering with a short capsule description of the part. The serious reader should consult the data sheet sections of this PLD Data Handbook for full electrical details on any part.

Programmable logic design and application notes

Programmable Logic Devices

PLA DEVICES

Signetics PLAs are particularly useful in the design of wide address decoders and random logic replacement. The primary advantage Signetics brings to these applications with their PLA devices is product term sharing, which is made possible via the two programmable arrays. The familiar PAL architecture supports a programmable AND array, followed by a fixed OR array. Better than 90% of the PAL devices that are available today are limited to 7 input wide gates. When pursuing a solution to a complex address decoding scheme, this restriction is prohibitive. The Signetics PLA devices support 100% of all product terms. Once a term is created, it can be shared with any or all of the output functions. No duplication of resources is incurred. The popular PLXX153 family support 32-input wide OR gates which are ideal for memory I/O decoders. The addition of programmable output polarity also enhances design efficiency and logic minimization.

The 2 programmable array concept dominates the Signetics PLD product line. With the exception of the PAL-type devices which have been geared for ulitmate performance, all Signetics PLDs have been architected with efficient and flexible PLA structures. With the largest breadth programmable product line in the industry, Signetics believes the designer can truly fill his requirements from the several product lines—PLA, PAL, and PLS. PLA device descriptions follow.

Signetics PLUS153D

Figure 1-1 depicts the Signetics PLUS153D. This bipolar PLA is pin and functionally equivalent to all other Signetics 153 type PLAs (i.e., the PLS153, PLC153, PLHS153), but requires no more than 12 nanoseconds to generate a stable output.

The PLUS153D has eight dedicated inputs and 10 bidirectional pins. The bidirectional pins may be adapted to suit the user's specific needs. 20-pin DIP or PLCC packages are available.

The output structure of the PLUS153D includes programmable polarity control on each output. Either active HIGH (non-inverting) or active LOW (inverting) outputs are configurable via the EX-OR gate associated with each I/O. Individual 3-State control of the I/O is also supported with the ten direction control AND terms (D1-D9).

Other benefits to the PLUS153D include full pin compatibility with most 20-pin combinational PAL® parts. The natural product term sharing capabilities of the PLA architecture yield complete freedom of configuration should the engineer implement a particularly creative decode configuration.

Signetics PLUS173D

Figure 1-2 depicts the Signetics PLUS173D. This bipolar PLA is functionally equivalent to the Signetics PLS173. The 24-pin PLUS173 has four more input pins than the PLUS153. The user may adapt the bidirectional pins to suit particular decoding needs, but the propagation delay time is still no more than 12 nanoseconds from stabilized input to stable output.

By having more inputs than the 153 part, the 173 can either resolve more input lines or generate more outputs functions for the same number of inputs. Distinct 3-State control over each output may be useful for controlling chip enables where contention (i.e., multiple access) may exist.

For speed and input width, the PLUS173D is probably the best single PLA available today for both memory and I/O decoding. Combining the 12 nanosecond t_{PD} with the distinguishable range of 12 to 21 inputs, the designer can easily decode say 16 input addresses as well as read/write qualifiers or encoded status signals. Output polarity control (Active-High or Active-Low) is achieved by programming the Exclusive-OR gate associated with each output.

The flexibility achieved with a PLA structure can be quickly appreciated by the designer who has experienced the frustration of the dedicated "OR" structures in PAL ICs. Currently, the only time penalty for the freedom granted by a PLA is a few nanoseconds!

Programmable logic design and application notes

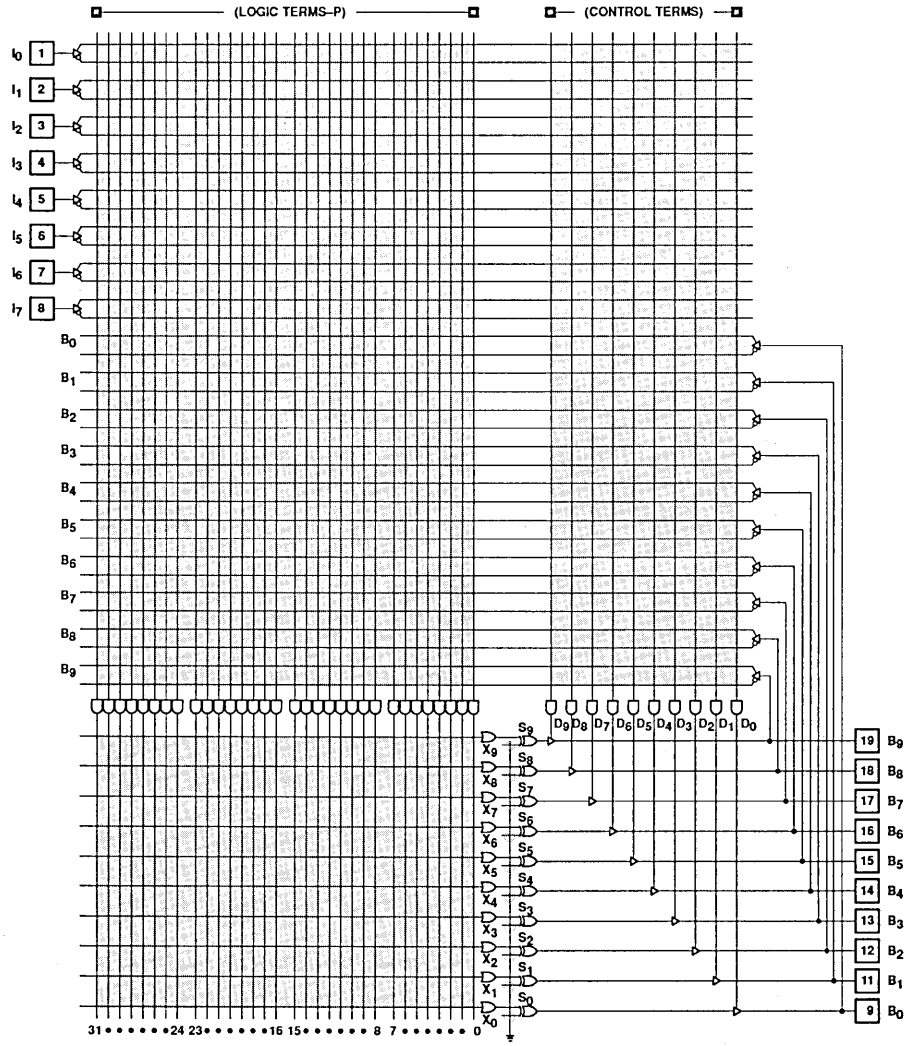


Figure 1–1. Signetics PLUS153D

Programmable logic design and application notes

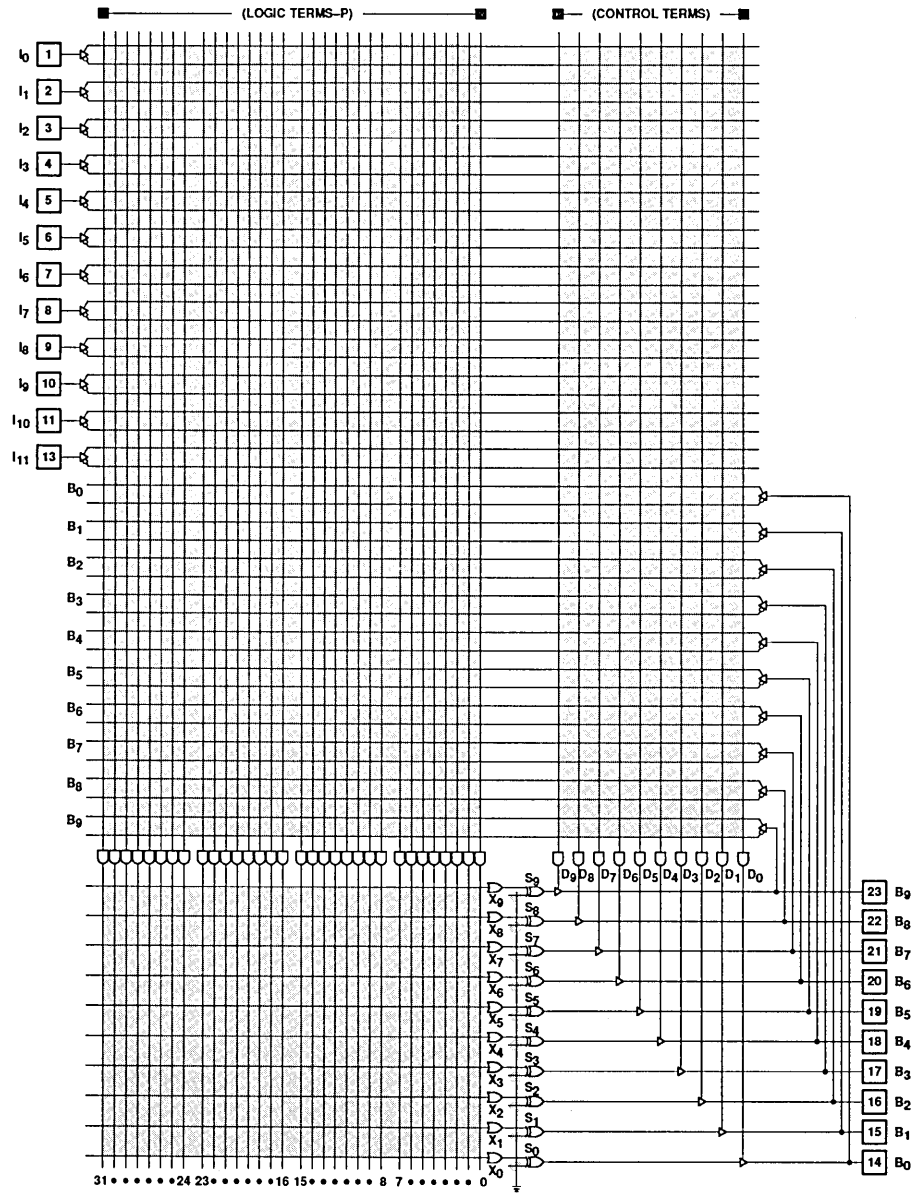


Figure 1–2. Signetics PLUS173D

Programmable logic design and application notes

Programmable Logic Devices

PAL®-TYPE DEVICES

Signetics provides state-of-the-art industry standard PAL devices, both bipolar and CMOS. The range of offers spans the entire gamut of performance options; zero-standby power generic devices specified over the commercial, industrial and military temperature ranges, or the ultimate in high speed, an ECL compatible 20EV8 device. Almost every option in between is also offered.

The PAL architecture consists of a programmable AND array, followed by a fixed OR array. The somewhat rigid architecture lends itself to less complex, narrower logic functions. There are three basic PAL-type device configurations. The XXL8 devices are strictly combinatorial. The XXRX series offers a range of registered and combinatorial outputs.

The XXV8 series is considered to be generic in nature, in that the output macros are variable (hence the "V") as combinatorial or registered. Most frequent applications include counters and shifters (the RX series), and small decoders and multiplexers (the L8 series).

Industry standard software can be used with Signetics PAL-type devices. Full support is also provided via the Signetics AMAZE Design software.

The Signetics PAL-type device descriptions follow. The line is being expanded continuously. If you don't find the device you need for your circuit, please contact Signetics toll-free at (800)227-1817, Extension 900.

Signetics PLHS18P8B

Figure 2–1 depicts the Signetics PLHS18P8B which is a bipolar, PAL-type device. The propagation delay time will be 15 nanoseconds maximum from stable inputs to stable outputs. The part has 10 inputs, eight bidirectional pins, and 72 product terms. Due to the programmable output polarity, the PLHS18P8B can functionally replace 13 other standard PAL devices. Being pin compatible to all 20-pin combinational PALs increases the parts' versatility considerably. The PLHS18P8B can sink I_{OL} = 24mA (max).

Output polarity control for this PAL-type part is achieved identically to the procedure for the PLUS153D and PLUS173D.

The PLHS18P8B is ideal for address and I/O decode for moderately fast microprocessors from both a speed and current drive capability.

Programmable logic design and application notes

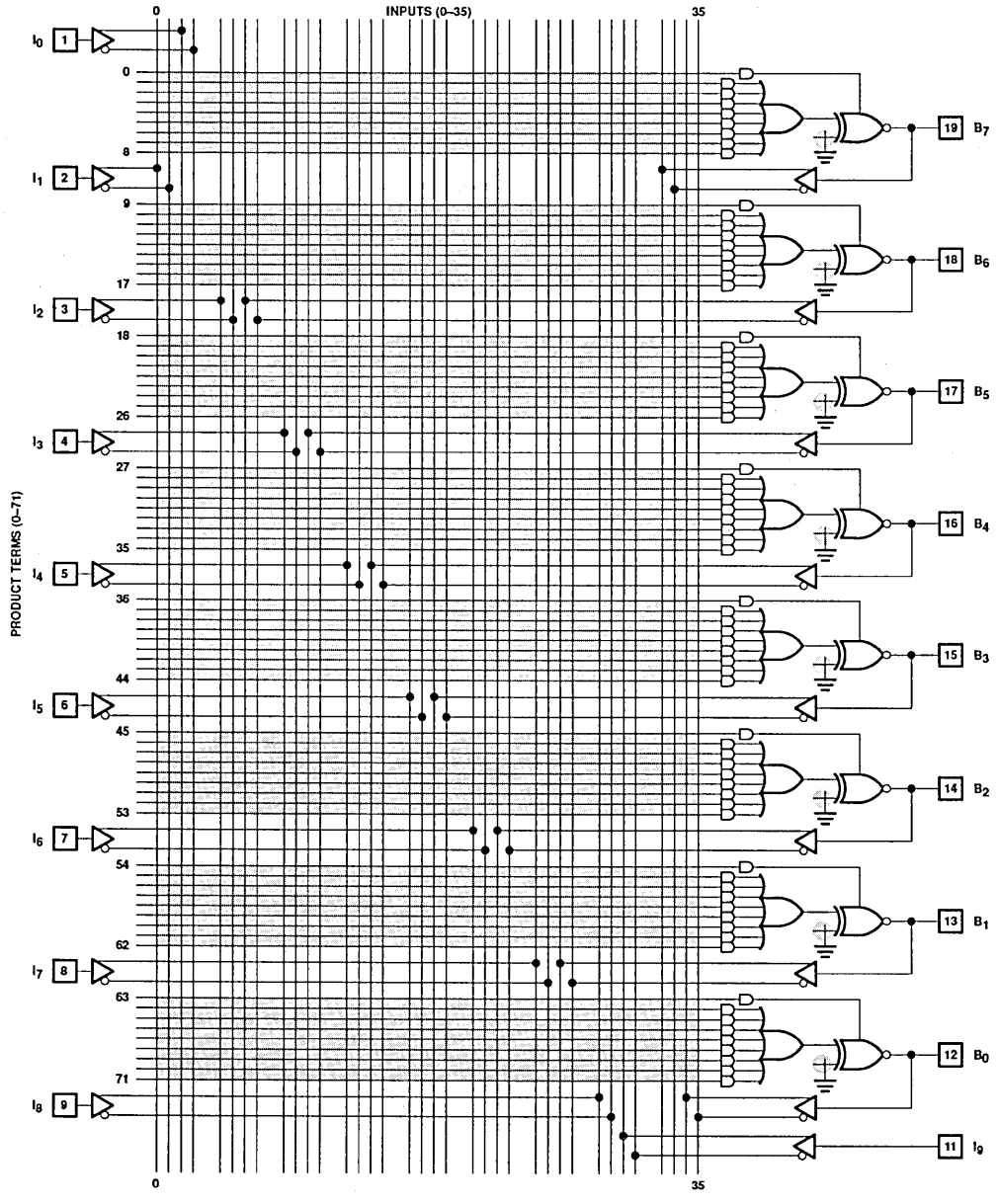


Figure 2–1. Signetics PLHS18P8B

Programmable logic design and application notes

The PLUS16L8D and –7

The PLUS16L8D and –7 PAL-type devices are functionally identical to other commercially available 16L8 PAL ICs. Figure 2–2 shows an extremely simplified version. Less flexible than a PLA, the PLUS16L8D/–7 provides raw speed and current drive so important for driving SRAM arrays on RISC processors or the control/data lines on rapid bus structures. The PLUS16L8D has a worst-case propagation delay of 10ns. The worst-case t_{PD} of the –7 is 7.5ns. 24mA output drive is guaranteed.

The PLUS16L8D/–7 have seven product terms per OR function and one per 3-State control. Six of the eight outputs can be configured as inputs or outputs. The PLUS16L8D/–7 are available in 20-pin plastic DIL or 20-pin PLCC packages.

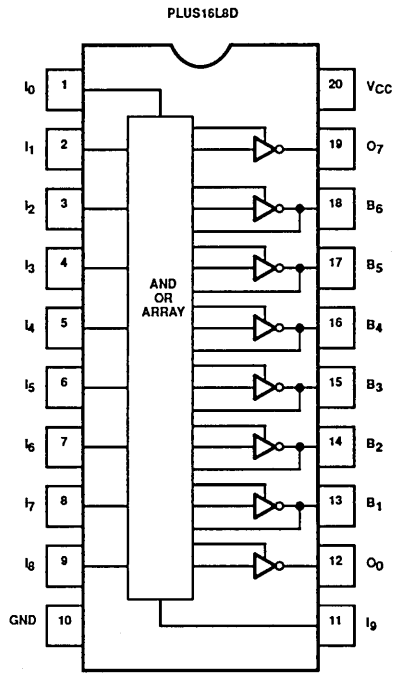


Figure 2–2. PLUS16L8 (D and –7)

Programmable logic design and application notes

The PLUS16R8D and –7

The PLUS16R8D and –7, like the PLUS16L8D and –7 is identical to other manufacturers' registered PAL devices. The parts have eight inputs, eight outputs, and eight D-flip-flops. Each flip-flop feeds an output pin through a 3-State buffer. The output of each D-flip-flop, $\overline{Q}$, is also fed back to the AND array. Each output is capable of driving 24mA I_{OL} max, with all ouputs simultaneously asserted.

The PLUS16R8D has a worst-case propagation delay of 10ns. The worst-case t_{PD} of the –7 is 7.5ns. The PLUS16R8D and –7 are available in 20-pin plastic DIP and 20-pin PLCC.

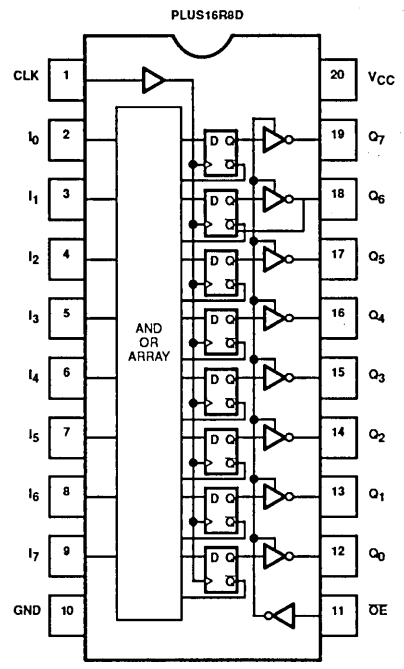


Figure 2–3. PLUS16R8 (D and –7)

Programmable logic design and application notes

The PLUS20L8D and –7

The PLUS20L8D and –7 devices have 14 inputs, two dedicated outputs and six bidirectionals. The t_{PD} are 10ns max and 7.5ns max, respectively. The 24mA of output low current of these devices can drive capacitive address line inputs and pc-board traces through long layouts. This makes the particularly suitable for driving SRAM, video DRAM, and FAST dynamic RAM arrays in 32-bit microprocessor environments.

Identical to other commercially available 20L8 PAL devices, the PLUS20L8D and –7 have 56 functional product terms which are hard-wired to eight OR gates. Each OR gate drives an Active-Low output. The 3-State control of each output is from a dedicated AND product term.

The worst-case propagation delays for the PLUS20L8D and 20L8–7 are 10ns and 7.5ns, respectively.

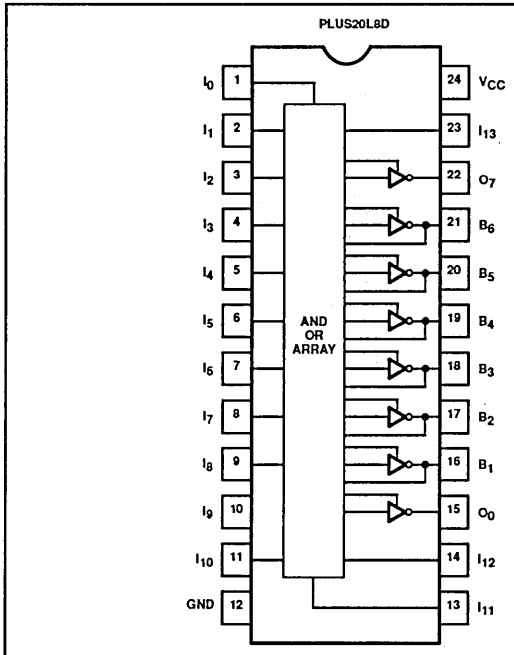


Figure 2–4. PLUS20L8 (D and –7)

Programmable logic design and application notes

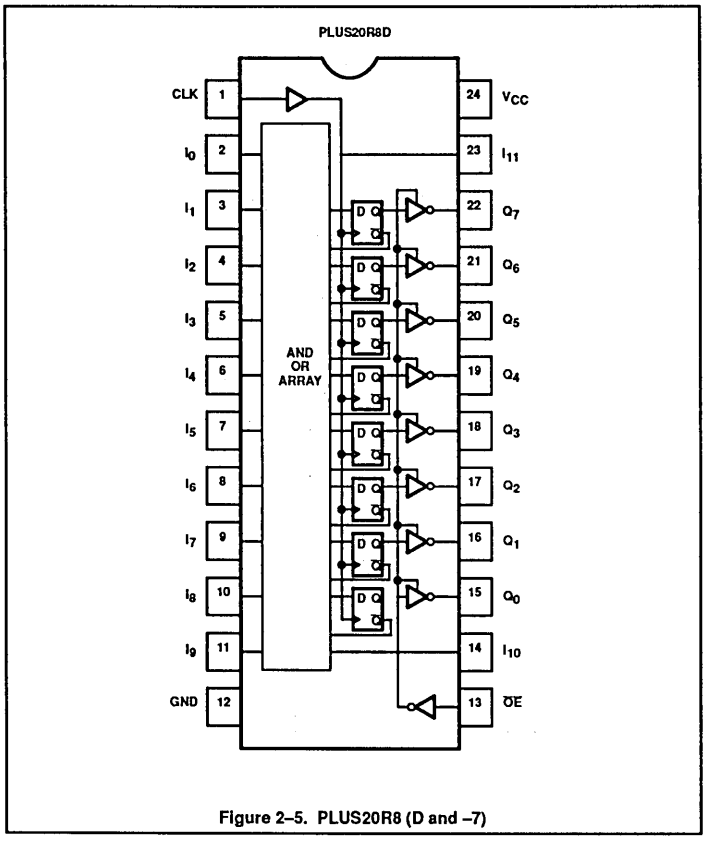


Figure 2–5. PLUS20R8 (D and –7)

The PLUS20R8D and –7

The PLUS20R8D and –7 are 24-pin versions of the 16R8 PAL device. With propagation delays of 10ns and 7.5ns max, the parts deliver 24mA of output low current drive. Eight D-flip-flops share a common clock and output enable line. The output of each flip-flop is dedicated to a separate output pin and is also fed back to the AND array.

The PLUS20R8D and –7 are available in 24-pin plastic DIL and 28-pin PLCC.

The PLC18V8Z

The PLC18V8Z is a multi-function, generic PAL-type device. It is pin-compatible with, and can replace 22 different 20-pin registered and combinatorial PAL devices. To accomplish this, the conventional 'single function' output pin has been replaced by a configurable Output Macro Cell. Each Macro Cell contains a D-flip-flop or a combinatorial I/O path. Output polarity and 3-State control functions are also individually configurable.

Each OMC is fed by nine AND product terms, which are hard-wired in the classic PAL fashion.

One of the key features of the part is its ability to sink 24 milliamps I_{OL}, compatible with other bipolar PAL devices—yet still comply with internal CMOS circuitry. The UV erasable version is available in 20-pin ceramic DIL with a quartz window.

Programmable logic design and application notes

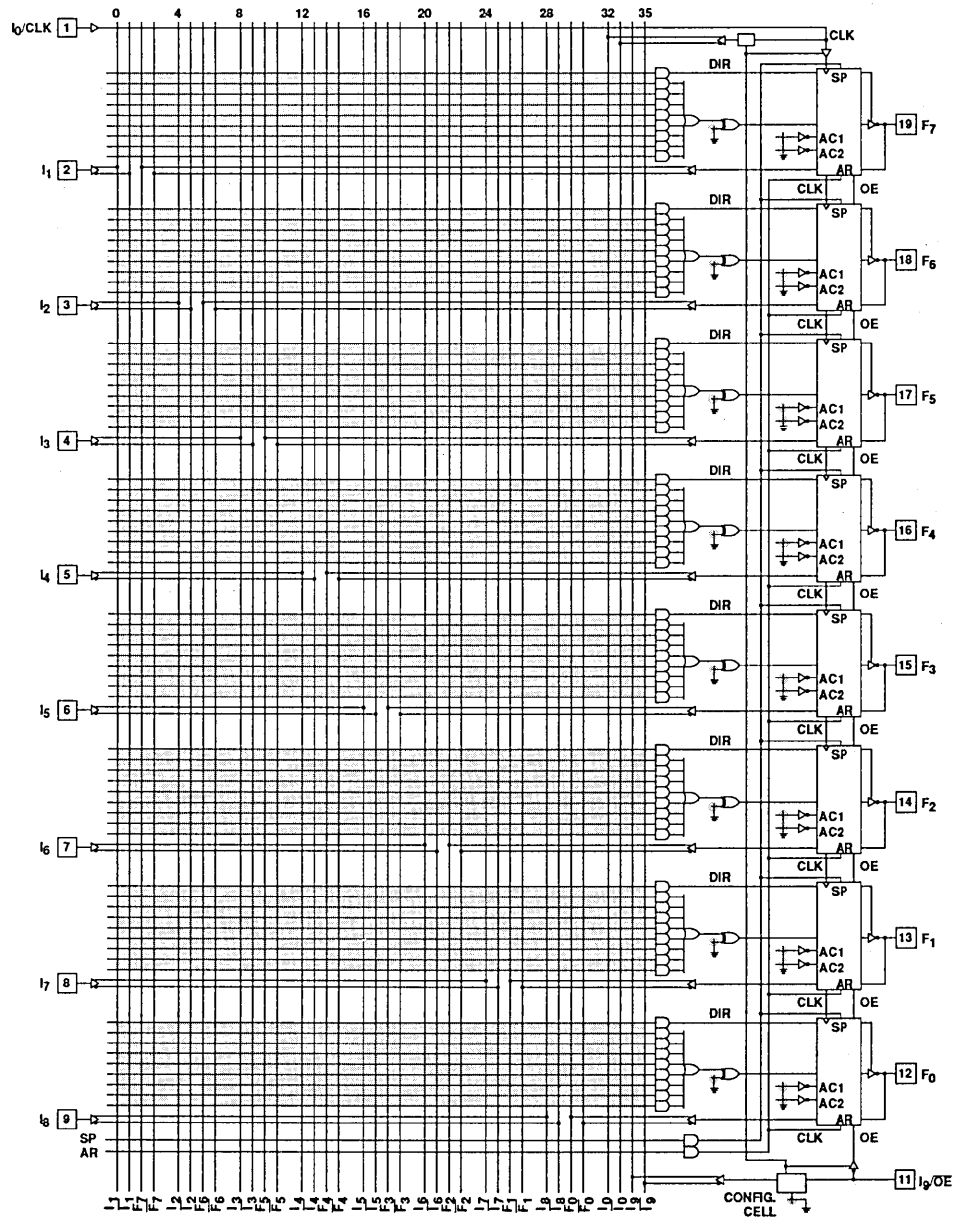


Figure 2–6. PLC18V8Z Architecture

Programmable logic design and application notes

Programmable Logic Devices

SEQUENCER DEVICES

Introduction

Ten years ago, in their search for a straightforward solution to complex sequential problems, Signetics originated Programmable Logic Sequencers. Signetics Programmable Sequencers represent a product line which combines the versatility of two programmable arrays (PLA concept) with flip-flops, to achieve powerful state machine architectures.

Each arrangement or "architecture" offers a variation of the basic concept which combines two programmable logic arrays with some flip-flops, in an undedicated fashion. The PLA product terms are not specifically dedicated to any particular flip-flop. All, none, or any mix in between may be connected to any flip-flop the designer chooses. The PLA structure therefore supports 100% product term-sharing as well as very wide OR functions preceding the flip-flops.

Signetics line of Programmable Logic Sequencers has been further customized to accommodate specific types of state machine designs. Some have both registered and combinatorial outputs, specifically for synchronous and asynchronous Moore-type state machines. Others have state or buried registers, as well as output registers. These devices (PLUS105, PLC42VA12 and PLUS405) are ideal for synchronous Mealy-type applications.

J-K and S-R register functions are another benefit. The logic functions provided by these types of registers far exceed the capability of a D-type register. The functionality of the J-K allows the designer to optimize the logic used in generating state transitions. Ninety percent of PAL devices have D-type registers. All the sequencers are equipped with three state options for bussing operations, JK or SR flip-flops and some form of register Preset/Reset functions.

Finally, all PLS devices have a Transition Complement Array. This asynchronous feedback path, from the OR array to the AND array, generates "complement" transition functions using a single term. Virtually hidden in between the AND array and the OR array is the Complement Array. This single NOR gate is not necessarily "an array," however the inputs and outputs of this complement gate span the entire AND array. The input(s) to the Complement Array can be any of the product terms from the AND array. The output of the Complement Array will be the 'complement' of the product term input. If several product terms are connected to the Complement Array, their respective complements can also be generated. The output of the Complement Array is fed back to the AND array, whereby it can be logically gated through another AND gate and finally propagated to the OR array. The significance being that the complement state of several product terms can be generated using one additional AND product term. For example, if an efficient method of sensing that no inputs were asserted was needed, the designer could connect the output of appropriate AND gates to the complement NOR gate. The output of the NOR gate could then be used to condition and then set or reset a flip-flop accordingly. As well, he could detect a particular state variable combination and force a transition to a new state, independent of the inputs. Or he could combine input signals and state (AND) terms to generate a new composite term. In any of these applications, the Complement Array greatly reduces the number of state transition terms required.

In order to present the material in the most concise fashion, a brief state equation tutorial is presented first. The PLUS105 description immediately follows. In this capsule description, the level of detail is expanded, so read it first for basic understanding. Each additional presentation will be done with regard to the fundamentals described for the PLUS105. Figure 3-3 shows the detailed drawing of the PLUS105 in full detail. Figure 3-4 shows a compressed rendition of the same diagram so that the reader can understand the diagram notation. The compressed shorthand version will be used for the rest of the sequencers.

Programmable logic design and application notes

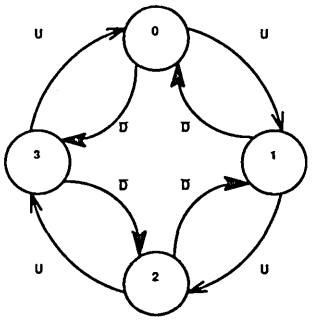


Figure 3-1. Up-Down Counter State Diagram

```
While     [STATE 0]
          IF     [U]     THEN          [STATE 1]
          IF     [/D]    THEN          [STATE 3]

While     [STATE 1]
          IF     [U]     THEN          [STATE 2]
          IF     [/D]    THEN          [STATE 0]

While     [STATE 2]
          IF     [U]     THEN          [STATE 3]
          IF     [/D]    THEN          [STATE 1]

When      [STATE 3]
          IF     [U]     THEN          [STATE 0]
          IF     [/D]    THEN          [STATE 2]
```

Figure 3–2. STATE EQUATIONS to Implement Up-Down Counter

State Equation Tutorial

STATE equation entry is a convenient way to describe elementary sequential machines in a manner which is directly related to a state diagram of the machine. The basic commands are few, but can be combined in a powerful fashion. Figure 3-1 shows a 4 state up-down counter for a machine with an U(up)/D(down) input line. Figure 3-2 shows the state equation syntax to implement Figure 3-1.

The basic meaning can be summarized in the following way. Simply, "while in state X" if input "Y" occurs, "transverse to state Z". This is a Moore machine model. Mealy may be accommodated by addition of the "with" operation which designates an output variable being associated as shown below:

A.) While [CURRENT STATE]
with [OUTPUT VARIABLE]
IF [INPUT VARIABLE]
then [NEXT STATE]

or

B.) While [CURRENT STATE]
IF [INPUT VARIABLE]
then [NEXT STATE]
with [OUTPUT VARIABLE]

If a latched output variable is desired, the addition of a prime notion (/) to the right of the output variable is required.

The designer must assign the binary values of choice to specific states for a state equation function to be implemented. The Signetics AMAZE manual details state equation solutions with more examples, but the advantage of state equations is that the designer can be less involved with the internal structure of the sequencer than required by other methods.

The PLUS105

This part (Figure 3-3) has sixteen logic inputs and eight outputs. It also has eight S-R flip-flops tied directly to those output pins through 3-State buffers (common control from pin 19). The user may select pin 19 to be an Output Enable signal or an asynchronous preset (PR) signal which is common to all flip-flops. Embedded into the device are 48 AND gates. All flip-flops are S-R type with an OR gate on both S and R. The designer may choose any number of product terms and connect them with any OR gate. The product terms can also be shared across any OR gate, as needed. Six of the 14 flip-flops are termed "buried registers" as their outputs are fed back to the AND array, regenerating both the Q and /Q state variables. There is no direct connection to an output. Both the input signals and the state variables Q and /Q are fed to the AND array through buffers which provided the TRUE (or noninverted) and Complement (inverted) renditions of the variable. This is critical for the efficient use of the AND array. The designer has all state and input variables necessary to generate any state transition signal to set and/or reset commands to the flip-flops. Because of this AND/OR arrangement, combined with complete freedom of configuration, all sequential design optimization methods are applicable.

There are many other feature capabilities suitable for creative usage. For example, it is common practice to use the 48 product terms with the 6-bit buried register, treating the output 8-bit register as an intermediate, loadable data register only. This provides a very good bus "pipeline" for the internal 6-bit machine. However, other logic options can be accomplished by combining internal state information (present state) with current input information, generating a next state which is different from the current internal state.

Programmable logic design and application notes

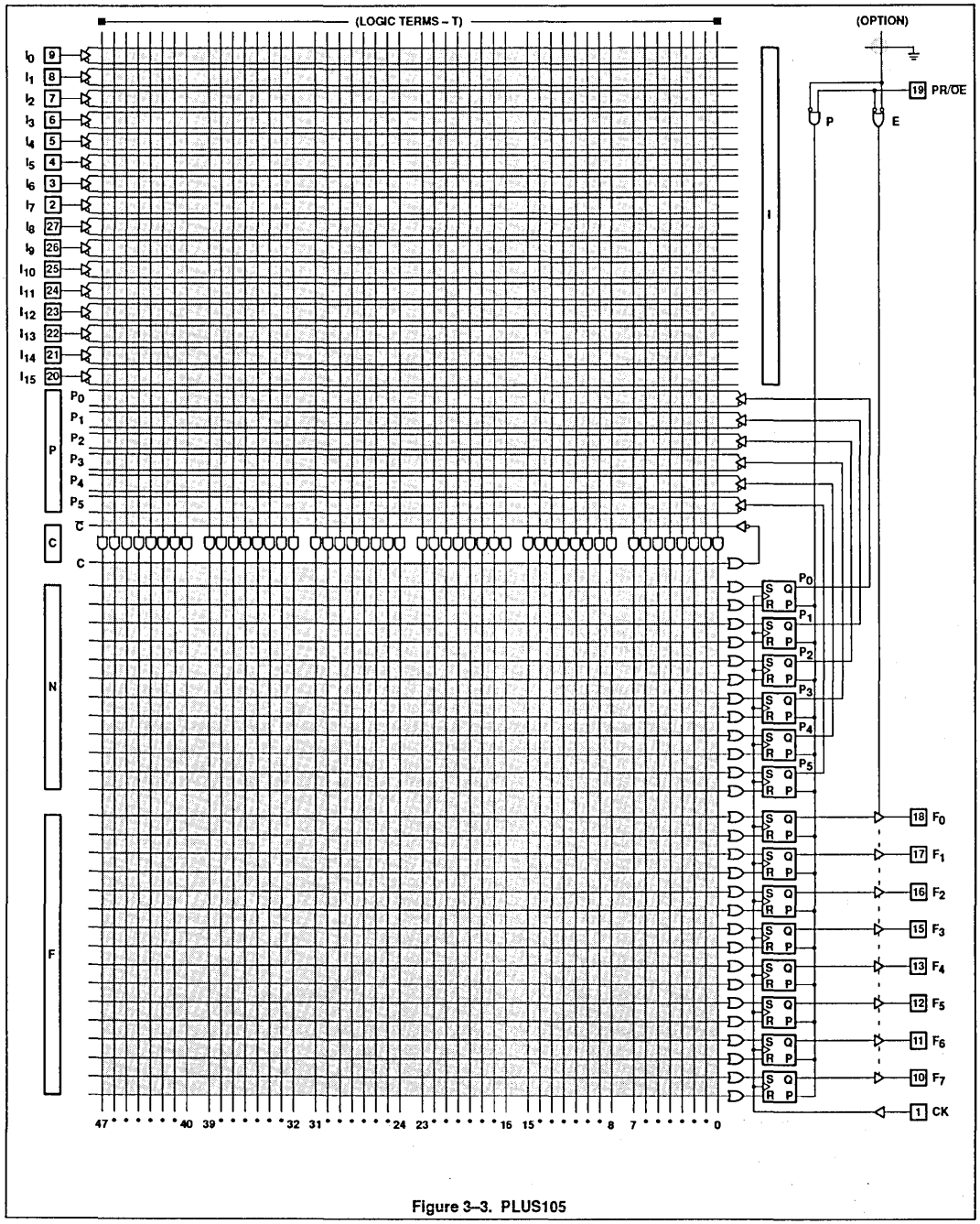


Figure 3–3. PLUS105

Programmable logic design and application notes

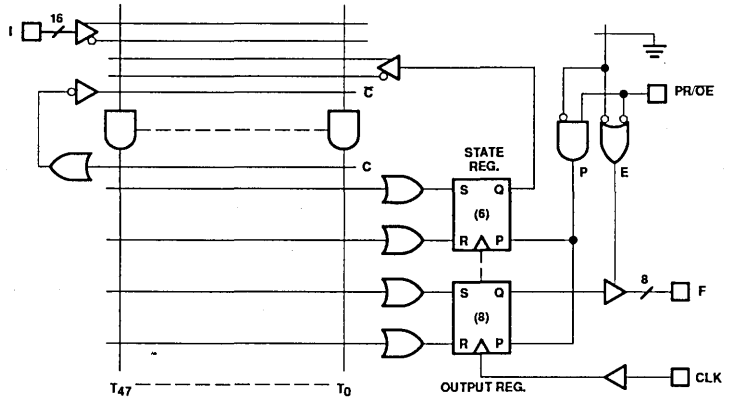


Figure 3-4. Compressed Drawing of PLUS105

Programmable logic design and application notes

The PLS155, 157, and 159A constitute a three part family of 20-pin sequencers that are well suited for high speed handshakers, counters, shift registers, pattern detectors and sequence generators. Additional applications include testability enhancement, demonstrated in the application examples of signature analysis and pseudo random number generation. The three devices are very similar in architecture. All have a total of 12 possible outputs. The difference is the ratio of combinational I/O to registered outputs available.

The PLS155

The PLS155 is a sequencer providing four J-K flip-flops with a PLA having 32 logic product terms and 13 control product terms. Eight combinational I/O are available in addition to the four registered outputs. All of the state variables and combinational variables are presented to the output pins by way of 3-State inverting buffers. The combinational and state variable outputs are fully connected (fed back) back to the AND array in both the True and Complemented form of the variable. The product includes a special feature that allows the user to configure the flip-flops as either J-K or D flip-flops on an individual basis. A Register Preload feature is supported via two product terms (La, Lb) which permit "back loading" of data into the flip-flops, directly from the output pins. The part can now be easily forced into any known state by enabling La, Lb, applying data at the outputs (previously "3-Stated"), and applying a clock pulse. Register Preset and Reset functions are controlled in 2 banks of 2 registers each. Note that control product terms are from the OR array.

The outputs of all variables are 3-State controlled by a unique partition. Pin 11 provides an Output Enable input (OE) which can be asserted with the EA and EB control product terms. EA controls the flip-flops F0 and F1, and EB controls F2 and F3. Each combinational output term has a distinct 3-State control term (D0 - D7) originating from the AND array of the PLA. Each combinational output variable can be programmed as inverting (active LOW) or non-inverting (active HIGH) by way of the output polarity EX-OR gate associated with each I/O pin.

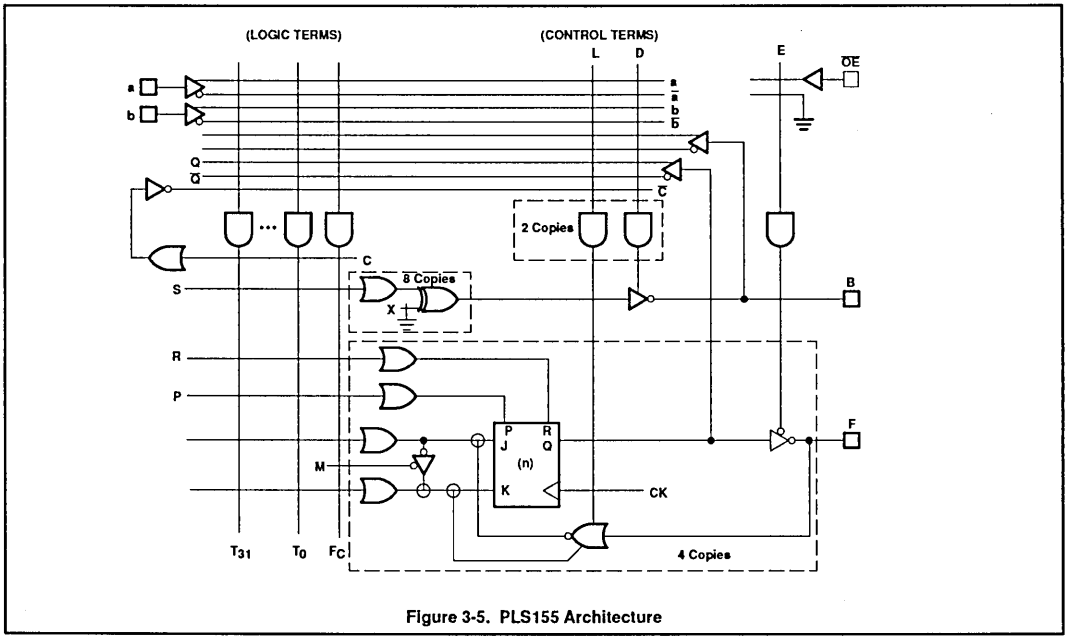


Figure 3-5. PLS155 Architecture

Programmable logic design and application notes

The PLS157

This sequencer features all the attributes of the aforementioned PLS155, however, two flip-flops have been added, at the expense of two of the combinational outputs. Pins 13 and 18 on the PLS157 are flip-flop driven, where the same pins on the PLS155 are combinatorial, driven from the PLA. Again, all variables (input, output, or state variables) fully connect over the PLA portion with both True and complemented versions supplied.

The number of product terms, the Complement array, Output Enable, 3-State configurations, Register Preload, etc., track the PLS155 part. As with the PLS155, distinct clock input on pin 1 is provided for synchronous operation. Register Preset and Reset are available in 2 banks. Pin F_4 and F_5 are controlled from the AND array (Product Terms P_B and R_B). The remaining 4 registers, F_0 - F_3, are controlled by the sum terms (from the OR array) P_A and R_A.

Designs requiring more than 16 states but less than or equal to 64 states are solid candidates for realization with the PLS157. It can be configured as a Moore machine for counter and shifter designs from the flip-flop outputs, or as high speed pulse generators or sequence detectors with the combinational outputs. Mixed solutions are also possible.

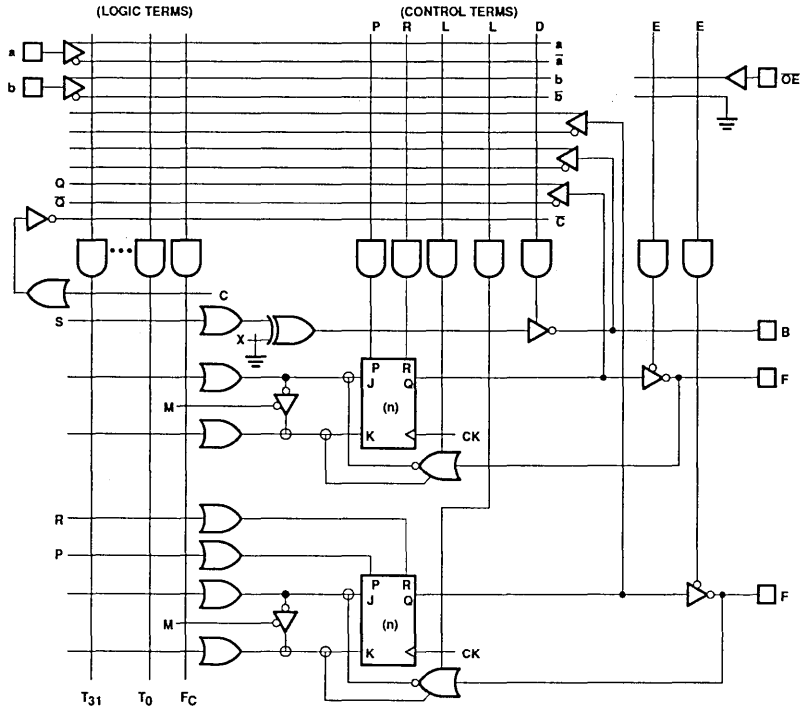


Figure 3-6. PLS157 Architecture

Programmable logic design and application notes

The PLS159A

By extending the PLS157 arrangement even further, the PLS159A can be derived. Again, maintaining identical input, product terms, Complement array and similar 3-State partitioning, the PLS159A also resides in a 20-pin package. The expansion to dual 4-bit banks of flip-flops, at the expense of 2 combinational outputs, enhances the number of available internal states while maintaining product term and pin compatibility. Note that all registers are controlled from the AND array in 2 groups of four.

The PLS159A is an octal part. It readily enters the environment of the 8-bit data operand as well as the bus oriented system.

For enhanced performance, the flip-flop outputs are inverted. To provide positive outputs for shifters and counters, the input variables and state feedback variables can be selectively inverted through an input receiver or the feedback path through the AND gate array.

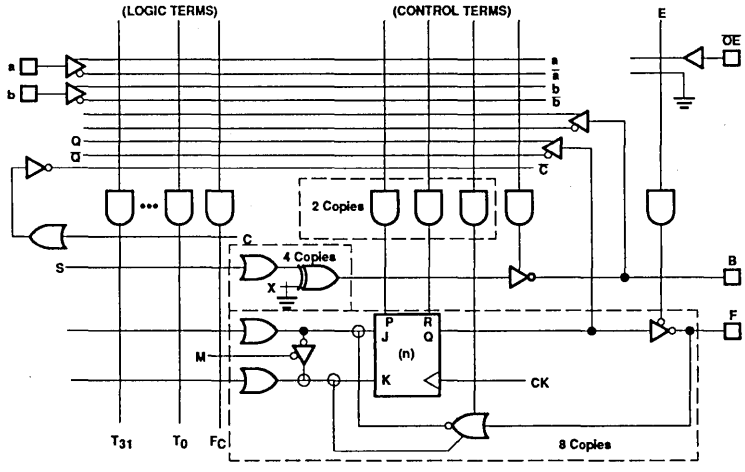


Figure 3-7. PLS159A Architecture

Programmable logic design and application notes

There are three basic members in the 24-pin package family: The PLS167A, the PLS168A, and the PLS179. The PLC42VA12 is discussed elsewhere.

The PLS167A

The PLS167A has 14 logic inputs and six registered outputs (S-R flip-flops). Six additional buried flip-flops reside beside the 48 product term AND array. This device can support state machine designs of up to 256 states—as two outputs feed back into the AND array, making a total of eight buried registers. There is complete feedback connectivity of the inputs and the state flip-flop outputs to the PLA AND gates. Organizationally it has much more in common with the PLS105A than the aforementioned 20-pin parts. The asynchronous Preset and the Output Enable are identical to the PLS105A.

By having the output latched state variable capability, it provides an automatic buffer for bus based systems. The current state may be presented, fully stable and synchronized to a bus—while the internal buried machine is transitioning to the next state based on current input conditions.

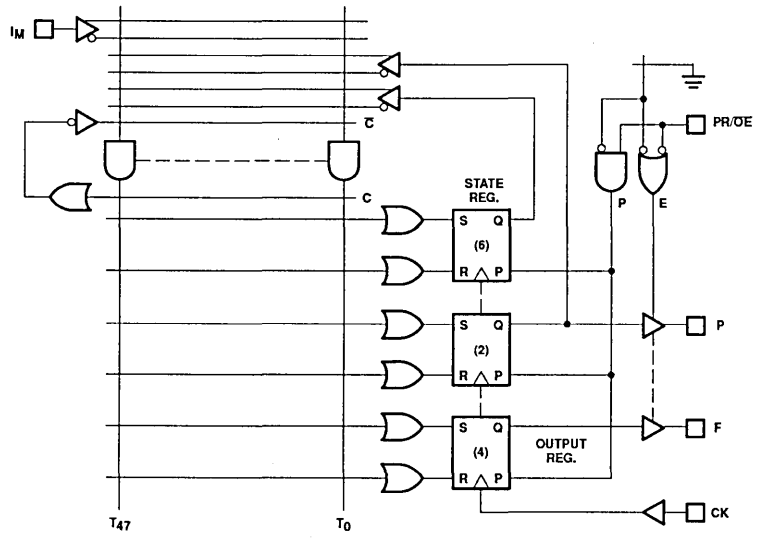


Figure 3-8. PLS167A

Programmable logic design and application notes

The PLS168A

This sequencer is a down-scaled version of the PLS105A. Having identical product terms, Complement array, asynchronous PRESET/Output Enable options, and 3-State controls, its primary difference is having 12 inputs compared to the PLS105A's 16 inputs. However, the PLS168A can become a state machine of up to 1024 states due to internal feed back of its six state registers, plus the feedback of four of the eight output registers. The PLS168A is packaged in a 300mil-wide 24-pin DIP or 28-pin PLCC.

This is also an octal part, providing an 8-bit register to a bus based system. State registers, interrupt vector synchronizers, counters, shifters, or just about any basic state machine can be generated and 3-State interfaced to a computer bus with a PLS168A. Outputs provided by the positive asserted sense make state transitioning and loading of state variables straightforward.

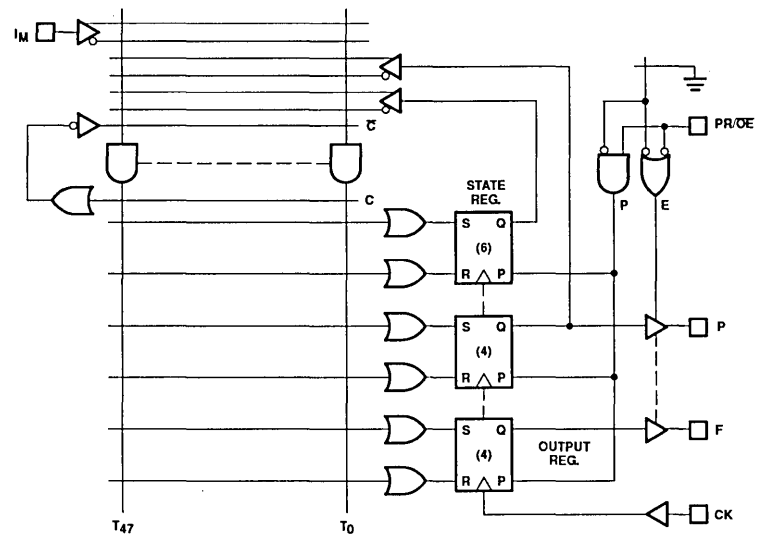


Figure 3-9. PLS168A

Programmable logic design and application notes

The PLS179

The PLS179 is architecturally similar to the PLS159A. The 3-State enable, number of product terms, flip-flop mode controls, register preload, etc., are all identical to the PLS159A. The four additional inputs are the dominant differentiating feature for this part as compared to the PLS159A. As with the PLS159A, the PLS179 Preset and Reset functions are controlled from the AND array in 2 groups of 4 registers each.

The PLS179 is also an octal part. Providing the state contents directly to the pin through 3-State buffers allows counters and other sequence generators direct access to an asserted low octal bus. Some design creativity will generate positive assertion through the pin inverters, for positive driven busses. Additional input pins expand the capability of the part beyond the PLS159A. Input combinations may be presented in a wider format, more fully decoded to the sequencer for faster reaction and less external circuitry than the PLS159A requires.

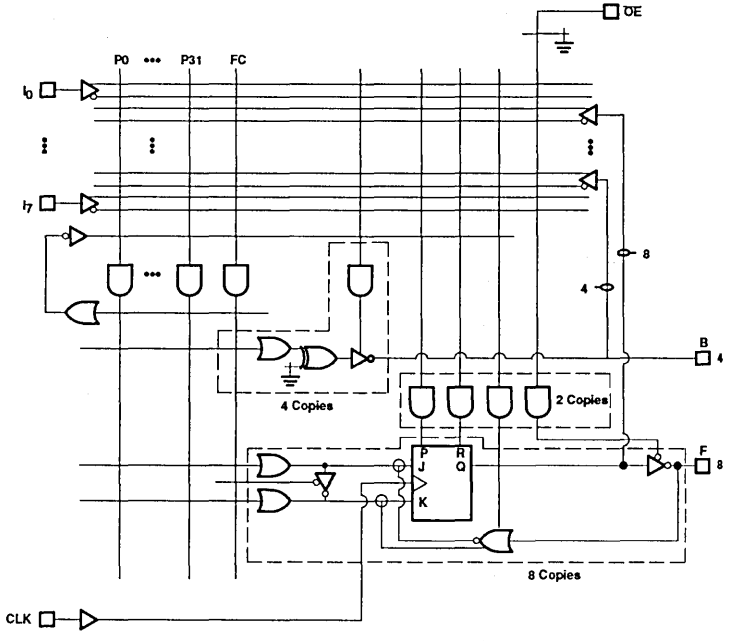


Figure 3-10. PLS179 Architecture

Programmable logic design and application notes

The PLUS405

The PLUS405 is a functional superset of the PLUS105. It is also much faster. The performance of the PLUS405 has been dramatically improved relative to the PLS105A. Available in two speed versions, the operating frequencies ($^1/_{t_{IS}} + t_{CKO}$) range from 37 to 45MHz (minimum guaranteed frequency). The clock frequencies, or toggle rate of the flip-flops, are 50MHz and 58.8MHz, respectively. The PLUS405 has 16 more product terms and two more buried state registers than the PLUS105. Equipped with two independent clocks, it is partitionable into two distinct state machines with independent clocks. And, it contains two independent Complement arrays, allowing full benefits over both machines.

The PLUS405 can be partitioned as one large state machine (16FFs) with 64 available p-terms using one clock and 16 inputs or alternately two state machines (8FFs each) with independent clocks, sharing 64 p-terms with 15 inputs in any combination the user desires. The Complement arrays can be used to generate the "else" transition over each state machine or alternately used as NOR gates. They can be coupled into a latch if needed.

The Asynchronous Preset option of the PLS105/167/168 architectures has been replaced with a Programmable Initialization feature. Instead of a Preset to all logic "1"s, the user can customize the Preset/Reset pattern of each individual register. When the INIT pin (Pin 19) is raised to a logic "1", all registers are preset/or reset. The clocks are inhibited (locked out) until the INIT signal is taken Low. Note that Pin 19 also controls the OE function. Either Initialization or OE is available, but not both.

A CMOS extension to the PLUS405 is Signetics PLC415, which is pin compatible and a functional superset of the PLUS405 architecture.

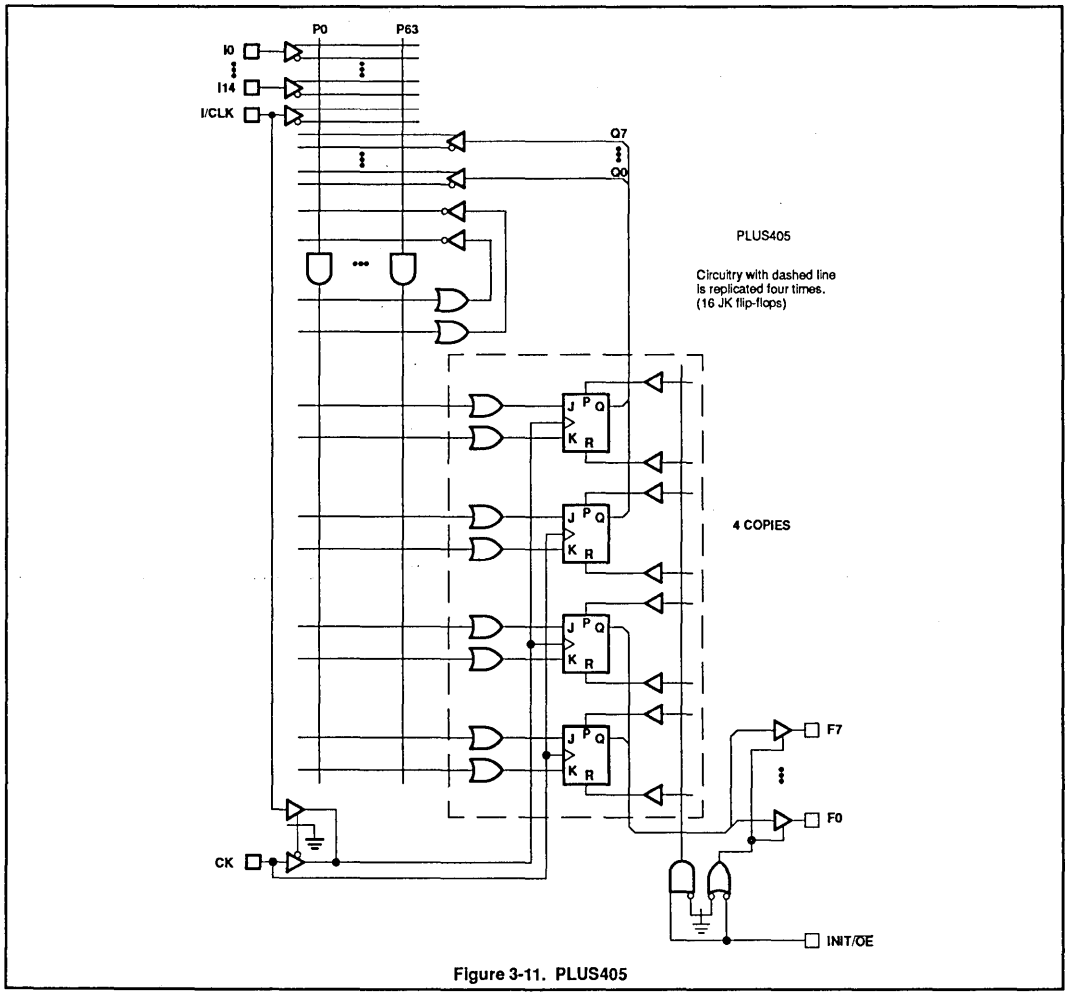


Figure 3-11. PLUS405

Programmable logic design and application notes

The Future Is Here Now.
Recent architectural extensions are currently available from Philips Components-Signetics. These include the PLC415 and PLC42VA12. These new "Super Sequencers" are available now for high-end new designs. Please check the data sheet section of this handbook for more information. See also the CMOS Sequencer Solutions section for more design examples using the PLC415 and the PLC42VA12.

Programmable logic design and application notes

Programmable Logic Devices

APPLICATION SECTION INTRODUCTION

This section provides examples of the wide ranging applications for Signetics PLD products. In microprocessors, for example, PLDs can solve complex interfacing problems. Their wide input gates make them ideal for microprocessor decoding, memory and I/O functions.

Communication is another key area where PLDs can solve difficult problems. Here PLDs simplify the process of developing products to an emerging standard. Signetics has provided a series of examples to show PLDs make it possible to change a design the same instant the standard changes. The examples contain a range of applications from whole protocols and simple scramblers to a customizable speech synthesis system.

Other examples in this section illustrate applications in home security and instrumentation.

The applications in this section are designed to show how Signetics PLDs can solve many classic design problems. However, it is important to note that each example exploits only one of the many facets of the product.

MICROPROCESSOR INTERFACING WITH SIGNETICS PLDS

Microprocessor interfacing is the art of connecting the attributes of a microprocessor, very skillfully to its surrounding environment. They must lineup carefully and match the appropriate timing, address and data signals to achieve an effective interaction. This section illustrates multiple interfacing examples.

Introduction

Architectural bottlenecks have migrated from one point to another within a system throughout the history of computer design. Currently, processor speeds and memory cycles have become so tightly designed that little margin exists should any incompatibility arise between them. Driven for both speed and pin compatibility, DRAM manufacturers have added additional modes to their designs such as the nibble and page modes. Processor designers have resurrected the multiple bus Harvard architectures, as evident in some of the commercially available RISC chips. And, by using small block read ahead caches, the processors hide slower DRAM accesses typically by bursting as many as four words in a read cycle. Attempting to match the DRAM to the processor, or perform parity or ECC at full speed (i.e., no wait states) requires a fine balance of time budgeting, cost tradeoff and impedance matching among other issues. Performing all of these functions has resulted in an address decode time between 10 and 20 nanoseconds, depending on the required set of tradeoffs. For example, a tight 80386 memory cycle at full speed may require 10 nanoseconds, which could be accomplished with a D-speed PLD, or with a fast PROM. Less than 10ns is desirable, so 7.5ns PAL-type devices will help. The new PHD16N8 and PHD48N22 are ideal.

System Partitioning

Currently, most 32-bit processors generate an address capable of logically spanning four gigabytes. This is accomplished with 32 bits of distinct address lines. Available memories occupy much less (i.e., 1 megabit or less). A 1 megabit DRAM requires twenty bits of address, so selecting across twelve bit fields may be appropriate. Single module selection (or common address banks) could be accomplished with any logic device which can decode (i.e., generate a select condition) over the 12 high order address bits. Many contenders exist for this reason. The classic solution would be the 74S133 (t_{PD} = 4ns) 13 input NAND gates with an additional 74S04 inverter to decode. Total decode time is at least eight nanoseconds. Depending on the cycle requirements, this may be required, but typically is not. A more efficient method is simply a PLD which combines the wide logic gate with "free" input inverters where required.

Additional select qualifiers may be needed to distinguish the precise assertion time of the select signal. The total number of decoding inputs will exceed the applied address signals.

Given the memory choices selected, the designer must choose a decoding device which meets his criteria. A typical system would have a mix of PROM (system functions), STATIC RAM (no wait memory or cache), DRAM (slower bulk store) or dual port memory (video RAM or shared store). Each will have different timing constraints. Most systems today will have much less than the four gigabytes they can address, but for software expansion reasons (or other system considerations) the memory may not run contiguously and small patches might be spread over the entire range. It will be important to decode precisely to known regions and avoid accidental reference to nonexistent regions.

In selecting a decode device, assuming one is required, several considerations become key. Should the software allow it, or the performance require it, the fastest decode is by distinct selection via direct connection to high order address lines. In today's organizations this will be the fastest, most fragmented memory space. Electrical drive pitfalls can exist here.

Programmable logic design and application notes

Tight layout of the board is also important so that precious, paid for nanoseconds are not given up to long PC connect lines, input capacitance and voltage reflections. Many of the Signetics candidates illustrated in this discussion are limited to speed applications requiring no more than 16mA output drive. They may be inappropriate for extremely dense RAM arrays with long pc-trace interconnects. For simple, fast decode purposes, the D-speed PAL-type devices are good, with a logical choice being the 7.5ns PAL ICs. The new PHD family of parts is even better.

Some straight forward decoding examples follow with criteria for selecting specific Signetics PLD products for decode. These examples exploit only one of many facets offered by these products. Other examples illustrate the use of Signetics PLDs for customized interrupt handling and a most extensive example shows a powerful solution to a SCSI bus interface.

Bus Size Decoding for the 68020 — PLHS18P8B

Address decode for this class of processor is shown in successive sections (i.e., 68030 and 80386). This example depicts a slightly different problem—"data bus sizing" which is accomplished by decoding the address and control signals replicated in the logic diagram in Figure 4-1. Basically the 68020 device will strobe data onto the 32-bit databus in byte oriented subfields of the large word. Sixteen bit ports can receive either the upper or lower 16 data lines. Other ports can respond to LLD (Data 0-7), LMDF (Data 8-15), UMD (Data 16-23), or UUD (Data 24-31). All subfields can be simultaneously asserted as dictated from decoding the size control lines (SIZ0, SIZ1) in conjunction with the low order address lines (A0, A1).

Because the solution requires no product term sharing and is intensive on neither input nor output pins, a simple fast PAL device is the best choice—the PLHS18P8B is designated. The basic operation is to decode the input lines to indicate whether the bus should have 8, 16, or 32 bits driven onto it. These signals are supplied to a device external to the microprocessor which then asserts the corresponding data. Figure 4-2 shows an appropriate pinlist under AMAZE with Figure 4-2 showing the logic equation file.

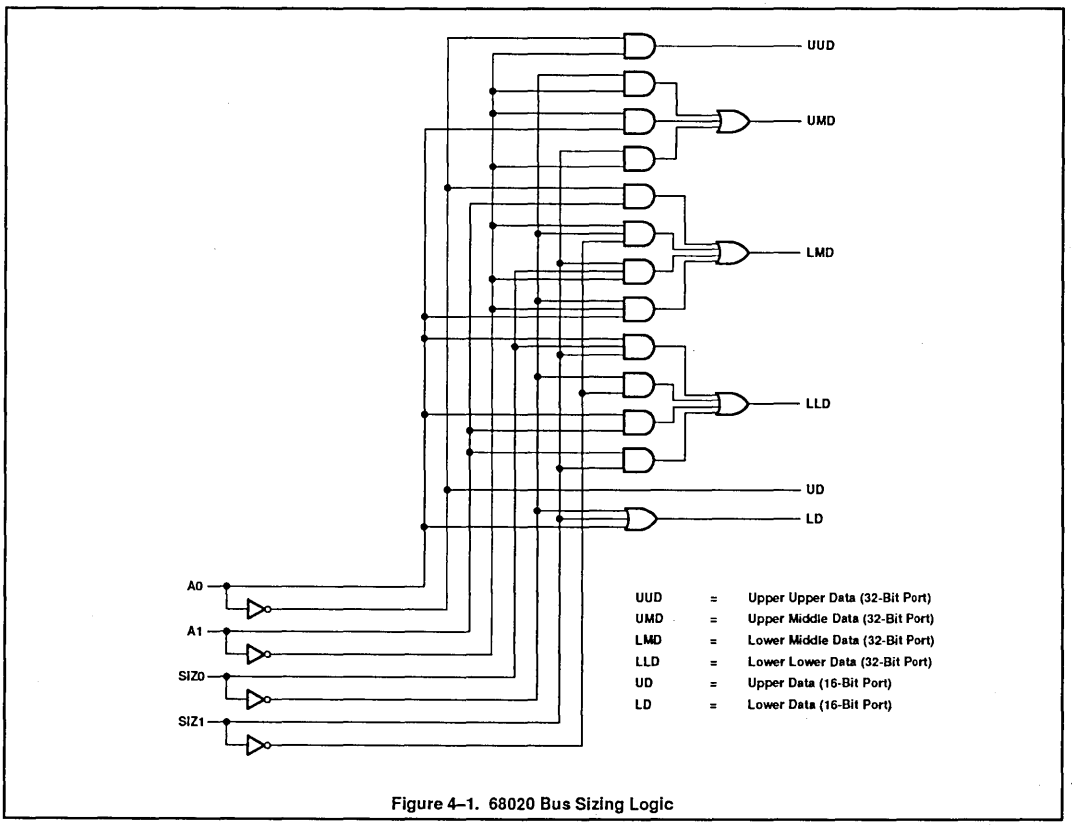


Figure 4–1. 68020 Bus Sizing Logic

Programmable logic design and application notes

```
        File Name    :  BYTESEL
        Date         :  4/31/1988
        @DEVICE
        PLHS18P8
        @DRAWING
        @REVISION
        @DATE
        @SYMBOL
        @COMPANY
        @NAME
        @DESCRIPTION
        @PINLIST
"<--------FUNCTION--------->   <--REFERENCE-->"
"PINLABEL  PIN #  PIN_FCT     PIN_ID  OE_CTRL"
A0           1       I         I0        -   ;
A1           2       I         I1        -   ;
SIZ0         3       I         I2        -   ;
SIZ1         4       I         I3        -   ;
N/C          5       I         I4        -   ;
N/C          6       I         I5        -   ;
N/C          7       I         I6        -   ;
N/C          8       I         I7        -   ;
N/C          9       I         I8        -   ;
GND         10       0V        GND       -   ;
N/C         11       I         I9        -   ;
N/C         12       /B        B0        D0  ;
N/C         13       /B        B1        D1  ;
LD          14       O         B2        D2  ;
UD          15       O         B3        D3  ;
LLD         16       O         B4        D4  ;
LMD         17       O         B5        D5  ;
UMD         18       O         B6        D6  ;
UUD         19       O         B7        D7  ;
VCC         20      +5V        VCC       -   ;
        @COMMON PRODUCT TERM
           "CPT_label = (expression)"
        @I/O DIRECTION
        @LOGIC EQUATION
         UUD= /A0*/A1 ;
         UMD= /SIZ0*A1+A1*A0+SIZ1*/A1 ;
         LMD= /A0*A1+/A1*/SIZ1*/SIZ0+SIZ1*SIZ0*/A1+/SIZ0*/A1*A ;
         LLD= A0*SIZ0*SIZ1+SIZ0*/SIZ1+A0*A1+A1*SIZ1 ;
         UD= /A0 ;
         LD= A0+SIZ1+/SIZ0 ;
```

Figure 4-2. BYTESEL Design File

Interfacing to SPARC — PLUS20L8-7

The SPARC™ processor is a modern RISC device configured from a popular CMOS gate array. Architectural details can be found in data sheets and literature. Supporting a full 32 bit address decode at full speed requires a 60 nanosecond instruction or data cycle. We will consider a 60ns part although a 30ns one is available. Figure 4-3 shows the pin definition and Figure 4-4 the basic timing. The address is driven out in two phases (low and high) and the data must be present 54 nanoseconds after the rising edge of clock 1. To meet the access time will require an SRAM of less than 60ns. In fact, the address generation requires 30ns from when the low address is valid to when the high address is valid. This is almost untenable and most designs will rely on the high order address lines seldom changing with respect to the low order ones. Therefore, assuming the high order lines are static and basing address calculations on the low order transitions seems reasonable. Detecting a change from one "segment" to another in the high lines can key a "wait" condition when addresses make big jumps. By not doing this, will force a very expensive SRAM solution if zero waits are required. Assuming zero wait states are desired, this will require a memory less than 20ns access, if the fastest (7.5ns) PLD is chosen. By virtue of its restricted width and even more restricted speed option for a zero wait state solution, the PLUS20L8–7 is the only contender. This restricts the SPARC address space to 32 independent modules. The low order address lines must be latched within the RAM or externally.

Full performance can be achieved for 32, 64K-bit static RAMs comprising $^1/_2$-mega word store at full speed. By allowing a single wait state, the options open enormously to include a full spectrum of SRAMs, PROMs, even DRAMs with any of the other decode devices. Figure 4-5 shows four such modules selecting off of AL17-AH21 address lines into 16K × 4-bit, 35ns SRAMs. This populates the entire lower two megaword space with high performance static RAM. The high order address lines (AH22-AH31) can select other such modules for expansion purposes.

Programmable logic design and application notes

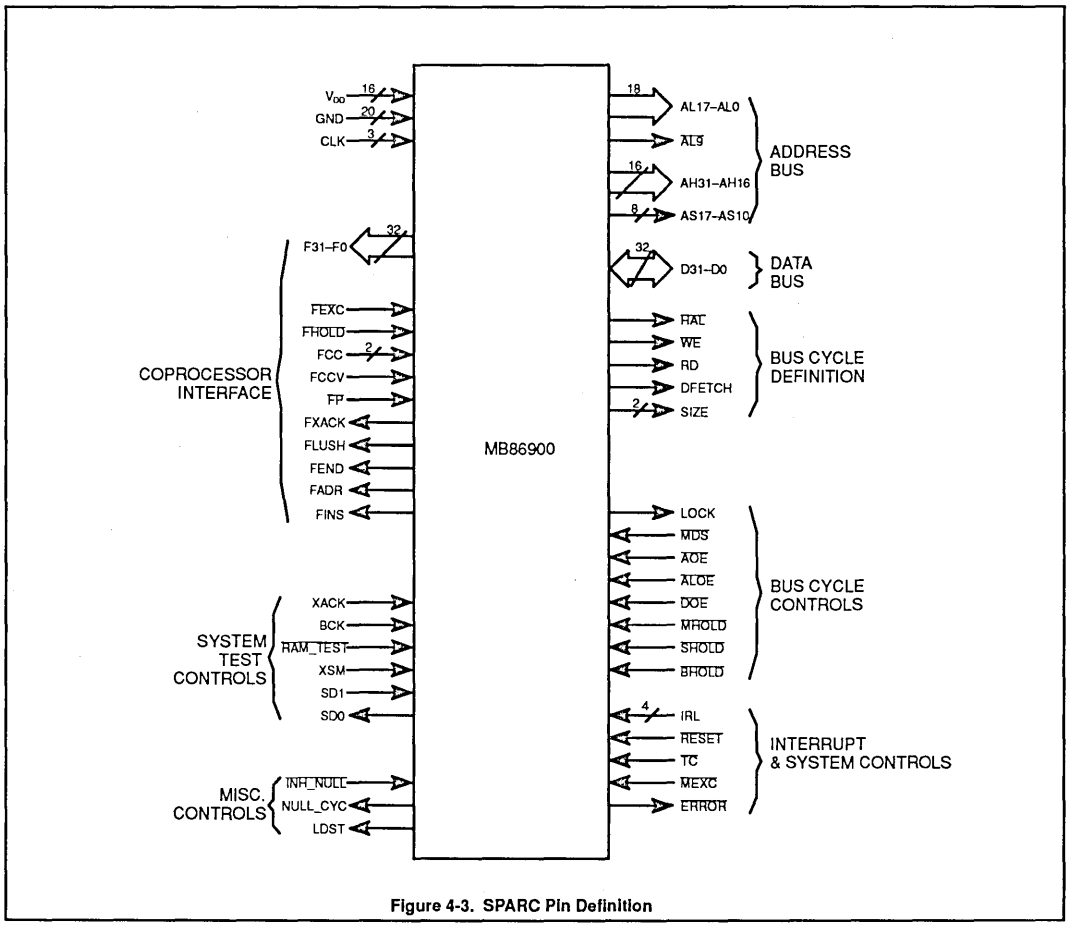


Figure 4-3. SPARC Pin Definition

Programmable logic design and application notes

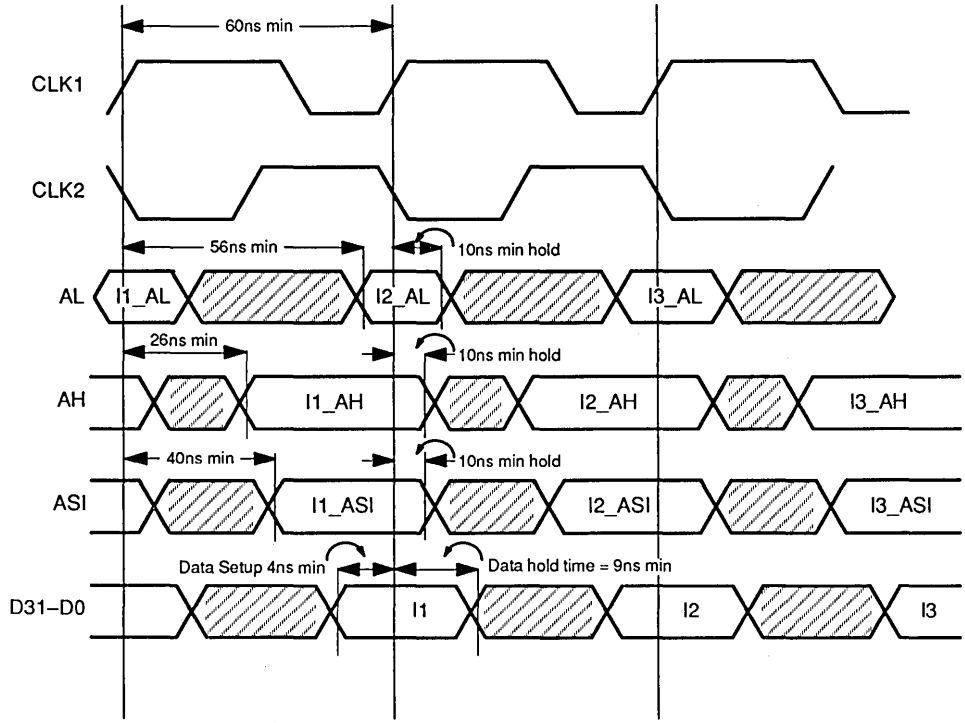


Figure 4-4. SPARC Instruction Fetch Timing

Programmable logic design and application notes

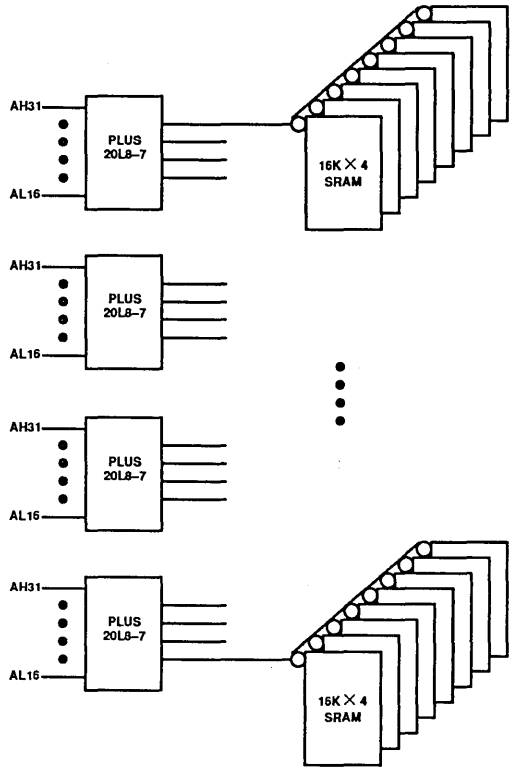


Figure 4–5. 1/2MEG SPARC SPACE

Programmable logic design and application notes

The 80386 Pipeline Decoder — PLUS153D

For example, select a mix of memory that will be located in small addressing chunk segments within the lower 16 megabytes of the 80386 four gigabyte address space. See Figure 4-6.

This decoding method will impact the memory addressing, but in a different way. The 80386 supports a mode whereby a next address can be asserted early (1 clock cycle) if the user asserts the NA# pin. The interleaved slower memory may be sued by getting the nest address earlier than normal to trigger an early memory cycle. Because we will be driving a single signal, NA#, a PLUS153 can be configured with eight inputs and nine of the bidirectionals configured as inputs (i.e., 17 inputs, one output). By decoding addresses 31-17 with the status signal M asserted high and ADS# asserted low, an NA# will assert early to initiate a pipelined early transfer with a slower RAM. This approach allows designers to tune their specific memory speeds to the processor, according to timing needs.

Figure 4-10 shows NA# generated for a pipelined address located in the lowest 128K of the address space. In Figure 4-11, the PLUS153D is shown as a single 17-input NAND function, most of the remaining portion of the part is unused. The address strobe and M signals are included to correctly qualify the address and not generate glitches into the NA# pin. Unless the p.c. board is poorly designed, the output drive of the PLUS153D will be adequate to drive the NA# pin and any additional PC-metal. Figure 4-12 shows the AMAZE equation to decode.

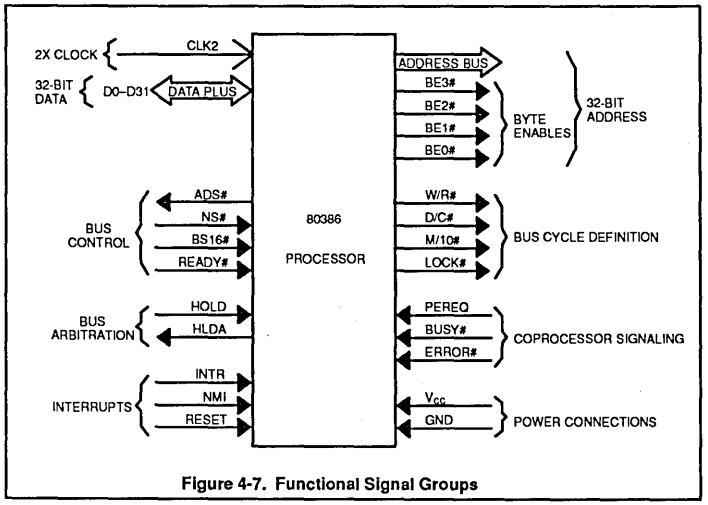


Figure 4-7. Functional Signal Groups

Figure 4-7 illustrates the 80386 signal groups.

Figure 4-8 illustrates various bus cycles with no wait states, and no pipelining.

Figure 4-9 illustrates bus cycles with and without pipelining.

Figure 4-10 illustrates the CLK2 time spans.

Figure 4-11 illustrates the NA# pinlist.

Figure 4-12 illustrates the .BEE file for NA# generation.

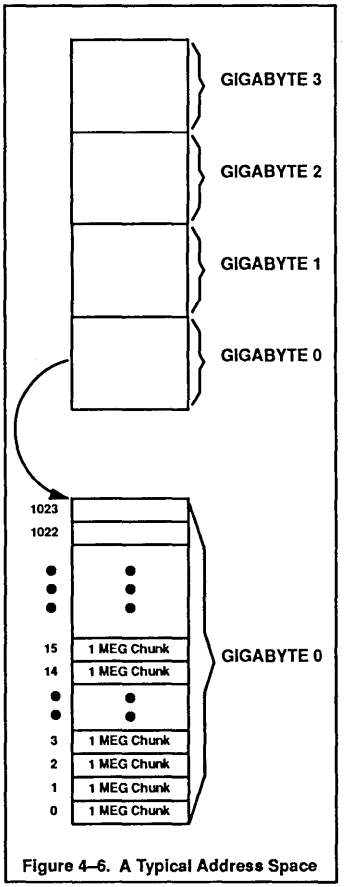
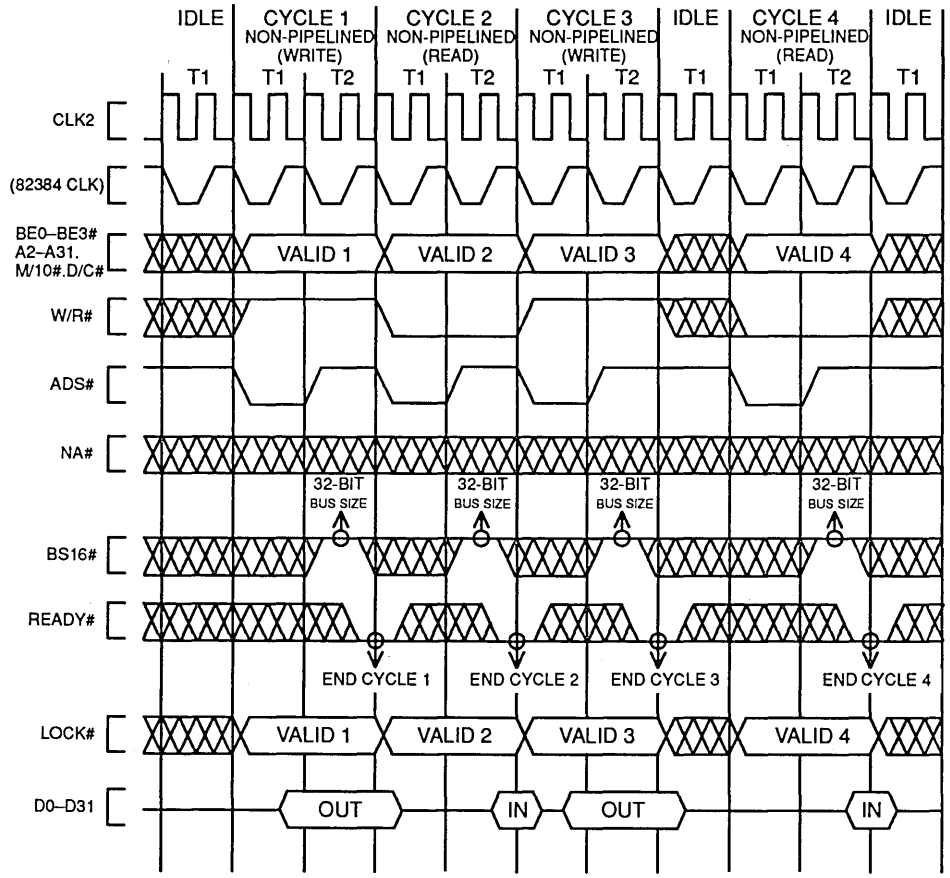


Figure 4–6. A Typical Address Space

Programmable logic design and application notes



IDLE STATES ARE SHOWN HERE FOR DIAGRAM VARIETY ONLY.

WRITE CYCLES ARE NOT ALWAYS FOLLOWED BY AN IDLE STATE.

AN ACTIVE BUS CYCLE CAN IMMEDIATELY FOLLOW THE WRITE CYCLE.

Figure 4-8. Various Bus Cycles and Idle States with Non-Pipeline Address

Programmable logic design and application notes

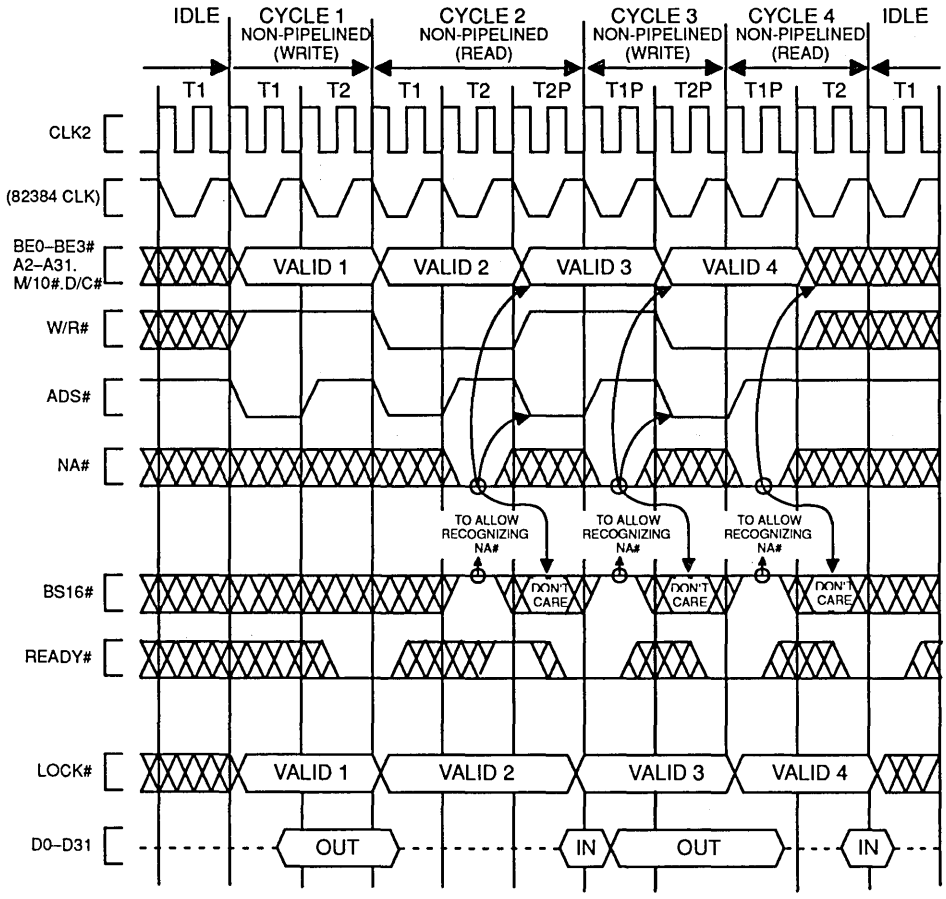


Figure 4-9. Mixed Pipelined/Non-Pipelined Signals

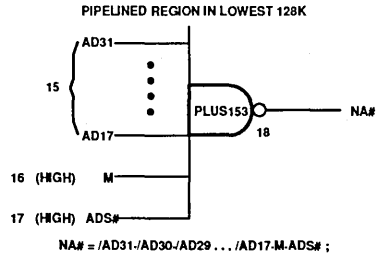


Figure 4-10. Pipelined Region NA# Generator

Programmable logic design and application notes

```
File Name : NA_386
Date : 5/16/1988
Time : 16:37:58

####################   P I N   L I S T   ####################

LABEL   ** FNC **  PIN----------PIN ** FNC   **    LABEL
AD31    ** I   **   1-|          |-20 ** +5V   **    VCC
AD30    ** I   **   2-|    P     |-19 ** I     **    AD22
AD29    ** I   **   3-|    L     |-18 ** I     **    AD21
AD28    ** I   **   4-|    U     |-17 ** I     **    AD20
AD27    ** I   **   5-|    S     |-16 ** I     **    AD19
AD26    ** I   **   6-|    1     |-15 ** I     **    AD18
AD25    ** I   **   7-|    5     |-14 ** I     **    AD17
AD24    ** I   **   8-|    3     |-13 ** I     **    M
AD23    ** I   **   9-|          |-12 ** I     **    /ADS
GND     ** 0V  ** 10-|          |-11 ** /O    **    /NA
                     -----------
```

Figure 4-11. Pinlist NA_386

```
File Name : NA_386
Date : 5/16/88
Time : 16:38:8

@DEVICE TYPE
   PLUS153
@DRAWING

@REVISION

@DATE

@SYMBOL

@COMPANY

@NAME
   /NA386

@DESCRIPTION
THIS DESIGN DRIVES THE NA# SIGNAL LOW WHEN ASSERTED ON AN
80386 PROCESSOR FOR A MEMORY REGION SPANNING THAT DECODED
BY THE EQUATION CONTAINED HEREIN.

@COMMON PRODUCT TERM

@I/O DIRECTION

@LOGIC EQUATION
   /NA=/(/AD31*/AD30*/AD29*/AD28*/AD27*/AD26*/AD25*/AD24*/AD23
      */AD22*/AD21*/AD20*/AD19*/AD18*/AD17*/ADS*M);
```

Figure 4-12. NA_386 .BEE File

Programmable logic design and application notes

68030 Address Decoding — PLUS173D

Although designers generally try to optimize every nanosecond of microprocessor performance, it is typically not necessary for the CPU to always operate at full speed. Operating the CPU, when acceptable, at a slower speed can bring about a more economical and compact system. This is due to higher costs associated with fast memory and greater board area usage of very wide memory configurations.

Some software routines in which slower performance may be acceptable are during power up initialization, diagnostic routines, or possibly some exception processing routines. Where speed is not critical, an 8-bit bus is the most economical and compact because of readily available byte wide memory components and buffers. The 68030 is easily interfaced to 8, 16, or 32-bit ports because it dynamically interprets the port size of the addressed device during each bus cycle. Figure 4-13 shows an example of interfacing both a relatively slow 200ns 8-bit EPROM and fast 35ns 32-bit RAM to a 68030. A PLUS173D was chosen for its high speed and large number of inputs and outputs. Figure 4-14 shows the AMAZE pinlist and Boolean equations for the device. The EPROM occupies memory space 0-32K while the RAM occupies addresses 64K-128K. However, please note that because not all of the upper memory address bits were decoded, the memory arrays will also appear at other addresses.

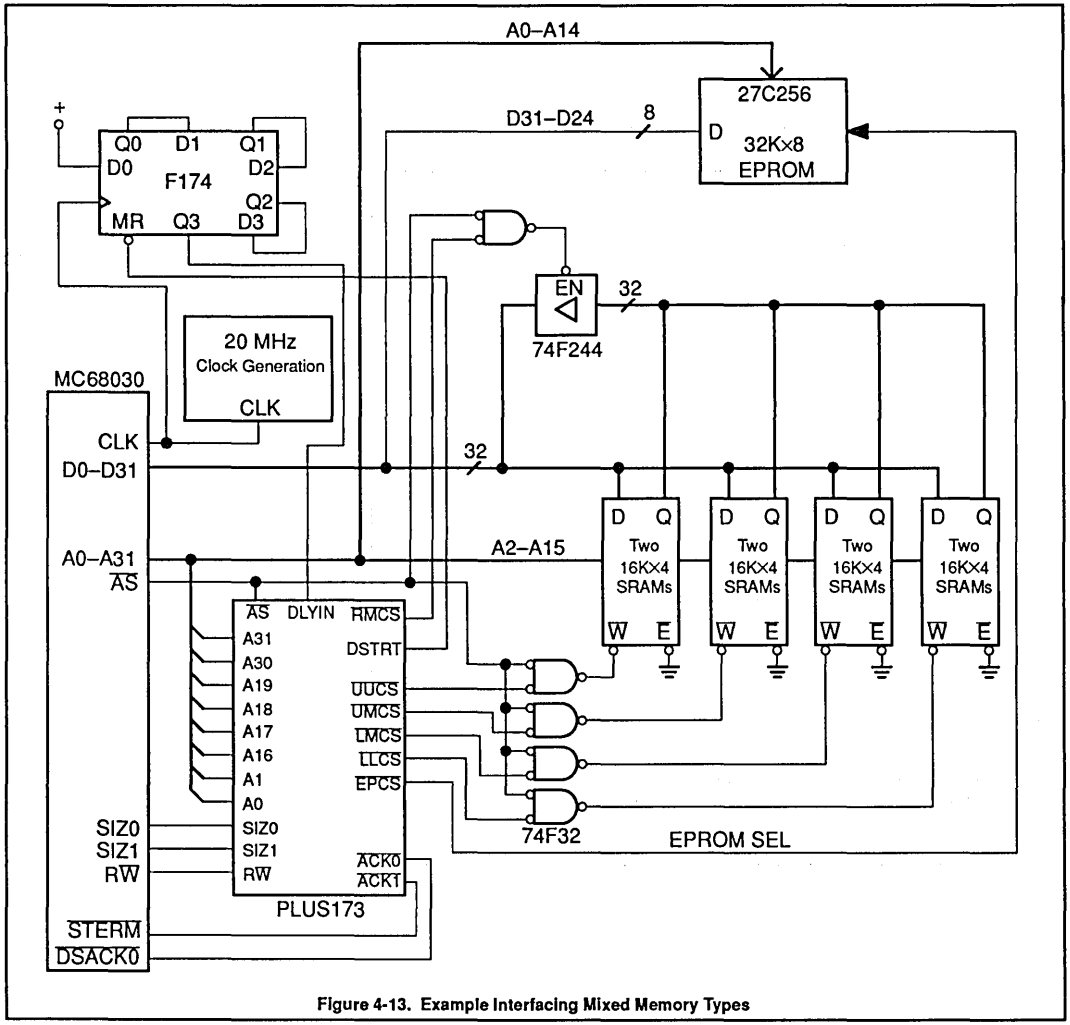


Figure 4-13. Example Interfacing Mixed Memory Types

Programmable logic design and application notes

```
####################   P I N   L I S T   ####################

LABEL** FNC **  PIN----------PIN    ** FNC  **   LABEL
DLYIN** I   **   1-|         |-24    ** +5V  **   VCC
/AS  ** I   **   2-|         |-23    ** /O   **   /RMS
A31  ** I   **   3-|    P    |-22    ** O    **   DSTRT
A30  ** I   **   4-|    L    |-21    ** /O   **   /UUCS
A19  ** I   **   5-|    U    |-20    ** /O   **   /UMCS
A18  ** I   **   6-|    S    |-19    ** /O   **   /LMCS
A17  ** I   **   7-|    1    |-18    ** /O   **   /LLCS
A16  ** I   **   8-|    7    |-17    ** /O   **   /EPCS
A1   ** I   **   9-|    3    |-16    ** /O   **   /ACK1
A0   ** I   **  10-|         |-15    ** /O   **   /ACK0
SIZ0** I    **  11-|         |-14    ** I    **   RW
GND  ** 0V  **  12-|         |-13    ** I    **   SIZ1
                   -----------


@DEVICE TYPE
PLUS173
@LOGIC EQUATION
  "EPROM enable"
/epcs = /(/a31*/a30*/a19*/a18*/a17*/a16*as);
  "start shift register during EPROM access"
dstrt = /a31*/a30*/a19*/a18*/a17*/a16*as;
  "DSACK0 after 4 clock cycles for EPROM access"
/ack0 = /(dlyin);
  "immediate STERM upon RAM access"
/ack1 = /(/a31*/a30*/a19*/a18*/a17*a16);
  "Byte select signals for RAM writes"
/uucs = /(a0*/a1*          /rw*a16*/a17*/a18*/a19*/A30*/a31);
/umcs = /( a0*/a1*          /rw*a16*/a17*/a18*/a19*/A30*/a31
         +/a1*/siz0*        /rw*a16*/a17*/a18*/a19*/A30*/a31
         +/a1*/siz1*        /rw*a16*/a17*/a18*/a19*/A30*/a31;
/lmcs = /( a0*/a1*          /rw*a16*/a17*/a18*/a19*/A30*/a31
         +/a1*/siz0*/siz1*/rw*a16*/a17*/a18*/a19*/A30*/a31
         +/a1* siz0* siz1*/rw*a16*/a17*/a18*/a19*/A30*/a31
         +/a1*a0*/siz0*     /rw*a16*/a17*/a18*/a19*/A30*/a31);
/llcs = /( a0* a1*          /rw*a16*/a17*/a18*/a19*/A30*/a31
         +/a0* siz0* siz1*/rw*a16*/a17*/a18*/a19*/A30*/a31
         +    /siz0*/siz1*/rw*a16*/a17*/a18*/a19*/A30*/a31
         + a1*        siz1*/rw*a16*/a17*/a18*/a19*/A30*/a31);
/rmcs = /(                  rw*a16*/a17*/a18*/a19*/A30*/a31);
```

Figure 4-14. AMAZE Pinlist and Boolean Equations

Programmable logic design and application notes

The 29000, SRAM and the PLUS20L8D

The 29000 processor can, in theory, access instruction memory every 40 nanoseconds (Figure 4-15) at top speed. The natural tendency is to place this part into the highest performance environment possible. This is still a very difficult problem. Similar to the SPARC, a simple brute force SRAM will yield no "wait states" by correctly combining, for instance, a 25ns access time with the instruction cycle. One additional, and very important requirement will be a current drive of 24mA. The PLUS20L8D PAL (Figure 4-16) can decode up to 20 inputs in 10 nanoseconds with 24mA of output drive.

Making some assumptions about SRAM input capacitance, p.c. board trace capacitance, etc., will assure the reader that the PLUS20L8D will not waste valuable time needed to achieve the maximum possible performance. The following outlines verification that the PLUS20L8D will meet timing requirements.

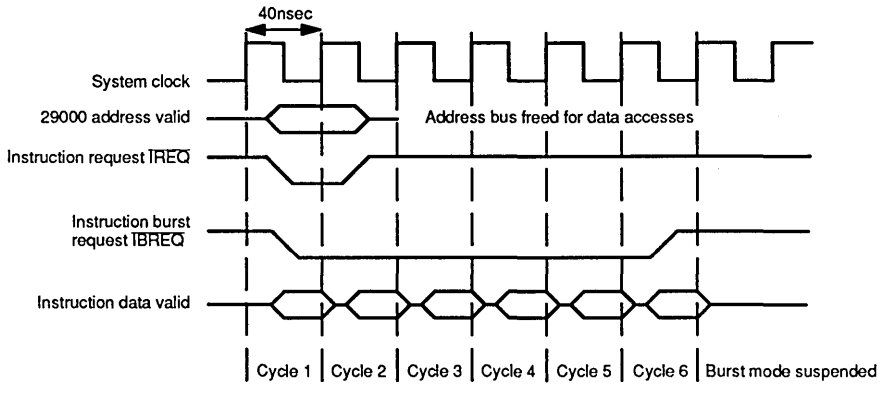


Figure 4-15. 29000 Burst Mode Instruction Timing

Programmable logic design and application notes

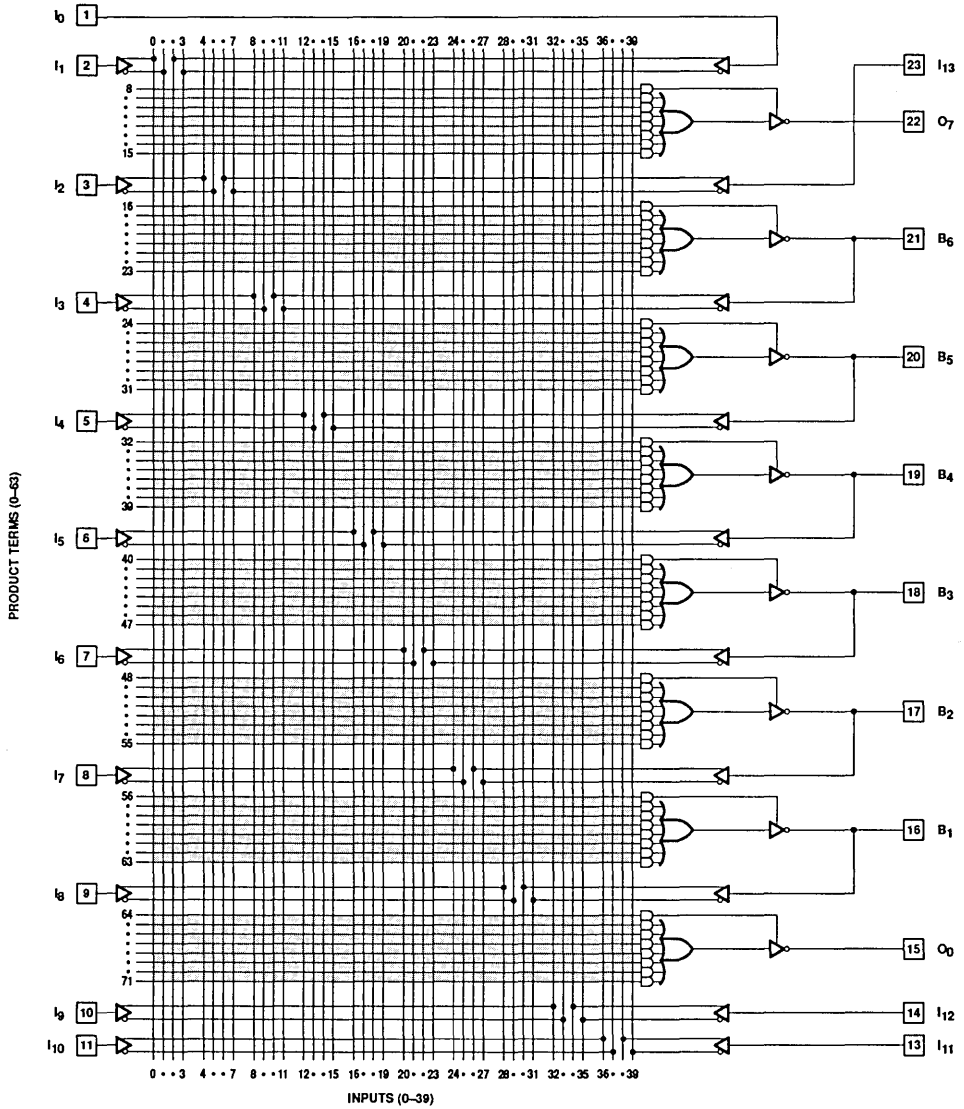


Figure 4-16. Signetics 20L8D

Programmable logic design and application notes

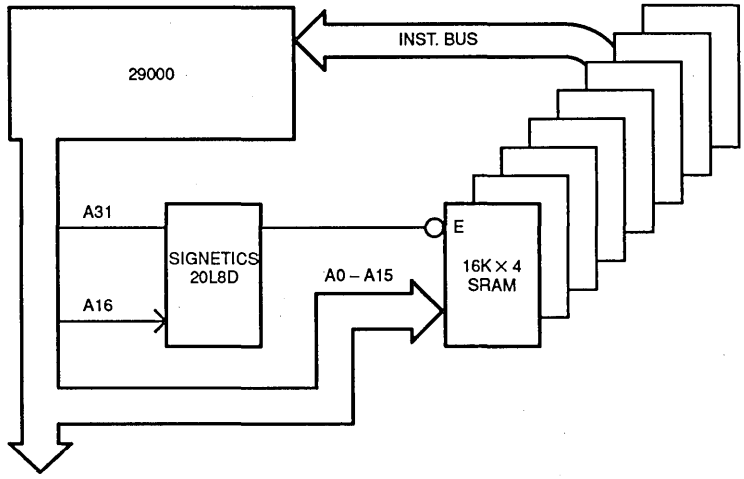


Figure 4-17. High Speed Burst Mode Configuration

Consider, for example, the configuration depicted in Figure 4-17 with Motorola MCM 6288 SRAMs. As depicted, there are eight modules forming a 16K word space. The 29000 is targeted to drive 80pf according to the data manual at full speed (25MHz clock). How much additional time delay will be attributed to the bus and RAM loading?

For the chosen RAM configuration C_{in} = 8 (5pf) = 40pf. We will have to assume some values for p.c. wiring. One common one today is about 20pf/ft. Placing our RAM on the same board, near the processor should require about a foot of trace/address line. The average C_L = (40 + 20) pf. This is just beyond the 10ns specified for testing the 20L8 (i.e., 50pf), however, if we ignore it, the extra loading will not significantly impact this small system.

To include it would incur additional assumptions about the pull up and pull down resistance of the 20L8 (these values are typically between 5 and 20 ohms), but are not strictly specified. The result is that we are within 2 nanoseconds of time delay by ignoring the capacitance.

Tdelay	(20L8)	= 10ns (MAX)
Tdelay	(RAM access)	= 25ns (MAX)
Tdelay	(29000)	= 5ns (MAX)
		40ns (MAX)

Adding anything into the data path or a poor circuit layout can take the design out of spec., but by these assumptions, is can succeed. Design slack may be generated simply by inserting the 20L8–7 and one gets 2.5ns of free time margin.

Programmable logic design and application notes

Interrupt Handler — PLS179

In the 1970's I.C. manufacturers made the error of introducing microprocessor chips without having family support chips available. Often, months or years passed before relatively simple family additions arrived. Later, a similar situation arose when bus standards, LAN standards, and disk standards failed to settle down for system designers to get sufficient market lead without LSI solutions. PLDs could have helped much here! As an example of designing a microprocessor family part, consider Figure 4-18, which depicts an interrupt handler. In particular, note that interrupt inputs will be latched into an 8-bit register. This in turn will be encoded to a 3-bit vector which may be appropriately enabled and applied to the microbus. Figure 4-18 shows the eight flip-flops as having J-K and /D inputs which will be generated with a PLS179 by switching the flip-flop control. Appropriate control signals for the various transactions might be as follows:

1. CLOCK – the system synchronous time base.
2. Interrupt Enable – when asserted high from the microprocessor, allows interrupts to be generated to the microprocessor.
3. Interrupt – a strobe or level defined to indicate a pending interrupt and a valid encoded vector.
4. Interrupt Acknowledge – a response signal from the microprocessor which may be used to enable the 3-bit vector onto the bus. As well, it may initiate clearing the currently asserted interrupt latch.
5. /INT0–/INT7 – eight possible interrupt request signals which must be asserted low and held there until service for that device has occurred.
6. Reset – this is a system override signal which will clear all flip-flops during initial operation.

Basic Operation

Initially, the part should be reset by asserting the RESET pin high, asynchronously. Then, when interrupts are enabled, the /D-inputs to the 8 flip-flops will be synchronously scanning for interrupt inputs (asserted low). This will put a nonzero value into the eight bit register which will generate an interrupt output, combinationally through the Complement array. In parallel, a 3-bit encoded vector will be applied on the VEC0, VEC1, VEC2 lines. Asserted high logic will be assumed for the vector. Presumably, a microprocessor will interrupt this, transfer control to a service routine and clear the interrupt. The clear will be accomplished by disabling interrupts and strobing the vector value back into the PLS179, using the IACK signal. Disabling the interrupts will put the registers into J-K mode. J is tied to zero and K is decoded from the specifically strobed vector. Therefore, synchronous clear of the high priority bit is done. Interrupts are then re-enabled and the process continues.

The PLS179 solution offers room for user alteration. For example, the IACK condition could be redefined as a combination of the Z80 IOREQ and M1 signals, or any specific splitting of internal signals could be easily done. The design could fit into a PLS159A, but there would be less room for variation for specific users exact needs. Figure 4-20 shows the pinlist for the handler. Figure 4-21 gives the corresponding design file. A simulation Ruler template is given in Figure 4-22, and Figure 4-23 shows a simulation log file for some example interrupt transactions. In Figure 4-23, the simulation begins by asserting RESET followed by successive assertion of each interrupt bit to demonstrate the vector encoding. The second half begins with all eight interrupts asserted simultaneously and each is cleared successively in descending priority. /INT0 is the highest priority. The interrupt is actually asserted through the PLS179 Complement array behaving as a simple NOR gate.

Programmable logic design and application notes

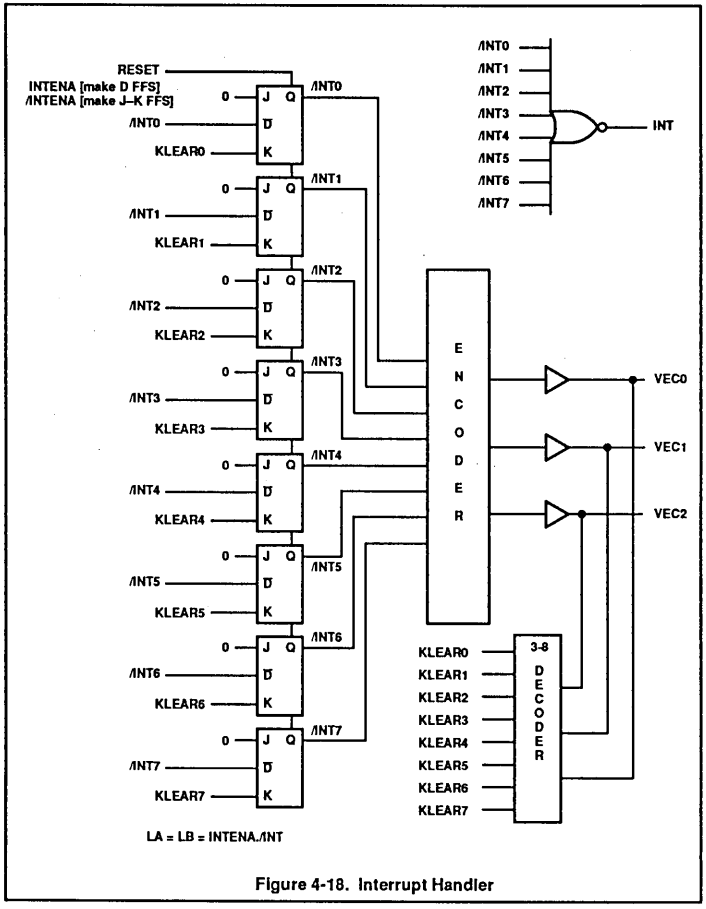


Figure 4-18. Interrupt Handler

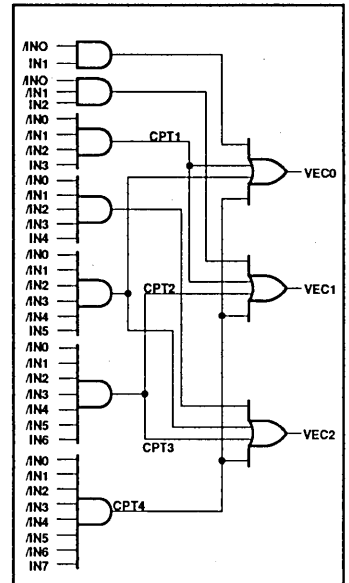


Figure 4-19. Encoder Logic Diagram

Programmable logic design and application notes

```
@DEVICE TYPE
PLS179
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
JOE USER
@NAME
GENERIC INTERRUPT VECTOR GENERATOR
@DESCRIPTION
@PINLIST

"<--------FUNCTION-------->        <--REFERENCE-->"
"PINLABEL   PIN #    PIN_FCT       PIN_ID    OE_CTRL"
CLOCK        1         CK          CLK       -    ;
ENA          2         I           I0        -    ;
RESET        3         I           I1        -    ;
IACK         4         I           I2        -    ;
N/C          5         I           I3        -    ;
N/C          6         I           I4        -    ;
N/C          7         I           I5        -    ;
N/C          8         I           I6        -    ;
N/C          9         I           I7        -    ;
INTERUPT    10         B           B0        D0   ;
VEC0        11         B           B1        D1   ;
GND         12         0V          GND       -    ;
OTE         13         /OE         /OE       -    ;
VEC1        14         B           B2        D2   ;
/INT0       15         /B          F0        EA   ;
/INT1       16         /B          F1        EA   ;
/INT2       17         /B          F2        EA   ;
/INT3       18         /B          F3        EA   ;
/INT4       19         /B          F4        EB   ;
/INT5       20         /B          F5        EB   ;
/INT6       21         /B          F6        EB   ;
/INT7       22         /B          F7        EB   ;
VEC2        23                     B3        D3   ;
VCC         24         +5V         VCC       -    ;
```

Figure 4-20. Interrupt Pin List

Programmable logic design and application notes

```
@COMMON PRODUCT TERM
CPT1=    /INT0*/INT1*/INT2*INT3 ;
CPT2=    /INT0*/INT1*/INT2*/INT3*/INT4*INT5 ;
CPT3=    /INT0*/INT1*/INT2*/INT3*/INT4*/INT5*INT6 ;
CPT4=    /INT0*/INT1*/INT2*/INT3*/INT4*/INT5*/INT6*INT7 ;
KLEAR0   =/VEC*/VEC1*/VEC0*IACK; "DECODE VECTOR 0"
KLEAR1   =/VEC2*/VEC1*VEC0*IACK; "DECODE VECTOR 1"
KLEAR2   =/VEC2*VEC1*/VEC0*IACK; "DECODE VECTOR 2"
KLEAR3   =/VEC2*VEC1*VEC0*IACK; "DECODE VECTOR 3"
KLEAR4   =VEC2*/VEC1*/VEC0*IACK; "DECODE VECTOR 4"
KLEAR5   =VEC2*/VEC1*VEC0*IACK; "DECODE VECTOR 5"
KLEAR6   =VEC2*VEC1*/VEC0*IACK; "DECODE VECTOR 6"
KLEAR7   =VEC2*VEC1*VEC0*IACK ; "DECODE VECTOR 7"
@COMPLEMENT ARRAY
/C = /(INT0 + INT1 + INT2 + INT3 + INT4 + INT5 + INT6 + INT7);
@I/O DIRECTION
D3 = ENA;
D2 = ENA;
D1 = ENA;
D0 = ENA;
@FLIP FLOP CONTROL
FC = /ENA;
@OUTPUT ENABLE
EA=OTE;
EB=OTE;
@REGISTER LOAD
LA=ENA;
LB=ENA;
@ASYNCHRONOUS PRESET/RESET
RA = RESET;
RB = RESET;
PA = 0;
PB = 0;
@FLIP FLOP MODE
M0,M1,M2,M3,M4,M5,M6,M7 = 1;
@LOGIC EQUATION
VEC0 = (/INT0*INT1 + CPT1 + CPT2 + CPT4);
VEC1 = (/INT0*/INT1*INT2 + CPT1 + CPT3 +CPT4);
VEC2 = (/INT0 * /INT1 * /INT2 */INT3*INT4 + CPT2 + CPT3 + CPT4);
INTERRUPT = (/C);
INT0:    J=0;
         K=KLEAR0;
INT1:    J=0;
         K=KLEAR1;
INT2;    J=0;
         K=KLEAR2;
INT3:    J=0;
         K=KLEAR3;
INT4:    J=0;
         K=KLEAR4;
INT5:    J=0;
         K=KLEAR5;
INT6:    J=0;
         K=KLEAR6;
INT7:    J=0;
         K=KLEAR7;
```

Figure 4-21. Interrupt Design File

```
@INPUTS
13 /OTE
3 RESET
2 INTENABLE

22 /INT7
21 /INT6
20 /INT5
19 /INT4
18 /INT3
17 /INT2
16 /INT1
15 /INT0

4 IACK

23 VEC2
14 VEC1
11 VEC0

@OUTPUTS
10 INTERRUPT
23 VEC2
14 VEC1
11 VEC0

22 /INT7
21 /INT6
20 /INT5
19 /INT4
18 /INT3
17 /INT2
16 /INT1
15 /INT0
```

Figure 4-22. Interrupt Ruler File

Programmable logic design and application notes

```
PLS179 INTERRUP          Time = 14:53:31    Date = 8/5/1987
"
"   C / INP .            I/O      <=PREV=>   <=NEXT=>   <=FOUT=>   TRACE TERMS
"   L O
"   K E 76543210         3210     76543210   76543210   76543210
"
    C 0 11111111         LLLH     LLLLLLLL   LLLLLLLL   NNNNNNNN  ;
    C 0 11111010         ....     LLLLLLLL   LLLLLLLL   HHHHHHHH  ;
    C 1 11111001         HHHL     LLLLLLLL   HLLLLLLL   ........  ;
    C 1 11111001         HHLL     HLLLLLLL   LHLLLLLL   ........  ;
    C 1 11111001         HLHL     LHLLLLLL   LLHLLLLL   ........  ;
    C 1 11111001         HLLL     LLHLLLLL   LLLHLLLL   ........  ;
    C 1 11111001         LHHL     LLLHLLLL   LLLLHLLL   ........  ;
    C 1 11111001         LHLL     LLLLHLLL   LLLLLHLL   ........  ;
    C 1 11111001         LLHL     LLLLLHLL   LLLLLLHL   ........  ;
    C 1 11111001         LLLL     LLLLLLHL   LLLLLLLH   ........  ;
    C 1 11111001         LLLL     LLLLLLLH   HHHHHHHH   ........  ;
    C 0 11111100         000.     HHHHHHHH   HHHHHHHL   LLLLLLLH  ;
    C 0 11111100         001.     HHHHHHHL   HHHHHHLL   LLLLLLHH  ;
    C 0 11111100         010.     HHHHHHLL   HHHHHLLL   LLLLLHHH  ;
    C 0 11111100         011.     HHHHHLLL   HHHHLLLL   LLLLHHHH  ;
    C 0 11111100         100.     HHHHLLLL   HHHLLLLL   LLLHHHHH  ;
    C 0 11111100         101.     HHHLLLLL   HHLLLLLL   LLHHHHHH  ;
    C 0 11111100         110.     HHLLLLLL   HLLLLLLL   LHHHHHHH  ;
    C 0 11111100         111.     HLLLLLLL   LLLLLLLL   HHHHHHHH  ;
"
"        -------X        ----      I/O CONTROL LINES
"                        BBBB      DESIGNATED I/O USAGE
"                        BBBO      ACTUAL I/O USAGE
"
"   PINLIST...
"   01 13 09 08 07 06 05 04 03 02 23 14 11 10
"   00 00 00 00 00 00 00 00 00 00 00 00 00 00
"   22 21 20 19 18 17 16 15 ;
```

Figure 4-23. Interrupt Simulation Log File

Programmable logic design and application notes

SCSI TARGET INTERFACE — PLUS105 AND PLUS153B

Overview

This application provides a complete solution to the SCSI Target Interface. As well, it includes a detailed rendering of the PLUS105 controlling transaction with a complete state equation solution.

Introduction

From its first introduction, the SCSI Bus (known as SASI in its initial days), has gained wide acceptance as a small computer peripheral bus. As the performance capabilities of mass storage peripheral devices increased, other bus specifications came into being to handle the increased performance requirements. Interfaces such as the High Speed SCSI, ESDI, and proposed byte/word wide bus for high performance mass storage devices (to replace the de facto standard SMD Interface) are gaining acceptance. Though different from each other, they present the system designer with surprisingly similar handshake requirements for the transfer of command, status, data and other information among hosts and/or targets connected to the bus.

In recent years several IC manufacturers have introduced single-chip controllers for the SCSI Bus, but none yet for the new proposed buses. The purpose of this application note is to use the SCSI Bus, known to most designers, as the vehicle to demonstrate the ease with which such buses can be handled by high performance, low cost programmable sequencers. The design described is based on the PLUS105 (or the higher performance PLUS405).

High performance programmable sequencers using the architecture exemplified in the PLUS105 have been available since Signetics invented and introduced the PLS105 in 1980.

Functional Description

The SCSI Interface described in this document is a Full Target Implementation that includes the following features:

- Arbitration Capability
- Reselect Capability
- Software Programmable Target I.D.
- Full DMA Interface
- Interrupt Generator

The Reselect and Arbitration capabilities enable the implementation of an efficient, intelligent target controller. Once a command is received, the target can disconnect from the SCSI Bus, execute the command and reconnect to the SCSI when data or status needs to be transferred to the requester. This reduces the amount of idle time on the bus; it also enables the target to receive multiple commands and execute them in the most efficient manner.

The software programmable Target I.D. allows the same design to be used for multiple targets sharing the same SCSI Bus. The DMA Interface is based on a straight-forward DMA Request/DMA Acknowledge Handshake protocol, enables fast data transfers without undue burden on the local intelligence.

An open collector, active low interrupt is provided to request service by the local intelligence at the completion of transfers or in the event of errors.

Programmer's Interface

The SCSI Port is operated through the use of five independently addressed registers: STATUS, COMMAND, TARGET ID, DATA IN, DATA OUT.

ADDRESS	REGISTER	ACCESS MODE
Base + 0	STATUS	Read only
Base + 0	COMMAND	Write only
Base + 1	TARGET ID	Write only
Base + 2	DATA IN	Read only
Base + 2	DATA OUT	Write only

The 5 registers are 8 bits wide with bit definitions as described below:

COMMAND REGISTER

IRQE	PORTE	DMAE	CTLS	MESG	CTRL	SFC	
7	6	5	4	3	2	1	0

IRQE – SET to enable the generation of interrupts from the Port.

PORTE – SET to enable operation of the SCSI Port. Negation of this signal is required to clear the interrupt generated at the completion of every command.

DMAE – SET to enable the DMA Interface of the SCSI Port.

CTLE – Defines the direction of transfers on the SCSI Bus
0 : From SCSI to TARGET
1 : From TARGET to SCSI

MESG – SET for MESSAGE Transfers on the SCSI Bus

Programmable logic design and application notes

CTRL – Defines the type of transfer on the SCSI Bus
0 : DATA Transfer
1 : COMMAND or STATUS Transfer

SFC – Two bit Function Code:
0 0 – Disconnect
0 1 – Transfer
1 0 – Reselect
1 1 – Arbitrate

STATUS REGISTER

CMPL	PARE	SRST	ATTN	SLCT	BUSY	ARBT	SLCD
7	6	5	4	3	2	1	0

CMPL – Function Completed. This signal gated with BIT 7 (IRQE) of the Command Register generates an Interrupt to the local intelligence.

PARE – Parity Error. The source of the error can be determined from the state of BIT 4 (CTLS) in the COMMAND Register:
CTLS=0 – Error on SCSI Bus
CTLS=1 – Error in local memory

SRST – Status of RESET signal on SCSI Bus

ATTN – Status of ATTENTION signal on SCSI Bus

SLCT – Status of SELECT signal on SCSI Bus

BUSY – Status of BUSY signal on SCSI Bus

ARBT – ASSERTED to indicated that the controller has won Arbitration and is in control of the SCSI Bus.

SLCD – SELECTED. If both SLCD and SLCT are ASSERTED, the controller is being selected by another device on the SCSI Bus.

TARGET ID REGISTER

not used					TID		
7	6	5	4	3	2	1	0

TID – Three bit code that defines the Target I.D.

Port Operation

As described in the previous section, the port can execute 4 commands:
Arbitrate, Reselect, Transfer and Disconnect.

Arbitrate : The port monitors the SCSI Bus for the "bus free" state. When the bus is free, the port starts the Arbitration sequence to gain bus mastership. If arbitration is won, the port will generate an interrupt with the appropriate status in the STATUS Register. If arbitration is lost, the port returns to the monitoring of the SCSI Bus.

Reselect : The port transfers the desired ID Byte from local memory (through DMA) to the SCSI Bus and waits for the appropriate response from the desired controller. When the desired controller responds, the Port generates an interrupt with the appropriate status in the STATUS Register.

Transfer : The port transfers data between local memory and the selected controller on the SCSI Bus until the DMA Termination Signal (DMACNT0) is asserted. At completion, the port generates an interrupt with the proper status in the STATUS Register.

Disconnect : The port relinquishes Bus mastership. This results in the "bus free" state allowing other controllers to use the SCSI Bus. This is also the IDLE state for the Port. The port should be programmed for this state when there is no SCSI work in progress.

Programmable logic design and application notes

Arbitration Software Sequence

```
PROC (arbitrate)
   negate DMAE, PORTE
   set SFC to ARBITRATE; assert PORTE
   ———— wait for completion ————
   negate PORTE
   IF port won arbitration
     THEN exit with normal status
     ELSE DO
       IF SRST
         THEN exit with RESET status
         ELSE DO
           set–up single byte DMA transfer
           negate CTLS; assert DMAE, PORTE
           ———— wait for completion ————
           negate DMAE, PORTE
           exit with Port Selected status
         END
     END
   END
```

RESELECT Software Sequence

```
PROC (reselect)
   negate DMAE, PORTE
   place reselect ID byte in local memory
   set–up single byte DMA transfer
   set SFC to RESELECT; assert DMAE, PORTE
   ———— wait for completion ————
   negate DMAE, PORTE
   IF good completion
     THEN exit with normal status
     ELSE IF SRST
       THEN exit with reset status
       ELSE exit with error status
END
```

TRANSFER Software Sequence

```
PROC (transfer)
   negate DMAE, PORTE
   set–up DMA controller
   set SFC to TRANSFER; set–up CTRL, CTLS, MESG; assert DMAE, PORTE
   ———— wait for completion ————
   negate DMAE, PORTE
   IF good completion
     THEN exit with normal status
     ELSE IF SRST
       THEN exit with reset status
       ELSE exit with error status
```

DISCONNECT Software Sequence

```
PROC (disconnect)
   negate DMAE, PORTE
   set SFC to DISCONNECT; assert PORTE
   ———— wait for completion ————
   negate PORTE
   exit with normal status
END
```

Programmable logic design and application notes

Hardware Description

The SCSI Interface described in this document is implemented using three Programmable Logic Devices and a hand–full of FAST SSI/Octals. Referring to the schematic in Figure 4-30 (page 550), the functions of the different components are as follows:

U6 – (74F244) Port STATUS Register

U7 – (74F273) Port COMMAND Register

U1 – (74F374) DATA OUT Register

U2 – (74F374) DATA IN Register

U8 – (74F273) : SCSI Bus signals Synchronization
: Partial Status Latch

U4 – (74145) Asserts proper signal on SCSI bus during Arbitration.

U5 – (PLUS153B) : Register Decode
: 3-bit TARGET ID Register.

U3 – (PLUS153B) :Parity Generator/Checker
:Arbitration Win Detection
:Port Selected Detection

U9 – (PLUS105) : Executes all commands
: Controls handshake with DMA controller
: Controls REQ/ACK Handshake with SCSI Bus
: Detects "bus free" state
: Implements "arbitration delay"

74F38's – High Current, Open Collector Drivers for SCSI Bus

74F14's – Schmitt Trigger Input Receivers for SCSI Bus

NOTES:

1. The interface requires an 8MHz Clock. The throughput of the interface can be increased by operating this circuit at 24MHz by using the PLUS405.
2. The interface is initialized by an active low signal: /SYSRESET
3. The DMA Interface consists of four signals:

   DMAENBL – Software controlled DMA Enable
   DMAREQ – asserted by the port (PLS105A) for each byte transfer
   /DMACYCLE – asserted by DMA controller as a response to DMAREQ
   /DMACNT0 – asserted when the DMA transfer count reaches 0
4. The processor (local intelligence) interface consists of 5 signals

   A1, A0 – The two least significant address bits
   /SYSSEL – A block decode signal for the SCSI Port
   /SYSREAD – Active low, READ signal
   0 – READ
   1 – WRITE
   /INTERRUPT– Active low, Open collector Interrupt

Programmable logic design and application notes

```
@DEVICE TYPE
PLUS105

@DRAWING                    DNW-SIG-105
@REVISION                   A
@DATE                       11-29-87
@SYMBOL                     U9
@COMPANY                    SIGNETICS
@NAME                       DIMITRIOUS DOUROS
@DESCRIPTION                SCSI CONTROLLER

@PINLIST

"<--------FUNCTION--------->   <--REFERENCE-->"
"PINLABEL   PIN #   PIN_FCT   PIN_ID     OE_CTRL"
8MHZ            1      CK     CLK            -   ;
/DMACNT0        2      I      I7             -   ;
PORTENB         3      I      I6             -   ;
SELECTED        4      I      I5             -   ;
WONARB          5      I      I4             -   ;
BBUSY           6      I      I3             -   ;
BSELECT         7      I      I2             -   ;
BACK            8      I      I1             -   ;
BRESET          9      I      I0             -   ;
CBUSY          10      O      F7             /OE ;
CSELECT        11      O      F6             /OE ;
REQUEST        12      O      F5             /OE ;
/ARBITRATE     13      O      F4             /OE ;
GND            14      0V     GND            -   ;
SFCMPL         15      O      F3             /OE ;
SPARERR        16      O      F2             /OE ;
SDRVENB        17      O      F1             /OE ;
DMAREQ         18      O      F0             /OE ;
LOW            19      /OE    PR/OE          -   ;
ICBUSY         20      I      I15            -   ;
ICSELECT       21      I      I14            -   ;
IREQUEST       22      I      I13            -   ;
PARERROR       23      I      I12            -   ;
CTLSCSI        24      I      I11            -   ;
SFC1           25      I      I10            -   ;
SFC0           26      I      I9             -   ;
/DMACYCLE      27      I      I8             -   ;
VCC            28      +5V    VCC            -   ;
```

Figure 4-24. PLUS105 SCSI_CTL Pinlist

Programmable logic design and application notes

```
@INTERNAL SR FLIP FLOP LABELS
Q0 Q1 Q2 Q3 Q4 Q5

@COMMON PRODUCT TERM

@COMPLEMENT ARRAY

@LOGIC EQUATION
@DEVICE SELECTION
SCSI_CTL/PLS105

@INPUT VECTORS

[PORTENB,SFC1,SFC0]

    "COMMAND CODE DEFINITIONS"

DISCONNECT      = 100B;
DATA_XFER       = 101B;
RESELECT        = 110B;
ARB_COMMAND     = 111B;

@OUTPUT VECTORS

[CBUSY,CSELECT,REQUEST,/ARBITRATE,SFCMPL,SPARERR,
SDRVENB,DMAREQ]

    "DISCONNECT STATE OUTPUTS"

DISCNCT_OUT'   = 00011000B;

    "POWER-UP STATE OUTPUTS"

POWER_UP_OUT' = 00010000B;
@STATE VECTORS
[Q5,Q4,Q3,Q2,Q1,Q0]

    "INITIALIZATION, IDLE, AND DON'T CARE STATES"

POWER_UP        = 3FH;
IDLE= 1FH;
ANY_STATE       = ------B;


    "PORT SELECTED RESPONSE STATES"

SELECTED_1      = 3CH;
SELECTED_2      = 18H;
SELECTED_3      = 19H;
SELECTED_4      = 1AH;


    "RESELECT STATE SEQUENCE"

RESELECT_1      = 10H;
RESELECT_2      = 11H;
RESELECT_3      = 12H;
RESELECT_4      = 13H;
RESELECT_5      = 3BH;
RESELECT_6      = 33H;
```

Figure 4-25.1 PLUS105 SCSI_CTL .SEE File (continued)

Programmable logic design and application notes

```
    "ARBITRATION STATE SEQUENCE"

ARBITRATE_1       = 00H;
ARBITRATE_2       = 0FH;
ARB_DELAY_GO    = 20H;
ARB_DELAY_IP    = 10----B;
ARB_DELAY_QU    = 2EH;

    "DATA TRANSFER SEQUENCE"

DATA_XFER_1       = 14H;
DATA_XFER_2       = 15H;
DATA_XFER_3       = 16H;
DATA_XFER_4       = 17H;
DATA_XFER_5       = 34H;
DATA_XFER_SL    = 1AH;

    "EVENT COMPLETION SEQUENCE"

COMPLETE_1      = 1CH;
COMPLETE_2      = 1EH;

@TRANSITIONS

 "------------ POWER-UP RESET RESPONSE ---------------"

WHILE     [POWER_UP]
    IF    []
          THEN [IDLE] WITH [POWER_UP_OUT']


    "------------ SCSI BUS RESET RESPONSE ---------------"

WHILE     [ANY_STATE]
    IF    [BRESET]
          THEN [COMPLETE_1] WITH [DISCNCT_OUT']


      "------------ DISCONNECT SEQUENCE ---------------"

WHILE     [IDLE]
    IF    [/BRESET*DISCONNECT*ICBUSY]
          THEN [COMPLETE_1] WITH [DISCNCT_OUT']
```

Figure 4-25.2 PLUS105 SCSI_CTL .SEE File (continued)

Programmable logic design and application notes

```
"------------- RESELECT SEQUENCE ---------------"

WHILE    [IDLE]
    IF   [/BRESET*/BSELECT*/BBUSY*RESELECT*ICBUSY*ICSELECT]
         THEN [RESELECT_1] WITH [CBUSY',CSELECT',SDRVENB',DMAREQ']

    WHILE    [RESELECT_1]
       IF    [/BRESET*DMACYCLE]
             THEN [RESELECT_2] WITH [/DMAREQ']

    WHILE    [RESELECT_2]
       IF    [/BRESET]
             THEN [RESELECT_3]

    WHILE    [RESELECT_3]
       IF    [/BRESET]
             THEN [RESELECT_4]

    WHILE    [RESELECT_4]
       IF    [/BRESET*ICBUSY]
             THEN [RESELECT_5]

    WHILE    [RESELECT_5]
       IF    [/BRESET*ICBUSY]
             THEN [RESELECT_5] WITH [/CBUSY']
       IF    [/BRESET*/ICBUSY]
             THEN [RESELECT_6]

    WHILE    [RESELECT_6]
       IF    [/BRESET*/ICBUSY*BBUSY]
             THEN [COMPLETE_1] WITH [CBUSY',/CSELECT',SFCMPL']


"------------ PORT SELECTED RESPONSE ---------------"

WHILE    [IDLE]
    IF   [/BRESET*PROTENB*/ICBUSY*/ICSELECT*/BBUSY*BSELECT*SELECTED*/PARERROR]
         THEN  [SELECTED_1] WITH [SFCMPL',DMAREQ']

    WHILE    [SELECTED_1]
       IF    [/BRESET*/PORTENB*DMACYCLE]
             THEN [SELECTED_2] WITH [CBUSY',/SFCMPL',SDRVENB',/DMAREQ']

    WHILE    [SELECTED_2]
       IF    [/BRESET]
             THEN [SELECTED_3]

    WHILE    [SELECTED_3]
       IF    [/BRESET]
             THEN [SELECTED_4] WITH [SFCMPL']

    WHILE    [SELECTED_4]
       IF    [/BRESET*PORTENB]
             THEN [COMPLETE_1]
```

Figure 4-25.3 PLUS105 SCSI_CTL .SEE File (continued)

Programmable logic design and application notes

```
"------------ ARBITRATION SEQUENCE ---------------"

WHILE    [IDLE]
    IF   [/BRESET*ARB_COMMAND*/ICBUSY*/BBUSY*/BSELECT]
         THEN [ARBITRATE_1]

    WHILE    [ARBITRATE_1]
       IF    [/BRESET*(BBUSY+BSELECT)]
             THEN [IDLE]
       IF    [/BRESET*/BBUSY*/BSELECT]
             THEN [ARB_DELAY_GO] WITH [CBUSY',ARBITRATE',SDRVENB']

    WHILE    [ARB_DELAY_IP]
       IF    [/BRESET*Q0]
             THEN [/Q0]
       IF    [/BRESET*/Q0]
             THEN [Q0]
       IF    [/BRESET*Q1*Q0]
             THEN [/Q1]
       IF    [/BRESET*/Q1*Q0]
             THEN [Q1]
       IF    [/BRESET*Q2*Q1*Q0]
             THEN [/Q2]
       IF    [BRESET*/Q2*Q1*Q0]
             THEN [Q2]
       IF    [/BRESET*/Q3*Q2*Q1*Q0]
             THEN [Q3]
       IF    [/BRESET*ARB_DELAY_QU]
             THEN [ARBITRATE_2]

WHILE    [ARBITRATE_2]
    IF   [/BRESET*/BSELECT*WONARB]
         THEN [COMPLETE_1] WITH [CSELECT',/ARBITRATE',SFCMPL']
    IF   [/BRESET*BSELECT]
         THEN [IDLE] WITH [/CBUSY',/ARBITRATE',/SDRVENB']


"---------------- DATA TRANSFER SEQUENCE ---------------"

WHILE    [IDLE]
    IF   [/BRESET*ICBUSY*/ICSELECT*DATA_XFER*CTLSCSI]
         THEN [DATA_XFER_1] WITH [REQUEST']
    IF   [/BRESET*ICBUSY*/ICSELECT*DATA_XFER*/CTLSCSI]
         THEN [DATA_XFER-1] WITH [DMAREQ']

    WHILE    [DATA_XFER_1]
       IF    [BRESET*PROTENB*IREQUEST*BACK]
             THEN [DATA_XFER_1] WITH [DMAREQ']
       IF    [/BRESET*PORTENB*DMACYCLE]
             THEN [DATA_XFER_2] WITH [/DMAREQ']

    WHILE    [DATA_XFER_2]
       IF    [/BRESET]
             THEN [DATA_XFER_3]

    WHILE    [DATA_XFER_3]
       IF    [/BRESET]
             THEN [DATA_XFER_4]
```

Figure 4-25.4 PLUS105 SCSI_CTL .SEE File (continued)

Programmable logic design and application notes

```
    WHILE    [DATA_XFER_4]
       IF    [/BRESET*PORTENB*/PARERROR*DMACNTO*/CTLSCSI]
             THEN [COMPLETE_1] WITH [SFCMPL']
       IF    [/BRESET*PORTENB*/PARERROR*SMACNTO*CTLSCSI]
             THEN [DATA_XFER_5] WITH [REQUEST']
       IF    [/BRESET*PORTENB*/PARERROR*/SMACNTO]
             THEN [DATA_XFER_1] WITH [REQUEST']
       IF    [/BRESET*/PORTENB]
             THEN [DATA_XFER_SL] WITH [SFCMPL']
       IF    [/BRESET*PORTENB*PARERROR]
             THEN [COMPLETE_1] WITH [SFCMPL',SPARERR']

    WHILE    [DATA_XFER_5]
       IF    [/BRESET*IREQUEST*BACK]
             THEN [COMPLETE_1] WITH [SFCMPL']


 "---------------- COMMAND COMPLETION SEQUENCE ----------------"

WHILE    [COMPLETE_1]
    IF   [/BRESET*/PORTENB]
         THEN [COMPLETE_2] WITH [/SFCMPL',/SPARERR']

WHILE    [COMPLETE_2]
    IF   [/BRESET*PORTENB]
         THEN [IDLE]

WHILE    [IDLE]
    IF   [/BRESET*/BSELECT*/ICBUSY*/SFC1*/SFC0]
         THEN [IDLE]
```

Figure 4-25.5 PLUS105 SCSI_CTL .SEE File (end)

Programmable logic design and application notes

```
@DEVICE TYPE

PLUS153

@DRAWING.............SIG_APN88_02
@REVISION............A
@DATE................8-8-88
@SYMBOL..............U3
@COMPANY.............SIGNETICS
@NAME................ASP APPLICATIONS GROUP
@DESCRIPTION.........CSI TARGET CONTROLLER. ARBITRATION/SELECTION LOGIC

@PINLIST

"<--------FUNCTION--------> <--REFERENCE-->"
"PINLABEL  PIN #  PIN_FCT     PIN_ID       OE_CTRL"
SD0          1       I          I0          -     ;
SD1          2       I          I1          -     ;
SD2          3       I          I2          -     ;
SD3          4       I          I3          -     ;
SD4          5       I          I4          -     ;
SD5          6       I          I5          -     ;
SD6          7       I          I6          -     ;
SD7          8       I          I7          -     ;
PAR1         9       O          B0          D0    ;
GND         10       0V         GND         -     ;
PAR2        11       O          B1          D1    ;
ID0         12       I          B2          D2    ;
ID1         13       I          B3          D3    ;
ID2         14       I          B4          D4    ;
A-WONARB    15       O          B5          D5    ;
A-SELECTED  16       O          B6          D6    ;
PARERROR    17       O          B7          D7    ;
GEN-PAR     18       O          B8          D8    ;
REC-PAR     19       I          B9          D9    ;
VCC         20       +5V        VCC         -     ;
```

Figure 4-26. PLUS153 SCSI_ARB Pinlist

Programmable logic design and application notes

```
@COMMON PRODUCT TERM
MAX-ID-MATCH = ID1 * ID1 * ID2 * SD7 ;
@I/O DIRECTION
@LOGIC EQUATION
A-SELECTED    =  ID0   * /ID1   * /ID2   * SD1
              + /ID0   *  ID1   * /ID2   * SD2
              +  ID0   *  ID1   * /ID2   * SD3
              + /ID0   * /ID1   *  ID2   * SD4
              +  ID0   * /ID1   *  ID2   * SD5
              + /ID0   *  ID1   *  ID2   * SD6
              +  MAX-ID-MATCH ;
A-WONARB      =  ID0   * /ID1   * /ID2   * SD1    * /SD2   * /SD3   * /SD4   * /SD5   * /SD6 *
/SD7
              + /ID0   *  ID1   * /ID2   *  SD2   * /SD3   * /SD4   * /SD5   * /SD6   * /SD7
              +  ID0   *  ID1   * /ID2   *  SD3   * /SD4   * /SD5   * /SD6   * /SD7
              + /ID0   * /ID1   *  ID2   *  SD4   * /SD5   * /SD6   * /SD7
              +  ID0   * /ID1   *  ID2   *  SD5   * /SD6   * /SD7
              + /ID0   *  ID1   *  ID2   *  SD6   * /SD7
              +  MAX-ID-MATCH ;
              " PARITY GENERATOR ( PAR1 AND PAR2 ARE PARTIAL TERMS )."
PAR1          =  SD0   * /SD1   * /SD2
              + /SD0   *  SD1   * /SD2
              + /SD0   * /SD1   *  SD2
              +  SD0   *  SD1   *  SD2 ;
PAR2          =  SD3   * /SD4   * /SD5
              + /SD3   *  SD4   * /SD5
              + /SD3   * /SD4   *  SD5
              +  SD3   *  SD4   *  SD5 ;
GEN-PAR       = PAR1   * PAR2   * /SD6   * /SD7
              + PAR1   */PAR2   *  SD6   * /SD7
              + PAR1   */PAR2   * /SD6   *  SD7
              + PAR1   * PAR2   *  SD6   *  SD7
              + /PAR1  */PAR2   * /SD6   * /SD7
              + /PAR1  * PAR2   *  SD6   * /SD7
              + /PAR1  * PAR2   * /SD6   *  SD7
              + /PAR1  */PAR2   *  SD6   *  SD7 ;
"PARITY ERROR GENERATOR. ERROR FLAGGED IF RECEIVED PARITY IS DIFFERENT
                      FROM CALCULATED PARITY."
PARERROR = /GEN-PAR * REC-PAR
             + GEN-PAR * /REC-PAR ;
```

Figure 4-27. PLUS153 SCSI_ARB .BEE File

Programmable logic design and application notes

```
@DEVICE TYPE

PLUS153

@DRAWING..............SIG_APN88_02
@REVISION..............A
@DATE..................8-8-88
@SYMBOL................U5
@COMPANY...............SIGNETICS
@NAME..................ASP APPLICATIONS GROUP
@DESCRIPTION...........SCSI TARGET CONTROLLER. REGISTER CONTROL LOGIC

@PINLIST

"<--------FUNCTION-------->   <--REFERENCE-->"
"PINLABEL  PIN #  PIN_FCT      PIN_ID   OE_CTRL"
/DMACYCLE   1        I          I0        -     ;
A0          2        I          I1        -     ;
A1          3        I          I2        -     ;
/SYSREAD    4        I          I3        -     ;
/SYSSEL     5        I          I4        -     ;
DB0         6        I          I5        -     ;
DB1         7        I          I6        -     ;
DB2         8        I          I7        -     ;
SDRVENB     9        I          B0        D0    ;
GND        10        0V         GND       -     ;
CTLSCSI    11        I          B1        D1    ;
/WRCTRL    12        /O         B2        D2    ;
/RDSTAT    13        /O         B3        D3    ;
/RDDATA    14        /O         B4        D4    ;
ID0        15        O          B5        D5    ;
ID1        16        O          B6        D6    ;
ID2        17        O          B7        D7    ;
BUFENBL    18        O          B8        D8    ;
/WRDATA    19        /O         B9        D9    ;
VCC        20        +5V        VCC       -     ;
```

Figure 4-28. PLUS153 SCSI_DCD Pinlist

Programmable logic design and application notes

```
@COMMON PRODUCT TERM
@I/O DIRECTION
@LOGIC EQUATION
BUFENBL = SDRVENB ;


                        SCSI CONTROLLER REGISTER MAP

 /DMACYCLE | CTLSCSI | /SYSSEL | /SYSREAD | A1 | A0 |       FUNCTION
-----------+---------+---------+----------+----+----+-----------------------
      H    |    X    |    L    |    L     | 0  | 0  | READ STATUS REG.
      H    |    X    |    L    |    H     | 0  | 0  | WRITE CONTROL REG.
      H    |    X    |    L    |    L     | 0  | 1  | DATA BUFFER    - READ
      H    |    X    |    L    |    H     | 0  | 1  |                - WRITE
      H    |    X    |    L    |    L     | 1  | X  | -- NOT VALID --
      H    |    X    |    L    |    H     | 1  | X  | SET TARGET ID
      L    |    L    |    X    |    X     | X  | X  | DMA : SCSI -> SYSTEM
      L    |    H    |    X    |    X     | X  | X  | DMA : SYSTEM -> SCSI

/RDSTAT = / ( /DMACYCLE * SYSSEL * /A0 * /A1 * SYSREAD ) ;

/WRCTRL = / ( /DMACYCLE * SYSSEL * /A0 * /A1 * /SYSREAD ) ;

/RDDATA = / ( /DMACYCLE * SYSSEL * A0 * /A1 * SYSREAD
              + DMACYCLE * /CTLSCSI ) ;

/WRDATA = / ( /DMACYCLE * SYSSEL * A0 */A1 * /SYSREAD
              + DMACYCLE * CTLSCSI ) ;

          "   ID0-2 ARE THE TARGET ID CODE. THE ID REGISTER IS IMPLEMENTED
              IN THE PLUS153 BY SUPPLYING A SET TERM ( WITH DB0-2 ) AND A
              HOLD TERM ( WITH ID0-2 ).
          "
ID0 =     /DMACYCLE   * SYSSEL   * A1   * /SYSREAD        * DB0
    + /(  /DMACYCLE   * SYSSEL   * A1   * /SYSREAD )      * ID0 ;
ID1 =     /DMACYCLE   * SYSSEL   * A1   * /SYSREAD        * DB1
    + /(  /DMACYCLE   * SYSSEL   * A1   * /SYSREAD )      * ID1 ;
ID2 =     /DMACYCLE   * SYSSEL   * A1   * /SYSREAD        * DB2
    + /(  /DMACYCLE   * SYSSEL   * A1   * /SYSREAD )      * ID2 ;
```

Figure 4-29. PLUS153 SCSI_DCD .BEE File

Programmable logic design and application notes

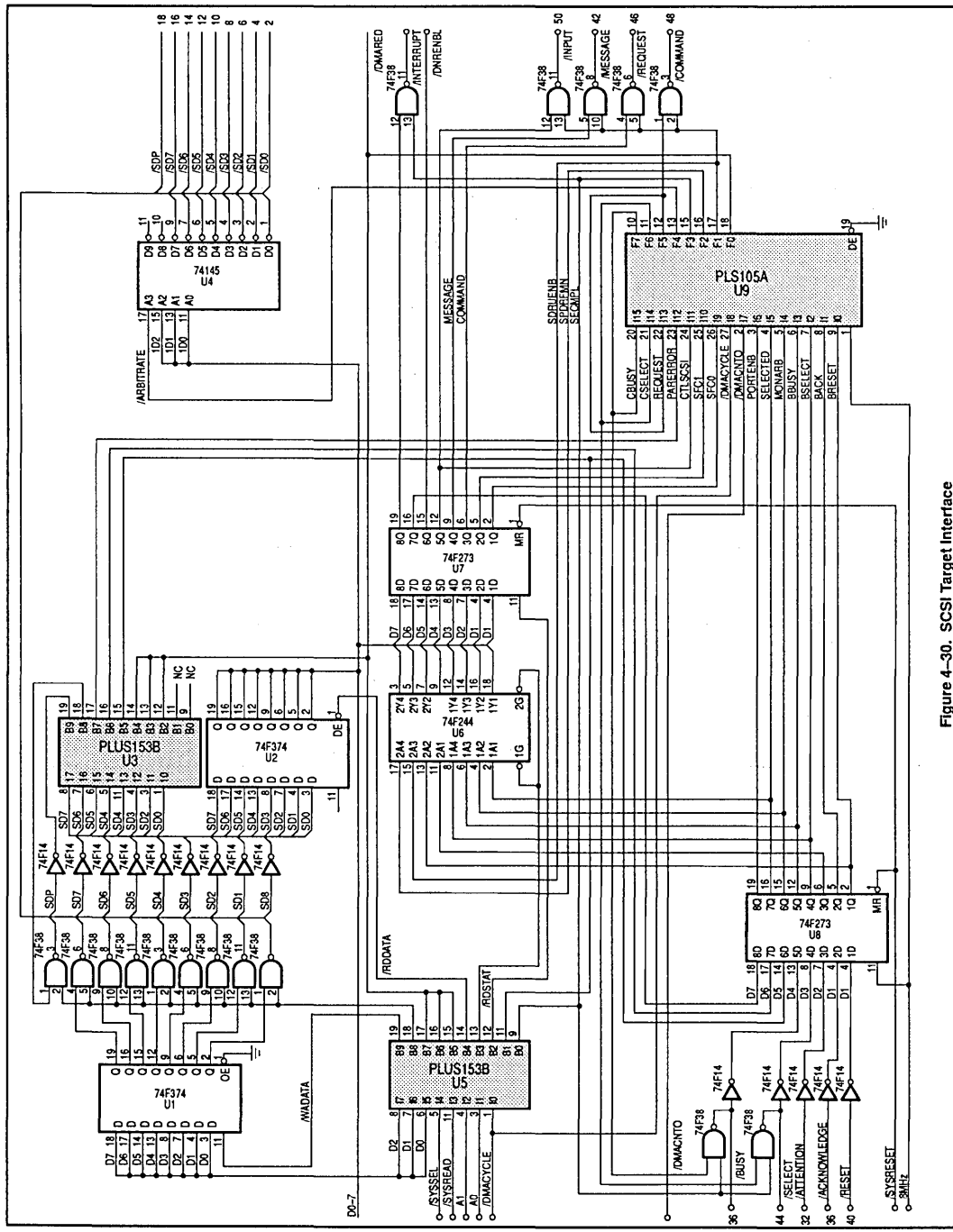


Figure 4-30. SCSI Target Interface

Programmable logic design and application notes

Programmable Logic Devices

COMMUNICATIONS USING PLDs

PLD devices are particularly appropriate for digital communications. High speed sequencers form a natural means of handshaking and protocol checking, where PLAs and PALs can decode parallel header information. But, PLDs fill a need for digital communications—that of the emerging "standard". Once an initial specification for a communication protocol is agreed upon, the manufacturers may generate product to meet the current specification. This will probably not embody the final specification, but will closely resemble it. Unfortunately, the penalty for having to redesign a gate array is relatively high should a communication protocol be implemented in one and require change. A PLD solution is an ideal embodiment for a product designed to implement an emerging standard because it can be changed when the "standard" changes.

This section covers several examples of data communication designs from whole protocols to simple scramblers—along with a customizable speech synthesis system using the Philips PCF8200.

The CCITT V.27 Scrambler – PLC18V8Z

The Radio Shack publication "Understanding Data Communications" contains a brief description of the CCITT V.27 recommended scrambler for use with 4800bps modems. The logic diagram for this circuit is a serial cascade of D flip-flops with Exclusive-OR gates tapped in and out of the data stream. This class of machines implements transformations based upon Galois polynomials which are often described by linear sequential machines (i.e., D-FFs and EX-OR gates).

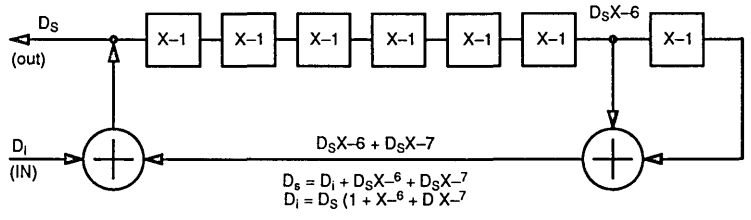


Figure 5-1. CCITT V.27 Scrambler

Programmable logic design and application notes

```
File Name:    CCITT V.27 Scrambler
Date:         10/31/1988
Time:         16:17:41

@DEVICE TYPE
PLC18V8Z
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST

"<--------FUNCTION--------->   <--REFERENCE-->"
"PINLABEL PIN #      PIN_FCT    PIN_ID      OE_CTRL"
CLOCK       1        CLK          I0/CLK       -   ;
DIN         2        I            I1           -   ;
N/C         3        I            I2           -   ;
N/C         4        I            I3           -   ;
N/C         5        I            I4           -   ;
N/C         6        I            I5           -   ;
N/C         7        I            I6           -   ;
N/C         8        I            I7           -   ;
N/C.        9        I            I8           -   ;
GND        10        0V           GND          -   ;
ENA        11        /OE          I9//OE       -   ;
X1         12        D            B0           D0  ;
X2         13        D            B1           D1  ;
X3         14        D            B2           D2  ;
X4         15        D            B3           D3  ;
X5         16        D            B4           D4  ;
X6         17        D            B5           D5  ;
X7         18        D            B6           D6  ;
DOUT       19        B            B7           D7  ;
VCC        20        +5V          VCC          -   ;

@COMMON PRODUCT TERM "CPT_LABEL = (expression)"
@I/O DIRECTION
    D7 = 1 ;
@LOGIC EQUATION
    DOUT = X7*/X6*/DIN + /X7*X6*/DIN + X7*X6*DIN + /X7*/X6*DIN ;
    X1:  D = DOUT   ;
    X2:  D = X1     ;
    X3:  D = X2     ;
    X4:  D = X3     ;
    X5:  D = X4     ;
    X6:  D = X5     ;
    X7:  D = X6     ;
```

Figure 5-2. V.27 Scrambler

Programmable logic design and application notes

A Novel Speech Synthesizer – PLS159A

The goal of this design was to build a small board capable of transmitting a variety of verbal messages in conjunction with the Philips PCF8200 speech synthesis part. Judicious partitioning resulted in a PLS159A acting as a controller, an HC4040 counter addressing an EPROM and an EPROM which provides binary data to the PCF8200.

The PLS159A transacts with the PCF8200 (Busy, etc.) and controls pulsing to the HC4040. Upon asserting the pushbutton, the cycle begins and proceeds to advance the HC4040 in increments 3FF (HEX) consecutive addresses. Figure 5-6 shows the system in full detail. The ultimate signal comes from the 8200, and drives an op-amp which delivers the final signal to the speaker. Various R-C combinations implement filtering and a simple pot provides the level control.

Programming the EPROM was accomplished by capturing short messages on audio tape, downloading to a Philips PCF8200 development system which analyzes and compresses the data for efficient storage. The result is an EPROM file which needs 3FF HEX addresses to store about 10sec of speech.

Figure 5-9 shows the AMAZE state equation entry file for the PLS159A which handshakes with the 8200 and 4040 parts.

```
@DEVICE TYPE
PLS159
@DRAWING
.....................PLD CONTROLLER FOR PCF8200 SPEECH SYNTHESIZER
@REVISION
.....................B
@DATE
.....................9/07/88
@SYMBOL
.....................PLS159
@COMPANY
.....................SIGNETICS CORPORATION
@NAME
@DESCRIPTION
This circuit will perform most functions required to achieve a minimum
configuration PCF8200 speech synthesizer system.

    REV-A **Original design modified to work in existing ASP demo board. ***
    REV-B **Fixed SEE/BEE file to eliminate random quits during utterances
           1)Gray code for sequencer.
           2)Input latch added on REQ signal from PCF8200 for
                 synchronization

FUNCTIONS PERFORMED:
           1.System Oscillator for sequencer controller
           2.Byte update control via EPROM to PCF8200

@PINLIST

"<---------FUNCTION---------> <--REFERENCE-->"
"PINLABEL   PIN #  PIN_FCT     PIN_ID         OE_CTRL"
6_MHZ_IN      1        CK          CK             -    ;
REQ           2        I           I0             -    ;
BUSY          3        I           I1             -    ;
END_ADRS      4        I           I2             -    ;
N/C           5        I           I3             -    ;
R_C           6        /B          B0             D0   ;
6_MHZ_OUT     7        O           B1             D1   ;
START_REQ     8        I           B2             D2   ;
N/C           9        /B          B3             D3   ;
GND          10        0V          GND            -    ;
N/C          11        /OE         /OE            -    ;
SV0          12        /O          F0             EA   ;
SV1          13        /O          F1             EA   ;
SV2          14        /O          F2             EA   ;
N/C          15        /B          F3             EA   ;
REQLATCH     16        /O          F4             EB   ;
RESET        17        /O          F5             EB   ;
COUNT        18        /O          F6             EB   ;
WRITE        19        /O          F7             EB   ;
VCC          20        +5V         VCC            -    ;
```

Figure 5-3. PLS159A SPEECHB Pinlist

Programmable logic design and application notes

```
@COMMON PRODUCT TERM

@COMPLEMENT ARRAY
@I/O DIRECTION
DO = 6_MHZ_OUT;
@FLIP FLOP CONTROL
FC = 1;

@OUTPUT ENABLE
EA = 0 ;
EB = 0 ;
@REGISTER LOAD

@ASYNCHRONOUS PRESET/RESET
@FLIP FLOP MODE

@LOGIC EQUATION
"    ** OSCILLATOR SECTION "
6_MHZ_OUT = R_C;
R_C = /(1) ;
"LATCH FOR SYNCHRONIZATION
OF PLD AND SPEECH CHIP
CLOCKS"
"CAPTURE OF REQ SIGNAL FROM
PCF8200"

/REQLATCH :     K = /REQ;
                J = REQ;
```

Figure 5-4. PLS159A SPEECHB .BEE File

```
@DEVICE SELECTION
SPEECHB/PLS159

"REVISION-B"

@STATE VECTORS
[SV2 , SV1 , SV0]

S0   = 000 B;
S1   = 011 B;
S2   = 101 B;
S3   = 110 B;
S4   = 100 B;
S5   = 010 B;
S6   = 111 B;

@INPUT VECTORS
@OUTPUT VECTORS
@TRANSITIONS

"Initialization at powerup"

WHILE [ S0 ]
        IF [  ] THEN [ S1 ] WITH [ /RESET', WRITE', COUNT']
WHILE [ S6 ]
        IF [  ] THEN [ S1 ] WITH [ /RESET', WRITE', COUNT']

"Wait for Start switch to be depressed"

WHILE [ S1 ]
        IF [/START_REQ * /BUSY] THEN [ S2 ] WITH [/WRITE']

WHILE [ S2 ]
        IF [  ] THEN [ S3 ] WITH [ WRITE' , COUNT']

WHILE [ S3 ]
        IF [  ] THEN [ S4 ] WITH [/COUNT']

WHILE [ S4 ]
        IF [REQLATCH] THEN [ S5 ] WITH [/WRITE',COUNT']
        IF [END_ADRS + /BUSY] THEN [ S6 ] WITH [RESET']

WHILE [ S5 ]
        IF [/REQLATCH] THEN [ S3 ] WITH [ WRITE']
```

Figure 5-5. PLS159A SPEECHB .SEE File

Programmable logic design and application notes

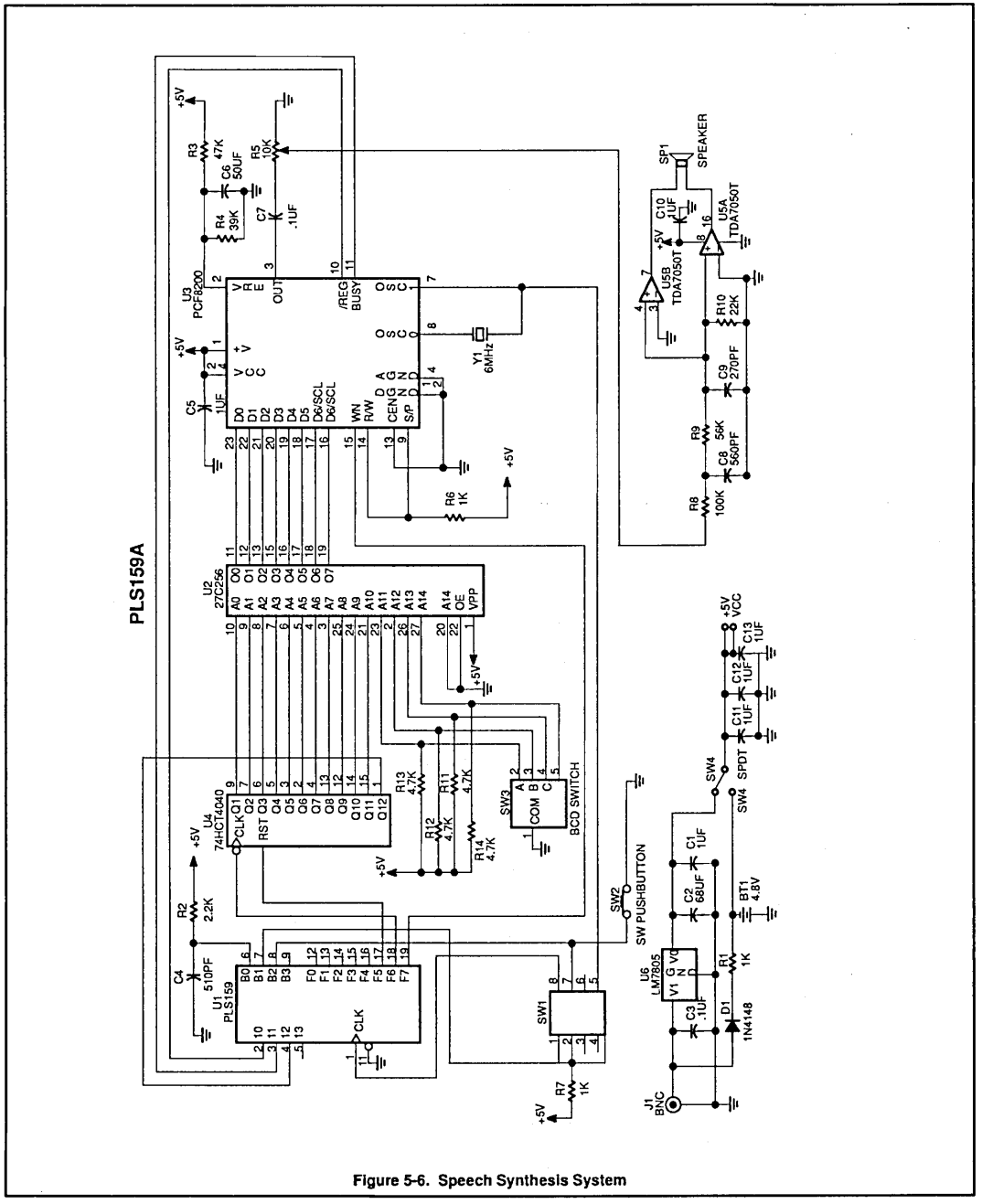


Figure 5-6. Speech Synthesis System

Programmable logic design and application notes

CCITT Forward CRC Polynomial – PLUS405

This application illustrates the use of Signetics PLUS405 in a high speed data communication application. Typically, larger polynomial encoders permit error checking over larger data streams than smaller ones. This design implements a sixteenth order polynomial and the figures that follow show the logic equations to implement it.

```
File Name : crc_gen
Date : 10/30/1987
Time : 11:10:56

####################   P I N   L I S T   ####################

LABEL    ** FNC **PIN---------- PIN **   FNC **  LABEL
CLK      ** CK  **  1-|        |-28  **   +5V **  VCC
N/C      ** I   **  2-|        |-27  **   I   **  G
N/C      ** I   **  3-|        |-26  **   I   **  N/C
D        ** I   **  4-|   P    |-25  **   I   **  N/C
N/C      ** I   **  5-|   L    |-24  **   I   **  N/C
XI13     ** I   **  6-|   U    |-23  **   I   **  XI12
XI14     ** I   **  7-|   S    |-22  **   I   **  XI11
XI15     ** I   **  8-|   4    |-21  **   I   **  XI10
XI16     ** I   **  9-|   0    |-20  **   I   **  XI9
XO16     ** O   ** 10-|   5    |-19  **   INT **  RESET
XO15     ** O   ** 11-|        |-18  **   O   **  XO9
XO14     ** O   ** 12-|        |-17  **   O   **  XO10
XO13     ** O   ** 13-|        |-16  **   O   **  XO11
GND      ** 0V  ** 14-|        |-15  **   O   **  XO12
                      ----------
```

Figure 5-7. PLUS405 Pinlist

```
File Name : crc_gen
Date : 10/30/1987
Time : 11:11:2

@DEVICE TYPE
PLUS405
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
This design implements the CRC-CCITT forward polynomial with a PLUS405.
The CRC-CCITT forward polynomial is

    GF(X) = 1 + X^5 + X^12 + X^16

    NOTE: Since this polynomial requires a 16-bit shift register together
    with an array of exclusive-OR's the contents of the 8 output flip-flops
    must be fed back into the array via the input pins of the PLUS405.
    Bits 1 to 8 are formed with the 8-bit internal register while the
    remaining 8-bits are made out of the 8-bit output register the outputs
    of which are externally fed back to inputs XI9 through XI16.

@INTERNAL FLIP FLOP LABELS
              X1, X2, X3, X4, X5, X6, X7, X8

@COMMON PRODUCT TERM

@COMPLEMENT ARRAY
@BURIED REGISTER CLOCK
@INIT/OE
    RESET =    R0, R1, R2, R3, R4, R5, R6, R7,
               R8, R9, R10, R11, R12, R13, R14, R15 ;
```

Figure 5-8. PLUS405 Design File

Programmable logic design and application notes

```
@LOGIC EQUATION
" ************************************************************
  *          X1 : INPUT = (D XOR XI16) * G                   *
  *                     = (/D * XI16 * G) + ( D * /XI16 * G); "*
  ************************************************************   "

  X1 : J = (D * /XI16 * G) + (/D * XI16 * G) ;
       K = /G + (D * XI16) + (/D * /XI16) ;

  X2 : J = X1    ;
       K = /X1   ;

  X3 : J = X2    ;
       K = /X2   ;

  X4 : J = X3    ;
       K = /X3   ;

  X5 : J = X4    ;
       K = /X4   ;

" ************************************************************
  *          X6 : INPUT = X5 XOR (( D XOR XI16 ) * G ) ; "     *
  ************************************************************   "

  X6 : J =  (/G * X5) + (/D * X5 * XI16) + (D * X5 * XI16)        +
            (G * D * /X5 * /XI16) + (G * /D * /X5 * XI16) ;
       K =  (/G * /X5) + (D * /X5 * XI16) + (/D * /X5 * /XI16)    +
            (G * D * X5 * /XI16) + (G * /D * X5 * XI16) ;

  X7 : J = X6    ;
       K = /X6   ;

  X8 : J = X7    ;
       K = /X7   ;

  XO9 :J = X8    ;
       K = /X8   ;

  XO10:J = XI9   ;
       K = /XI9  ;

  XO11:J = XI10  ;
       K = /XI10 ;

  XO12:J = XI11  ;
       K = /XI11 ;
" ************************************************************
  *          X13 :INPUT = X12 XOR (( D XOR XI16 ) * G ) ; "    *
  ************************************************************   "

XO13 : J =  (/G * XI12) + (/D * XI12 * XI16) + (D * XI12 * XI16)  +
            (G * D * /XI12 * /XI16) + (G */D * /XI12 * XI16) ;
       K =  (/G * XI12) + (D * /XI12 * XI16) + (/D * /XI12 * /XI16)+
            (G * D * XI12 * /XI16) + (G * /D * XI12 * XI16) ;

XO14 : J =  XI13 ;
       K =  /XI13;

XO15 : J =  XI14 ;
       K =  /XI14;

XO16 : J =  XI15 ;
       K =  /XI15;
```

Figure 5-9 PLUS405 Logic Equations

Programmable logic design and application notes

Programmable Logic Devices

INSTRUMENTATION

Instrumentation typically involves the measurement and often the display of physical world parameters. Digital systems are particularly effective in this area and as usual, are largely limited only by the designer's imagination. This section describes three distinct examples of instrumentation provided from customer interaction.

The last design involves implementing the sweep circuitry for an oscilloscope. This can be extended and modified for similar designs.

Another example is a pulse width monitor which can be used in controlling power pulses, radio strength or radar/sonar timing measurements.

The first example shows several parts being used to make a plethysmographic monitor (i.e., heart rate). This has straightforward medical and health applications. The beauty of PLD solutions to these problems is simple – they are readily modifiable for adaption for another end use. The oscilloscope circuit could be altered for a laser light show. The pulse width monitor could pick up a timing pulse from a disk and the plethysmographic system could be configured for animals rather than humans.

Heart Beat Monitor – PLS159A, PLS168A and PLS153

PLDs can be used as powerful building blocks in implementing the digital portion of a low cost portable heart beat monitor. This monitor is capable of displaying the heart beat in pulses per minute. Figure 6-1 shows the system block diagram. The digital portion is inside the dashed lines.

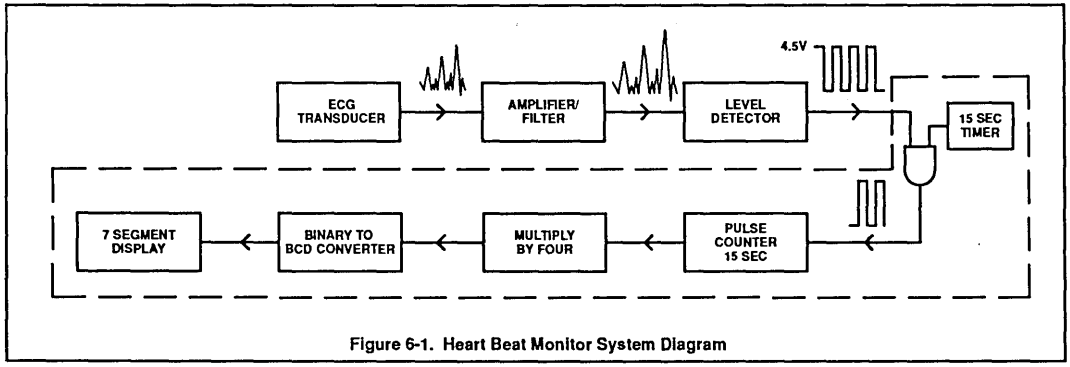


Figure 6-1. Heart Beat Monitor System Diagram

Programmable logic design and application notes

Operation

A transducer generates the heart beat pulses which are amplified and filtered. A level detector (one-shot) converts the amplified signals to TTL level signals. A 15 second timer is used to count the number of pulses in fifteen seconds. The number is multiplied by four to approximate the number of pulses in sixty seconds. A binary to BCD converter is used to display the result. A start switch resets the system and initiates the count.

Transducer

There are several techniques to monitor the blood flow in the peripheral system. These techniques include optical plethysmography, ultrasonic flow measurement, piezoelectric pickup of peripheral arterial pulse, Korotkoff sounds, and recording the ECG.

Light plethysmography is used as the transducer in this design. The Tektronix light plethysmograph (Figure 6-2), operates by measuring the reflectance of skin to red light. As blood flows into the skin's capillary bed with each heart beat, the reflectance changes and this change is amplified and observed.

Amplifier/Filter

Signetics SA741 OP-amp is used as bandpass filter with a gain of 20, and a frequency response of 1–200 Hz. Figure 6-3 shows the circuit diagram of the amplifier/filter stage.

Level Detector PLS153

The amplified signal is fed through the 'Level Detector' stage to create a square wave. A Schmitt-Trigger is used to generate the square wave. Application Note 18 in the Signetics PLD Data Manual explains the implementation of the Schmitt-Trigger in detail. A PLS153 is used to create the Schmitt Trigger. The PLS153 also holds the glue logic and other functions for the system explained further in this article (see Figure 6-4).

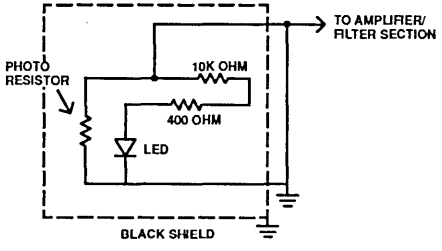


Figure 6-2. Tektronix Plethysmograph as Transducer

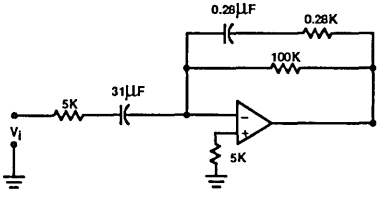


Figure 6-3. Amplifier/Filter Stage

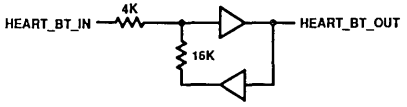


Figure 6-4. SCHMITT-TRIGGER Diagram

Programmable logic design and application notes

```
@DEVICE TYPE
PLS153
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST
"<---------FUNCTION---------->  <--REFERENCE-->"
"PINLABEL    PIN #    PIN_FCT    PIN_ID  OE_CTRL"
SET             1       I          I0       -  ;
RST             2       I          I1       -  ;
HEART_BT_IN     3       I          I2       -  ;
15_SEC          4       I          I3       -  ;
N/C             5       I          I4       -  ;
N/C             6       I          I5       -  ;
N/C             7       I          I6       -  ;
N/C             8       I          I7       -  ;
N/C             9       B          B0       D0 ;
GND            10       0V         GND      -  ;
RESET          11       O          B1       D1 ;
NTRESET        12       O          B2       D2 ;
NO_HRT_BT      13       O          B3       D3 ;
HEART_BT_OUT   14       O          B4       D4 ;
BT_IN_15_SEC   15       O          B5       D5 ;
N/C            16       B          B6       D6 ;
N/C            17       B          B7       D7 ;
N/C            18       B          B8       D8 ;
N/C            19       B          B9       D9 ;
VCC            20       +5V        VCC      -  ;
```

Figure 6-5. HTBT_153 Pinlist

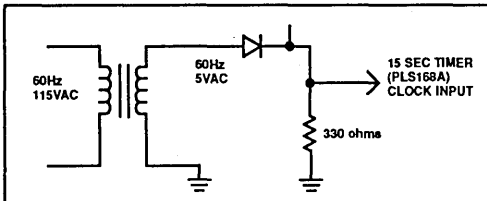


Figure 6-7. Clock Input for the PLS168

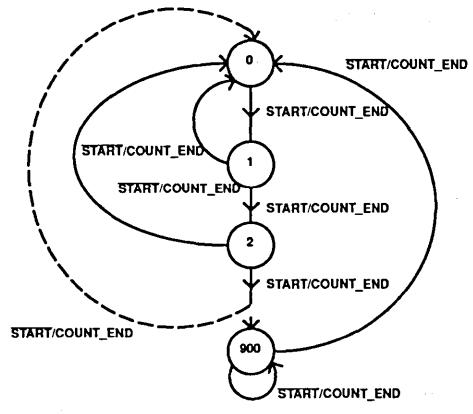


Figure 6-8. State Transition Diagram

```
@COMMON PRODUCT TERM
@I/O DIRECTION
@LOGIC EQUATION
"Level Detector (Schmitt Trigger)"
 HEART_BT_OUT = HEART_BT_IN;
"Debouncer"
 RESET         = /SET + /NTRESET;
 NTRESET       = /RESET + /RST;
"15 Second Timer ANDed with the output of the
Level Detector"
 BT_IN_15_SEC = 15_SEC * HEART_BT_OUT;
```

Figure 6-6. HTBT_153 .BEE File

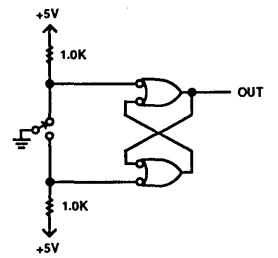


Figure 6-9. Circuit Diagram of Debouncer

Fifteen Second Timer

To create a fifteen second timer, a counter can be constructed with the PLS168A such that the number of counts is equivalent to 15 seconds. To achieve this, the 60Hz signal from the power line is passed through a transformer and a half-wave rectifier to create the clock input to the PLS168A (see Figure 6-7).

The number of counts needed to create the 15 second time interval is calculated in the following manner:

15 seconds = 15 X 60 (cycles/seconds) = 900

where '60' is the 60Hz clock input to the PLS168A.

Figure 6-8 shows the state transition diagram for the counter. Figure 6-10 is the timing diagram to generate the number of pulses in 15 seconds.

The reset switch is used to initiate the count. This switch is debounced using the PLS153 of Figure 6-4. Figure 6-9 shows the circuit diagram of the debouncer. The design equations for the debouncer are shown as part of the design equation files of Figure 6-6.

Programmable logic design and application notes

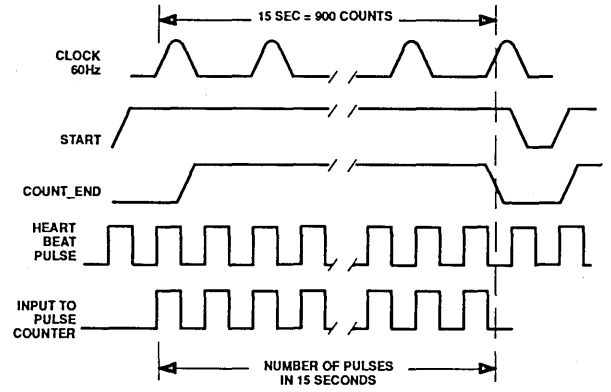


Figure 6-10. PLS168 Timing Diagram

If n = number of registers with feedback, then an n-bit counter can be created with any of Signetics sequencers using only n+1 terms.

Table 6-1 shows the implementation of the 15 second timer in the PLS168A. Input variable I0 is the input from the reset switch and '15_SEC' is the Count_End signal. At the 900th count (terms 12 & 13; HHHLLLLHLL), output '0' goes to a logic low, indicating that the end of the count has been reached.

```
Cust/Project     -15 Second Timer with Reset
Rev/I. D.        -
PLS168A

__                                                    ___________________
T !                                                   !       OPTION P/E        !L!
E !-------------------------------------------------------------------------------
R ! !     INPUT VARIABLE     !   PRESENT STATE    !     NEXT STATE      !OUTPUT !
M !C!1 1 ------------------------------------------------------------------------
__!_!1_0_9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!3_2_1_0!
 0!.!- - - -,- - - -,- - - H!- -,- - - -,- - - L!- -,- - - -,- - - H!- - - H!
 1!.!- - - -,- - - -,- - - H!- -,- - - -,- - L H!- -,- - - -,- - H L!- - - H!
 2!.!- - - -,- - - -,- - - H!- -,- - - -,- L H H!- -,- - - -,- H L L!- - - H!
 3!.!- - - -,- - - -,- - - H!- -,- - - -,L H H H!- -,- - - -,H L L L!- - - H!
 4!.!- - - -,- - - -,- - - H!- -,- - - L,H H H H!- -,- - - H,L L L L!- - - H!
 5!.!- - - -,- - - -,- - - H!- -,- - L H,H H H H!- -,- - H L,L L L L!- - - H!
 6!.!- - - -,- - - -,- - - H!- -,- L H H,H H H H!- -,- H L L,L L L L!- - - H!
 7!.!- - - -,- - - -,- - - H!- -,L H H H,H H H H!- -,H L L L,L L L L!- - - H!
 8!.!- - - -,- - - -,- - - H!- L,H H H H,H H H H!- H,L L L L,L L L L!- - - H!
 9!.!- - - -,- - - -,- - - H!L H,H H H H,H H H H!H L,L L L L,L L L L!- - - H!
10!.!- - - -,- - - -,- - - H!H H,H H H H,H H H H!L L,L L L L,L L L L!- - - H!
11!.!- - - -,- - - -,- - - L!- -,- - - -,- - - -!L L,L L L L,L L L L!- - - H!
12!A!- - - -,- - - -,- - - L!H H,H L L L,L H L L!L L,L - - -,- L - -!- - - H!
13!A!- - - -,- - - -,- - - H!H H,H L L L,L H L L!H H,H L L L,L H L L!- - - L!
14! !       ,       ,       !  ,       ,       !  ,       ,       !       !
15! !       ,       ,       !  ,       ,       !  ,       ,       !       !
16! !       ,       ,       !  ,       ,       !  ,       ,       !       !
17! !       ,       ,       !  ,       ,       !  ,       ,       !       !
18! !       ,       ,       !  ,       ,       !  ,       ,       !       !
19! !       ,       ,       !  ,       ,       !  ,       ,       !       !
20! !       ,       ,       !  ,       ,       !  ,       ,       !       !
*******************************************************************************
43!A!       ,       ,       !  ,       ,       !  ,       ,       !       !
44!A!       ,       ,       !  ,       ,       !  ,       ,       !       !
45!A!       ,       ,       !  ,       ,       !  ,       ,       !       !
46!A!       ,       ,       !  ,       ,       !  ,       ,       !       !
47!A!       ,       ,       !  ,       ,       !  ,       ,       !       !

                            R                                       1
                            E                                       5
                            S
                            E                                       S
                            T                                       E
                                                                    C
```

Table 6-1. PLS168A Program Table

Programmable logic design and application notes

Each line (or term) in Table 6-1 is part of the state transition of the 15 second counter/timer. Term 11 is used to reset the counter when the 'RESET' switch goes low. With Don't Cares (–) in the 'Present State' column, any time reset becomes low, the counter resets to zero, regardless of the counter's current state. Terms 0–10 create the counter. The complement array is used to avoid any undefined states and also to force the counter to a known state upon power up.

To count the number of heart beats in 15 seconds, the end of count output of the 15 second timer is ANDed with the heart beat pulses. The result of this 'AND' function is the number of pulses in 15 seconds. This 'AND' function is also implemented in the PLS153 of Figure 6-4.

Pulse Counter and Multiply by Four

The resources on a single PLS159A can be used to construct the 'pulse counter' and 'multiply by four' portion of the heart beat monitor. Figure 6-11 shows the block diagram of the Counter, Shift-Register, and an internal oscillator used as the clock for the shift-register.

To calculate the number of heart beats in one minute, the counter first counts the number of heart beats in 15 seconds. The counter is clocked by the 'NO_HRT_BT' signal which is the number of heart beats in 15 seconds.

When the fifteen seconds is over, the binary number stored in the counter is multiplied by four. The final value of this multiplication is the number of heart beats per minute.

The shift register multiplies the binary output of the counter by four by shifting this value twice to the left. During the 15 second period within which the counter is counting the number of heart beats, the shift register is disabled. When the 15 second period is over the counter is disabled and the shift register and the oscillator which clocks the shift register are enabled.

The shift register and the counter use the same flip-flops. When the PLS159A is in the 'counter' mode, the flip-flops are 'J-K' type. When the PLS159A is in the shift register mode, the flip-flops are switched to 'D' type.

CLOCK: The clock for the PLS159A is supplied from two sources. The number of heart beats clocks the device when the PLS159A is in the counter mode. The oscillator takes over when the device is in the shift register mode. The combinatorial outputs of the PLS159A can be used to create the oscillator. Application Note 13 (AN13) in the Signetics PLD Data Manual explains how this oscillator is implemented. Figure 6-12 shows the oscillator circuit diagram.

The oscillator output is enabled when the 15 second time period is over, so when the shift is over the outputs reflect the heart beat per minute in binary form. The maximum heart beat under extenuating circumstances can reach 300. Therefore, an 8-bit counter could measure a normal person's heart beat.

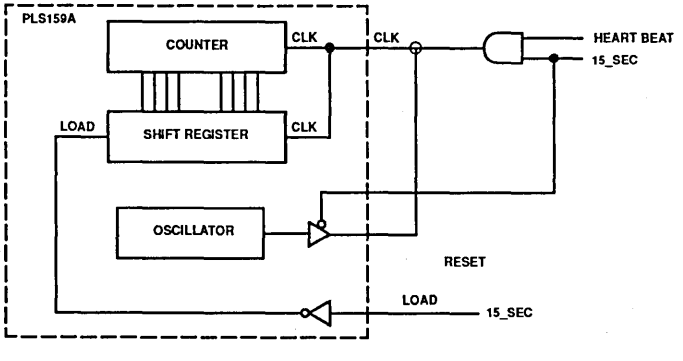


Figure 6-11. Counter Shift Register – Oscillator

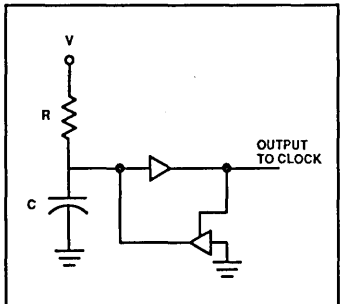


Figure 6-12. Oscillator Circuit

Programmable logic design and application notes

```
Binary Counter (Terms 0-7):      Counts the number of Heart Beat Pulses

Shift Register (Terms 8-14):     Shifts the final count by 2 thus
                                 multiplying the result by four.
Oscillator (Terms 16,17,D0,D1): Used to clock the PLS159 when in shift reg
                                 mode
Flip Flop Control (FC):          Switches the flip flop mode into 'D'
```

```
PLS159A                ! F/F TYPE       ! E(b)= ! E(a)= !POLARTY!
                       !________________!_______!_______!______!
 T !                   !A:A:A:A:A:A:A:A! .   .  ! .   .  !L:L:L:L!
 E !-------------------------------------------------------------
 R ! !   I    ! B(i)  !     Q(p)       !      Q(n)      ! B(o)  !
 M !C!-------------------------------------------------------------
___!_!3_2_1_0!3_2_1_0!7_6_5_4_3_2_1_0!7_6_5_4_3_2_1_0!3_2_1_0!
  0!-!- H - H!- - - -!- - - -,- - - -!- - - -,- - - - 0!. . . .!
  1!-!- H - H!- - - -!- - - -,- - - L!- - - -,- - - 0 -!. . . .!
  2!-!- H - H!- - - -!- - - -,- - L L!- - - -,- - 0 - -!. . . .!
  3!-!- H - H!- - - -!- - - -,- L L L!- - - -,- 0 - - -!. . . .!
  4!-!- H - H!- - - -!- - - -,L L L L!- - - 0,- - - - -!. . . .!
  5!-!- H - H!- - - -!- - - L,L L L L!- - 0 -,- - - - -!. . . .!
  6!-!- H - H!- - - -!- - L L,L L L L!- 0 - -,- - - - -!. . . .!
  7!-!- H - H!- - - -!- L L L,L L L L!0 - - -,- - - - -!. . . .!
  8! !     !      !      ,       !      ,       !      !
  9!-!- H - L!- - - -!- - - -,- - - H!. . . .,. A . .!. . . .!
 10!-!- H - L!- - - -!- - - -,- - H -!. . . .,A . . .!. . . .!
 11!-!- H - L!- - - -!- - - -,- H - -!. . . A,. . . .!. . . .!
 12!-!- H - L!- - - -!- - - -,H - - -!. . A .,. . . .!. . . .!
 13!-!- H - L!- - - -!- - - H,- - - -!. A . .,. . . .!. . . .!
 14!-!- H - L!- - - -!- - H -,- - - -!A . . .,. . . .!. . . .!
 15! !     !      !      ,       !      ,       !      !
 16!-!- - - -!- - H -!- - - -,- - - -!- - - -,- - - -!. . . A!
 17!-!- - - -!- - - -!- - - -,- - - -!- - - -,- - - -!. . A .!
 Fc!-!- - - H!- - - -!- - - -,- - - -!                       .
 Pb!-!- L - -!- - - -!- - - -,- - - -!                       .
 Rb! !     !      !      ,       !                           .
 Lb! !     !      !      ,       !                           .
 Pa!-!- L - -!- - - -!- - - -,- - - -!                       .
 Ra! !     !      !      ,       !                           .
 La! !     !      !      ,       !                           .
 D3! !     !      !      ,       !                           .
 D2! !     !      !      ,       !                           .
 D1!-!- - - -!- - - H!- - - -,- - - -!                       .
 D0!-!- - - L!- - - -!- - - -,- - - -!                       .
```

Table 6-2. Binary Counter, Shift Register, and Oscillator

Table 6-2 illustrates the implementation of the binary counter, shift register, and oscillator. 'I2' is the RESET signal input. 'I0' is the '15_SEC' signal input used to enable/disable the counter, shift register, and oscillator. 'B0' is the oscillator output used to clock the PLS159A when it is in the shift register mode.

Programmable logic design and application notes

Binary to BCD Converter

82S135 and 82S126 PROMs can be used to generate BCD code to drive three 7-segment displays. A look-up table is programmed in the PROMs to generate the correct BCD number. The 7-segment displays have their own decoders and display drivers. Figure 6-13 shows the overall system diagram.

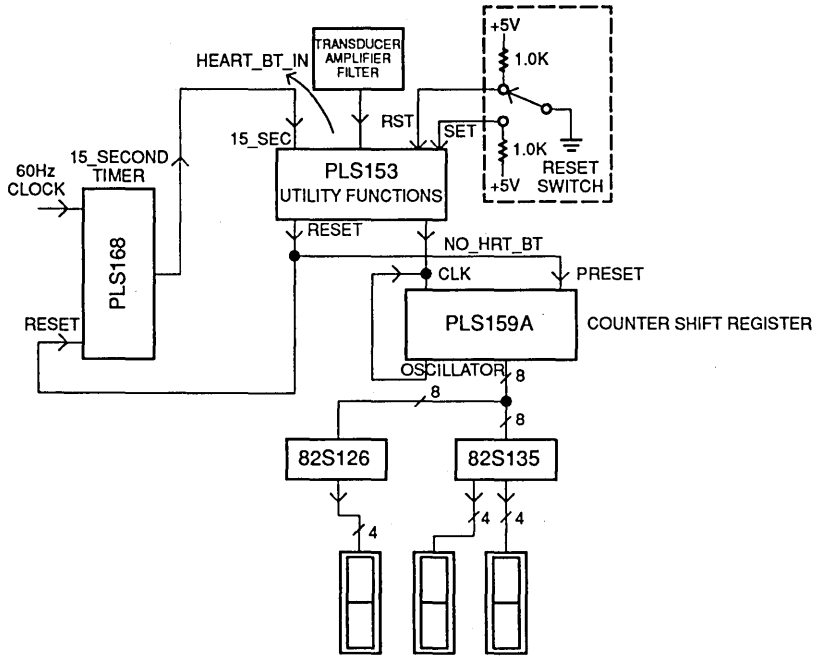


Figure 6-13. Overall System Diagram

Programmable logic design and application notes

The Pulse Width Monitor – PLS168

Simple in concept, this design was implemented at the fuse table level. Its elementary operation is that of a programmable timer which can detect a pulse condition exceeding a specified duration. This customer used the part in a system where the timer prevented the destruction of expensive, high-power equipment.

```
@DEVICE TYPE
PLS168/BCA
@DRAWING
TRANSMITTER FAULT MONITOR
@REVISION
@DATE
4/4/1988
@COMPANY
@NAME
@DESCRIPTION
THIS DEVICE IS PROGRAMMED TO FUNCTION AS 12-BIT UP COUNTER RUNNING AT 2.5MHZ (400 NS BIT RESOLUTION
/ TOT. CNT OF 1.64 MS). THE COUNTER IS DYNAMICALLY CONTROLLED TO START COUNTING WHEN THE (SIGIN)
INPUT IS ASSERTED HIGH.

IF THE (SIGIN) INPUT IS NOT NEGATED BY THE TIME THE COUNTER ELAPSES TO 105 US THE COUNTER ASSERTS
THE ALARM OUTPUT HIGH. AT THIS POINT THE COUNTER CONTINUES TO COUNT UNTIL AN ELAPSED TIME OF 922 US
+ 105 US (1.027 MS).

THIS COUNT VALUE RESETS THE COUNTER BACK TO ZERO WHERE IT THEN IDLES UNTIL THE NEXT (SIGIN) INPUT
REQUEST.

DURING THE 922 US CYCLE COUNT THE SEQUENCER SAMPLES THE (SIGIN INPUT) EVERY 53 US. IF THE INPUT IS
ASSERTED HIGH THE SEQUENCER WILL SET THE ALARM FLAG AND TIME OUT FOR THE SPECIFIED ALARM TIME OUT
CYCLE (922 US).
```

Figure 6-14.1 PLS168/BCA TRFDECT .BEE File (continued)

Programmable logic design and application notes

<————-12 BIT COUNTER————->

(TIME)	(DETECT POINTS)	(MODE)	(ST)	(F1)	(F0)	(P9)	(P8)	(P7)	(P6)	(P5)	(P4)	(P3)	(P2)	(P1)	(P0)
				2048	1024	512	256	128	64	32	16	8	4	2	1
				Q11	Q10	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
00	SIG * SIGINDLY * /AIN	START	P	L	L	L	L	L	L	L	L	L	L	L	L
10 US	/(SIGIN)* CNT = 25	TEST	P	L	L	L	L	L	L	L	H	H	L	L	H
105 US	(SIGIN) * CNT = 256	ALARM	P	L	L	L	L	H	H	H	H	H	H	H	H
155 US	(SIGIN) * CNT = 388	ALARM	P	L	L	L	H	H	L	L	L	L	H	L	L
208 US	(SIGIN) * CNT = 520	ALARM	P	L	L	H	L	L	L	L	L	H	L	L	L
261 US	(SIGIN) * CNT = 653	ALARM	P	L	L	H	L	H	L	L	L	H	H	L	H
314 US	(SIGIN) * CNT = 785	ALARM	P	L	L	H	H	L	L	L	H	L	L	L	H
367 US	(SIGIN) * CNT = 918	ALARM	P	L	L	H	H	H	L	L	H	L	H	H	L
420 US	(SIGIN) * CNT = 1050	ALARM	P	L	H	L	L	L	L	L	H	H	L	H	L
473 US	(SIGIN) * CNT = 1182	ALARM	P	L	H	L	L	H	L	L	H	H	H	H	L
526 US	(SIGIN) * CNT = 1315	ALARM	P	L	H	L	H	L	L	H	L	L	L	H	H
579 US	(SIGIN) * CNT = 1448	ALARM	P	L	H	L	H	H	L	H	L	H	L	L	L
632 US	(SIGIN) * CNT = 1580	ALARM	P	L	H	H	L	L	L	H	L	H	H	L	L
685 US	(SIGIN) * CNT = 1712	ALARM	P	L	H	H	L	H	L	H	H	L	L	L	L
738 US	(SIGIN) * CNT = 1845	ALARM	P	L	H	H	H	L	L	H	H	L	H	L	H
791 US	(SIGIN) * CNT = 1978	ALARM	P	L	H	H	H	H	L	H	H	H	L	H	L
844 US	(SIGIN) * CNT = 2110	ALARM	P	H	L	L	L	L	L	H	H	H	H	H	L
897 US	(SIGIN) * CNT = 2243	ALARM	P	H	L	L	L	H	H	L	L	L	L	H	L
955 US	(NORMAL)* CNT = 2387	NORMAL	P	H	L	L	H	H	L	H	L	L	L	H	H

——————END NORMAL SIGNAL MODE CHECK AND TIME OUT——————

Figure 6-14.2 PLS168/BCA TRFDECT .BEE File (continued)

Programmable logic design and application notes

```
----------------------------BEGIN TEST MODE CHECK-------------------------
-------------------------------------------------------------------------
1.578 MS    JUMP         CNT = 3945     TEST    N     H  H  H  H  L  H  H  L  H  L  L  H
-------------------------------------------------------------------------
1.582 MS    (SIGIN)   * CNT = 3957     ALARM   P     H  H  H  H  L  H  H  H  L  H  L  H
-------------------------------------------------------------------------
1.588 MS    (SIGIN)   * CNT = 3970     ALARM   P     H  H  H  H  H  L  L  L  L  L  H  L
-------------------------------------------------------------------------
1.592 MS    (SIGIN)   * CNT = 3982     ALARM   P     H  H  H  H  H  L  L  L  H  H  H  L
-------------------------------------------------------------------------
1.597 MS    (SIGIN)   * CNT = 3994     ALARM   P     H  H  H  H  H  L  L  H  H  L  H  L
-------------------------------------------------------------------------
1.602 MS    (SIGIN)   * CNT = 4006     ALARM   P     H  H  H  H  H  L  H  L  L  H  H  L
-------------------------------------------------------------------------
1.607 MS    (SIGIN)   * CNT = 4019     ALARM   P     H  H  H  H  H  L  H  H  L  L  H  H
-------------------------------------------------------------------------
1.612 MS    (SIGIN)   * CNT = 4032     ALARM   P     H  H  H  H  H  H  L  L  L  L  L  L
-------------------------------------------------------------------------
1.617 MS    (SIGIN)   * CNT = 4044     ALARM   P     H  H  H  H  H  H  L  L  H  H  L  L
-------------------------------------------------------------------------
1.622 MS    (SIGIN)   * CNT = 4057     ALARM   P     H  H  H  H  H  H  L  H  H  L  L  H
-------------------------------------------------------------------------
1.627 MS    (SIGIN)   * CNT = 4069     ALARM   P     H  H  H  H  H  H  H  L  L  H  L  H
-------------------------------------------------------------------------
1.632 MS    (SIGIN)   * CNT = 4082     ALARM   P     H  H  H  H  H  H  H  H  L  L  H  L
-------------------------------------------------------------------------
1.636 MS    (SIGIN)   * CNT = 4090     ALARM   P     H  H  H  H  H  H  H  H  H  L  H  L

----------------------------END TEST MODE CHECK---------------------------

IF THE (SIGIN) INPUT IS NEGATED BEFORE THE ELAPSED TIME OF 105 US THE COUNTER CONTINUES TO COUNT
UNTIL THE ELAPSED TIME OF 955 MS. AT THIS POINT THE COUNTER AGAIN RESETS ITSELF UNTIL THE NEXT
(SIGIN) INPUT REQUEST.

DURING TEST MODE, THE SEQUENCER SAMPLES THE SIGIN PULSE AT A REPETITION RATE OF 72 US. WITH A
PULSE WIDTH OF 6.5 US. THE TIME OUT FOR AN ALARM DETECT IS BASED ON SAMPLE A CHECK ONCE EVERY 5
US, AFTER THE 10 US SAMPLE FOR SIGIN NEGATION.
END
```

Figure 6-14.3 PLS168/BCA TRFDECT .BEE File (end)

Programmable logic design and application notes

```
@PINLIST
"<---------FUNCTION---------> <--REFERENCE-->"
"PINLABEL   PIN #   PIN_FCT    PIN_ID    OE_CTRL"
CLK           1        CK         CK        -   ;
SIGIN         2        I          I5        -   ;
N/C           3        I          I4        -   ;
SIGDLYIN      4        I          I3        -   ;
AIN           5        I          I2        -   ;
IA1           6        I          I1        -   ;
IA0           7        I          I0        -   ;
FA0           8        O          F0        /OE ;
FA1           9        O          F1        /OE ;
SIGDLYOUT    10        O          F2        /OE ;
ALARM        11        O          F3        /OE ;
GND          12        0V         GND       -   ;
N/C          13        O          P0        /OE ;
N/C          14        O          P1        /OE ;
N/C          15        O          P2        /OE ;
N/C          16        O          P3        /OE ;
/OE          17        PR         PR//OE    -   ;
N/C          18        I          I11       -   ;
N/C          19        I          I10       -   ;
N/C          20        I          I9        -   ;
N/C          21        I          I8        -   ;
N/C          22        I          I7        -   ;
N/C          23        I          I6        -   ;
VCC          24        +5V        VCC       -   ;
```

Figure 6-15. PLS168/BCA TRFDECT Pinlist

```
File Name : TRFDECT
Date : 9/14/1988
Time : 13:24:44

Cust/Project     -

Date             - 4/4/1988
Rev/I. D.        -


PLS168

__                                                    _____________________
T !                                                   !       OPTION P/E          !L!
E !---------------------------------------------------------------------------------
R ! !      INPUT VARIABLE     !   PRESENT STATE    !     NEXT STATE       !OUTPUT !
M !C!1 1 ---------------------------------------------------------------------------
__!_!1_0_9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!3_2_1_0!
 0!A!- - - -,- - H -,L L - - -!- -,- - - -,- - - -!L L,L L L L,L L L L!- H - - !
 1!.!- - - -,- - - -,H - - -!- -,- - - -,- - - L!- -,- - - -,- - - H!- - - -!
 2!.!- - - -,- - - -,H - - -!- -,- - - -,- - L H!- -,- - - -,- - H L!- - - -!
 3!.!- - - -,- - - -,H - - -!- -,- - - -,- L H H!- -,- - - -,- H L L!- - - -!
 4!.!- - - -,- - - -,H - - -!- -,- - - -,L H H H!- -,- - - -,H L L L!- - - -!
 5!.!- - - -,- - - -,H - - -!- -,- - - L,H H H H!- -,- - - H,L L L L!- - - -!
 6!.!- - - -,- - - -,H - - -!- -,- - L H,H H H H!- -,- - H L,L L L L!- - - -!
 7!.!- - - -,- - - -,H - - -!- -,- L H H,H H H H!- -,- H L L,L L L L!- - - -!
 8!.!- - - -,- - - -,H - - -!- -,L H H H,H H H H!- -,H L L L,L L L L!- - - -!
 9!.!- - - -,- - - -,H - - -!- L,H H H H,H H H H!- H,L L L L,L L L L!- - - -!
10!.!- - - -,- - - -,H - - -!L H,H H H H,H H H H!H L,L L L L,L L L L!- - - -!
11!.!- - - -,- - - -,H - - L!H H,H H H H,H H H H!L L,L L L L,L L L L!- - - H!
12!.!- - - -,- - - -,H - L H!H H,H H H H,H H H H!L L,L L L L,L L L L!- - H L!
13!.!- - - -,- - - -,H - H H!H H,H H H H,H H H H!L L,L L L L,L L L L!L L L L!
14!.!- - - -,- - - -,- - - -!- -,- - - -,- - - -!- -,- - - -,- - - -!- - - -!
```

Table 6-3.1 Transmitter Fault Detector Program Table (continued)

Programmable logic design and application notes

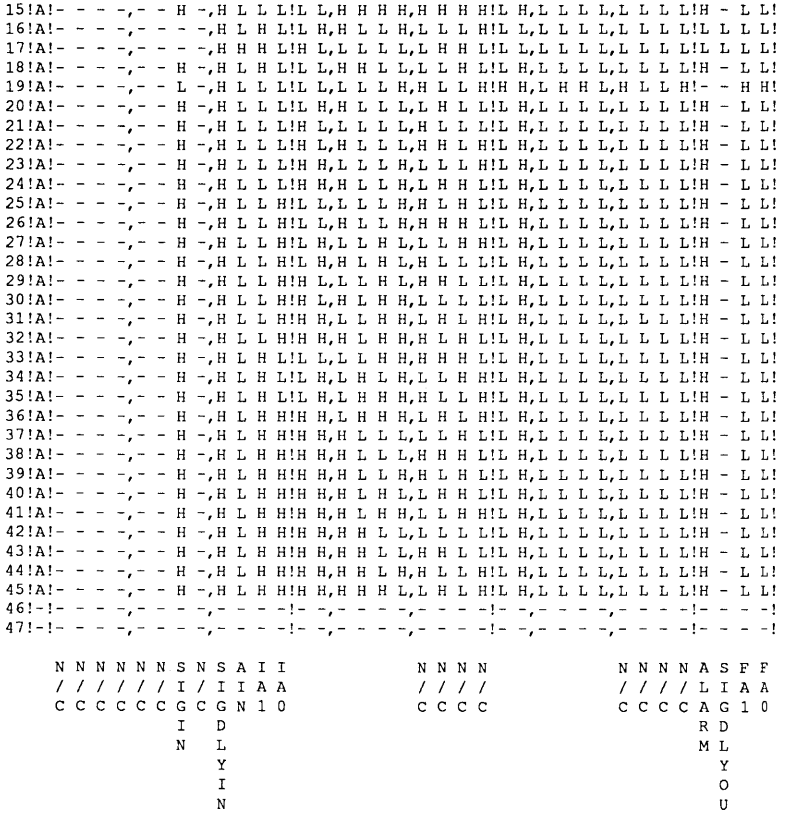


Table 6-3.2 Transmitter Fault Detector Program Table (end)

Programmable logic design and application notes

Scope Trace Sweep Circuit – PLS153 and PLS155

Jerry Liebler submitted this spectrum analyzer sweep circuit design.

```
@DEVICE TYPE
PLS153
@DRAWING
1
@REVISION
1
@DATE
1-27-88
@SYMBOL
@COMPANY
Tektronix Inc.
@NAME
sweep logic
@DESCRIPTION
This chip forms the sweep logic circuit for the 2710
spectrum analyzer.
@PINLIST
"<---------FUNCTION---------> <--REFERENCE-->"
"PINLABEL   PIN #   PIN_FCT    PIN_ID      OE_CTRL"
EOSWP-        1         I          I0          -   ;
TRIGIN        2         I          I1          -   ;
SLOPE         3         I          I2          -   ;
AUTOTRIG-     4         I          I3          -   ;
SSTRIG        5         I          I4          -   ;
SINGLSWP      6         I          I5          -   ;
MANSWP-       7         I          I6          -   ;
SGDIS         8         I          I7          -   ;
HOLDOFF       9         B          B0          D0  ;
GND          10         0V         GND         -   ;
RSFFI2       11         O          B1          D1  ;
RSFFI1       12         O          B2          D2  ;
SGDFFI       13         O          B3          D3  ;
SSFDDO       14         O          B4          D4  ;
SSDFFI       15         O          B5          D5  ;
STIN         16         I          B6          D6  ;
STOUT        17         O          B7          D7  ;
SWPGATE-     18         B          B8          D8  ;
SWPGATE0     19         O          B9          D9  ;
VCC          20         +5V        VCC         -   ;
```

Figure 6-16. PLS153 SWP Pinlist

Programmable logic design and application notes

```
@COMMON PRODUCT TERM
SWPTRIG=/(EOSWP- )+/((STOUT*/STIN)+/AUTOTRIG-)+/SSDFFO+/RSFFI2;
@I/O DIRECTION
D0=RSFFI2;
D8=/SWPGATE*/SGDIS;
@LOGIC EQUATION
RSFFI2=(EOSWP- )*/RSFFI1;
RSFFI1=/HOLDOFF*/RSFFI2;
SWPGATE = SGDFFI*/SWPTRIG + /MANSWP- + SWPGATE*(EOSWP- )*SWPTRIG;
SWPGATE- =0;
SGDFFI= SWPTRIG+/MANSWP-+(/SWPTRIG*(EOSWP- )*SGDFFI);
HOLDOFF = 0;
STOUT=((TRIGIN*/SLOPE)+(/TRIGIN*SLOPE));
SSDFFO= SSDFFI*SSTRIG+/SINGLSWP+(EOSWP- )*/SSTRIG*SSDFFO;
SSDFFI= /SSTRIG+/SINGLSWP+(EOSWP- )*SSTRIG*SSDFFI;
```

Figure 6-17. PLS153 SWP .BEE File

```
@DEVICE TYPE
PLS155
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST
"<---------FUNCTION---------> <--REFERENCE-->"
"PINLABEL    PIN #    PIN_FCT    PIN_ID    OE_CTRL"
HCLOCK         1          CK        CK        -    ;
VSYNC          2          I         I0        -    ;
ENDVSWP        3          I         I1        -    ;
VIDMON         4          I         I2        -    ;
TC-            5          I         I3        -    ;
VIDMON-        6          O         B0        D0   ;
PE-            7          O         B1        D1   ;
VMTST          8          I         B2        D2   ;
FMVID-         9          I         B3        D3   ;
GND           10          0V        GND       -    ;
N/C           11          /OE       /OE       -    ;
SWPGATE       12          B         B4        D4   ;
BLANK         13          O         B5        D5   ;
QBAR          14          /O        F0        EA   ;
ST0           15          /O        F1        EA   ;
ST1           16          /O        F2        EB   ;
TRIGGER       17          /O        F3        EB   ;
HSOUT         18          O         B6        D6   ;
HSIN          19          I         B7        D7   ;
VCC           20          +5V       VCC       -    ;
```

Figure 6-18. PLS155 SWP3 Pinlist

Programmable logic design and application notes

```
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@I/O DIRECTION
d4=blank;
@FLIP FLOP CONTROL
@OUTPUT ENABLE
@REGISTER LOAD
@ASYNCHRONOUS PRESET/RESET
@FLIP FLOP MODE
m0=1;
@LOGIC EQUATION
blank=(/(/qbar)+hsout)*vidmon+vmtst*/fmvid-;
swpgate=0;
hsout=hsin'
vidmon-=/vidmon;
/qbar : j=vsync*vidmon;
  k=endvswp+/vidmon;
/trigger: d=/(tc-);
pe-=/(st0*/st1*vsync);
```

Figure 6-19. PLS155 SWP3 .BEE File

```
@DEVICE SELECTION
swp3/pls155
@STATE VECTORS
    [st1,st0]
    odd   =00b;
    stb1  =01b;
    stb2  =10b;
    even  =11b;
@INPUT VECTORS
    [VSYNC]
    vi    =1b;
    notvi =0b;
@OUTPUT VECTORS
@TRANSITIONS
while [odd]
if [vi] then [stb1]
if [notvi] then [odd]

while [stb1]
if [vi] then [stb2]
if [notvi] then [odd]

while [stb2]
if [vi] then [stb1]
if [notvi] then [even]

while [even]
if [vi] then [stb1]
if [notvi] then [even]
```

Figure 6-20. PLS155 SWP3 .SEE File

Programmable logic design and application notes

Programmable Logic Devices

GENERAL APPLICATIONS

Motor Stepper Controller with the PLS155

Jim Greene designed and constructed the Stepper Motor Controller. This circuit allows control of bidirectional stepper motors for both single wave drive (only one phase on at a given time), and 2-phase drive (2 phases on at a time).

1. The clock (CLK) input can be driven by a continuous pulse train for steady rate movement or a clock that ramps up and down to provide for acceleration and deceleration. (Dependent on application.)
2. With the addition of control to another flip-flop, the PLS155 could provide half-step capability for finer resolution.
3. The preset and reset terms on the flip-flops could be used with a product term to provide an inhibit function if necessary.
4. The PLS155 will probably not have enough current drive for most stepper motor applications, therefore, a power buffer like the one shown (Figure 7-5) could be used. The components and values can be changed to fit your application.

Figure 7-2 shows the PLS155 pinlist, Figure 7-4 the logic diagram, Figure 7-3 corresponding design file, and Table 7-1 the final program table.

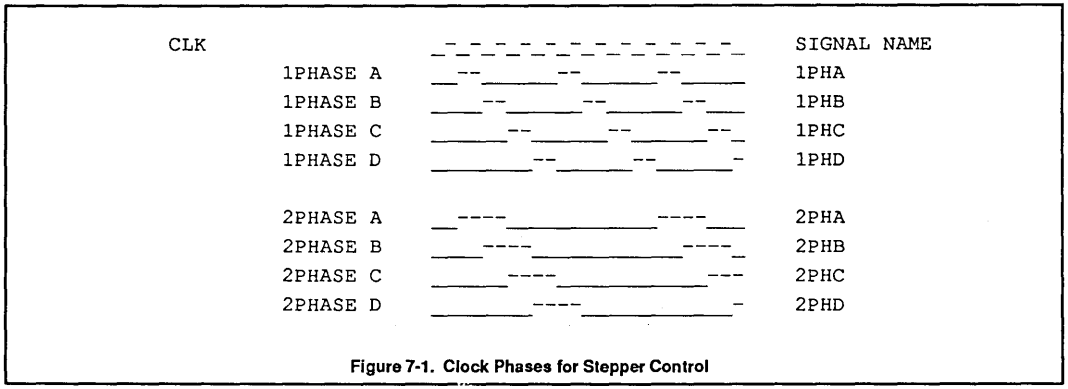


Figure 7-1. Clock Phases for Stepper Control

Programmable logic design and application notes

```
@DEVICE TYPE
PLS155
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST

"<---------FUNCTION---------> <--REFERENCE-->"
"PINLABEL    PIN # PIN_FCT     PIN_ID   OE_CTRL"
CLK           1       CK         CK       -     ;
DIR           2       I          I0       -     ;
N/C           3       I          I1       -     ;
N/C           4       I          I2       -     ;
N/C           5       I          I3       -     ;
2PHA          6       O          B0       D0    ;
2PHB          7       O          B1       D1    ;
2PHC          8       O          B2       D2    ;
2PHD          9       O          B3       D3    ;
GND          10       0V         GND      -     ;
N/C          11       /OE        /OE      -     ;
1PHA         12       O          B4       D4    ;
1PHB         13       O          B5       D5    ;
1Q           14       /O         F0       EA    ;
2Q           15       /O         F1       EA    ;
N/C          16       /B         F2       EB    ;
N/C          17       /B         F3       EB    ;
1PHC         18       O          B6       D6    ;
1PHD         19       O          B7       D7    ;
VCC          20       +5V        VCC      -     ;
```

Figure 7-2. PLS155 STEP Pinlist

Programmable logic design and application notes

```
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@I/O DIRECTION
d0-1; d1=1; d2=1; d3=1; d4=1; d5=1; d6=1;
d7=1;
@FLIP FLOP CONTROL
@OUTPUT ENABLE
 ea=0;
@REGISTER LOAD
 la =0;
@ASYNCHRONOUS PRESET/RESET
ra=/1;
@FLIP FLOP MODE
m1 = 1; m0 = 1;
@LOGIC EQUATION
1phd = 1q*2q;
1phc = 2q*/1q;
1phb = /2q*/1q;
1pha = /2q*/1q;
2phd = (1q*2q)*(2q+1q)+(/1q+/2q)*(/2q*/1q);
2phc = (2q*/1q)*(/1q+/2q)+(/2q+1q)*(1q*2q);
2phb = (/2q*1q)*(/2q+1q)+(2q+/1q)*(2q*/1q);
2pha = (/2q*/1q)*(2q+/1q)+(2q+1q)*(/2q*1q);
/2q :j = /dir*/1q+1q*dir;
     k = /dir*/1q+1q*dir;
/1q :j = 1;
     k = 1;
```

Figure 7-3. PLS155 STEP .BEE File

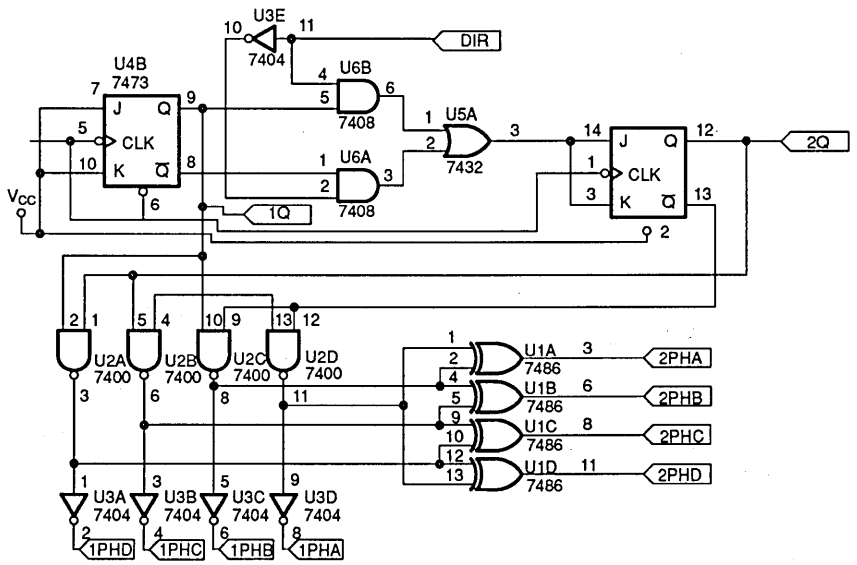


Figure 7-4. Stepper Motor Logic Diagram

Programmable logic design and application notes

```
File Name : STEP
Date : 9/14/1988
Rev/I. D.
PLS155                          !FF TYPE! EB EA !       !    POLARITY    !
                                !       !      !       !                !
---                             !-------!------!-----------------------!
 T !                            !A:A:.:.!O  :.  !.      !H:H:H:H:H:H:H:H!
 E !-------------------------------------------------------------------------
 R ! !   I   !        B(I)     ! Q(p)  ! Q(n)  ! P ! R !      B(O)      !
 M !C!-------------------------------------------!B!A!B!A!---------------!
---! !3 2 1 0!7 6 5 4 3 2 1 0!3 2 1 0!3 2 1 0!. .!. .!7 6 5 4 3 2 1 0!
  0!A!- - - -!- - - -,- - - -!- - L L!O O - -!. . . .!A . . .,A A . .!
  1!A!- - - -!- - - -,- - - -!- - L H!O O - -!. . . .!. A . .,. A A .!
  2!A!- - - -!- - - -,- - - -!- - H L!O O - -!. . . .!. . A .,. . A A!
  3!A!- - - -!- - - -,- - - -!- - H H!O O - -!. . . .!. . . A,A . . A!
  4!A!- - - -!- - - -,- - - -!- - - H!O O O -!. . . .!. . . .,. . . .!
  5!A!- - - -!- - - -,- - - -!- - - L!O O O -!. . . .!. . . .,. . . .!
  6!A!- - - -!- - - -,- - - -!- - - -!O O - O!. . . .!. . . .,. . . .!
  7!A!O O O O!O O O O,O O O O!O O O O!O O O O!A A A A!A A A A,A A A A!
*************************************************************************
*************************************************************************
 31!O!O O O O!O O,O O O O!O O,O O O O!O O O O!A A A A!A A A A,A A A A!
 Fc!O!O O O O!O O,O O O O!O O,O O O O O!                               .
 Lb!.!O O O O!O O,O O O O!O O,O O O O O!                              .
 La!.!O O O O!O O,O O O O!O O,O O O O O!                              .
 D7!-!- - - -!- -,- - - -!O O,O O O O O!                              .
 D6!.!- - - -!- -,- - - -!- -,- - - - -!                              .
 D5!-!- - - -!- -,- - - -!- -,- - - - -!                              .
 D4!.!- - - -!- -,- - - -!- -,- - - - -!                              .
 D3!-!- - - -!- -,- - - -!- -,- - - - -!                              .
 D2!-!- - - -!- -,- - - -!- -,- - - - -!                              .
 D1!-!- - - -!- -,- - - -!- -,- - - - -!                              .
 D0!.!- - - -!- -,- - - -!- -,- - - - -!                              .

      N N N D 1 1 1 1 2 2 2 2 N N 2 1 N N 2 1          1 1 1 1 2 2 2 2
      / / / I P P P P P P P P / / Q Q / / Q Q          P P P P P P P P
      C C C R H H H H H H H H C C     C C              H H H H H H H H
              D C B A D C B A                          D C B A D C B A
```

Table 7-1. STEPPER CONTROL Program Table

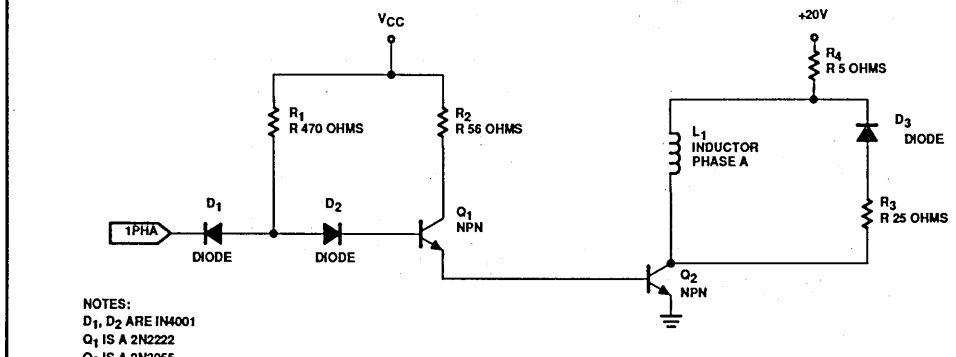


Figure 7-5. STEPPER CONTROL Power Buffer

Programmable logic design and application notes

Programmable Logic Devices

SECURITY SYSTEMS – Neil Kellet

Security systems are typified by some sensing circuit (perceiving intrusion, fire, etc.) and some basic activation circuit. Simple logic or complex sequences may be used with the sensors or the alarm generation circuits. Two of the following solutions utilize the innate capability of CMOS EPLDs to work well with RC timing circuits in generating time delays and relaxation oscillators.

A Programmable Alarm System – PLS168

This design illustrates an expansion of the design using additional PLS153 devices which were deleted in this version for brevity.

A basic alarm controller can be considered as a black box with several inputs and several outputs (Figure 8-1). Some inputs are used for detection and others for control. Detect inputs are driven from a variety of alarm transducers such as reed switches, smoke detectors, pressure mats, etc. An *ARM* input switches the system into a state which allows detection of the various alarm conditions and a *RESET* input is used to reset the system after an alarm has been triggered and dealt with or on re-entering the protected area. Outputs from the system include a sounder, a beacon and status indicators.

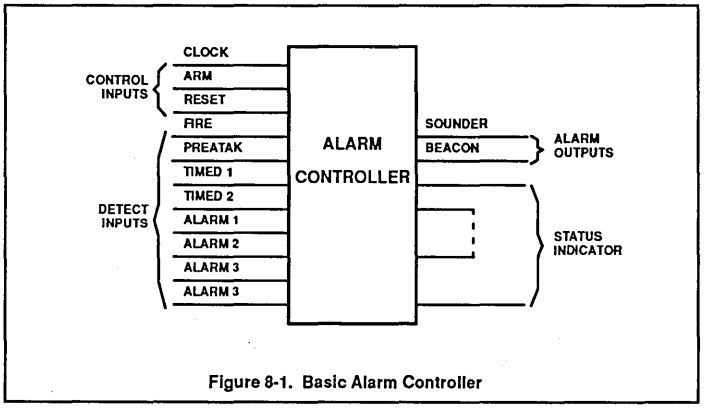


Figure 8-1. Basic Alarm Controller

Programmable logic design and application notes

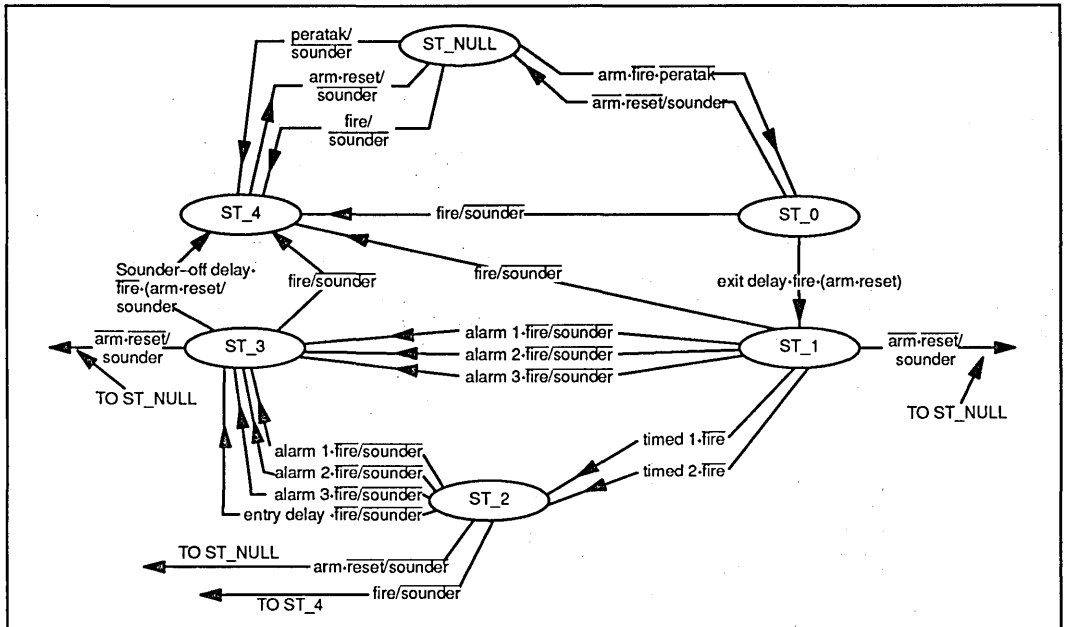


Figure 8-2. State Diagram for the Alarm Controller

Detect inputs can be divided into timed, untimed, fire and personal attack inputs. Timed circuits allow entry/exit delay circuits for front and rear doors, to delay operation of the alarm for approximately 16 seconds. Untimed circuits cause the alarm to operate immediately when an alarm condition occurs. These would be used to protect unusual means of entry, such as windows. Both the timed and untimed circuits should operate only if the system is armed.

The personal attack circuit is a special case untimed circuit and should operate only when the system is disarmed. The fire-detect circuit is again a special case untimed circuit and should operate regardless of whether the system is armed or not.

Outputs from the controller drive an external sounder and beacon. After 128 seconds, the sounder should turn off if the alarm has been triggered by either a timed or general untimed circuit. However, when a fire or personal attack triggers the system, the sounder should not turn off until the system is reset and the alarm condition removed.

State Machine Implementation

This design is best implemented as a state machine. The state diagram is derived from the verbal system description. Please note from Figure 8-2 the controller can be in one of six possible states. Examine the transitions from *ST_NULL* as an example. If a personal attack or fire condition occurs while in this state, a transition to *ST_1* takes place as indicated by the arrows on the diagram. Also at this time the sounder and beacon are activated, thus giving the alarm. If the fire and personal attack conditions have not occurred and the *ARM* SWITCH is set, then a transition to *ST_0* takes place.

Similarly, other arrows on the state diagram represent transitions between other states when specified input conditions occur. Output parameters are shown to the right of the slash line. Where there are no output parameters specified in a transition term, this indicates that no output changes are desired during this transition. That is, an output will hold its present value until told to change.

PLD Implementation

Having defined the desired system operation it is now time to select the required device to implement the desired system function from the PLD Data Manual. In this case, the device selected is the PLS168. Figure 8-3 shows the pinning information for the alarm controller. A 10-bit counter within the controller produces the entry/exit and sounder turn-off delays since this makes more efficient use of the PLD facilities than implementing the delays as part of the state machine. This counter uses seven internal registers with feedback and three without. For those registers without feedback, external wiring feeds their outputs back into the device to complete the 20-bit counter. Pins five to ten are used for this purpose. OutputT7 also forms part of the counter.

Three other registers form the state registers and are labeled SR0, SR1 and BEACON. State vectors for these registers have to be chosen with care to ensure that the beacon output is activated at the correct time. Other inputs and outputs are as already discussed. Note that the PR/OE pin is not used. This pin must be tied to ground in the final circuit.

Once the pin information has been entered, any Boolean equations desired can be defined using the Boolean equation entry (.BEE) file of AMAZE. List 1 shows the .BEE file for the alarm controller. Any internal registers used in either the Boolean equation or state equation entry file are given names in this file, in this case 1 to 16. Equations for the 10-bit counter are entered after the title line *@LOGIC EQUATION*, using registers t1 to t10in. Register SR0 halts and clears the counter while the controller is in certain states. This needs to be considered when defining the state vectors.

Programmable logic design and application notes

State Equation Entry

The state equation entry (.SEE) file of AMAZE uses a state-transition language, parameters of which are taken directly from the state diagram. Information is entered into this file in a free format. The only points to remember are that the square brackets should be used throughout to define the state registers and transitions, semicolons should be used to mark the end of vector definition and apostrophes should be used to indicate a registered output. State vectors can be defined in the state equation entry file as shown in List 2. State vectors are simply a means of labeling an arrangement of state registers which can be used later to define state transitions. Because we are using the *BEACON* output register as a state register also and SR0 is being used to halt and clear the 10-bit counter, particular care must be taken in defining the state vectors in this instance.

From the state diagram, the counter must begin counting during states *ST_0, ST_2* and *ST_3* and it must be cleared during states *ST_1, ST_4* and *ST_NULL*. State *ST_NULL* represents the power-up state of the PLS168 in which all register outputs are at logic one. Thus the inactive state of the counter is defined as being when SR0 is at logic one, therefore, SR0 must be at this level during states *ST_1* and *ST_4* and at logic zero during other states. The alarm beacon is considered to be active by an active-low signal and must be activated during states *ST_3* and *ST_4*. Register SR1 must therefore be chosen to ensure mutual exclusivity between state vectors. Input and output vectors can be defined in the same manner in terms of input and output label names. In this case, however, the label names are used directly. State transitions can now be derived directly from the state diagram. This is done using a Pascal-like state transition language and can clearly be seen in Table 8-1. Note that multiple *IF* statements can be implemented as such or as *CASE* statements as shown. Entry/exit and sounder turn-off delay times are represented as a decoding of the 10-bit counter states. Thus to get the desired 16 second entry/exit delay. t7 must be decoded and to achieve the 128 second sounder turn-off delay t10in must be decoded.

```
    STATE MACHINE AND TIMER FOR BURGLAR ALARM

@INTERNAL SR FLIP FLOP LABELS
t6  t5  t4  t3  t2  t1
@LOGIC EQUATION
t1: s = /t1*/sr0 ;
    r =  t1*/sr0
      +  sr0 ;
t2  s =  t1*/t2*/sr0 ;
    r =  t1* t2*/sr0
      +  sr0 ;
t3  s =  t1* t2*/t3*/sr0 ;
    r =  t1* t2* t3*/sr0 ;
      +  sr0 ;
t4  s =  t1* t2* t3*/t4*/sr0 ;
    r =  t1* t2* t3* t4*/sr0
      +  sr0 ;
t5  s =  t1* t2* t3* t4*/t5*/sr0 ;
    r =  t1* t2* t3* t4* t5*/sr0
      +  sr0 ;
t6  s =  t1* t2* t3* t4* t5*/t6*/sr0 ;
    r =  t1* t2* t3* t4* t5* t6*/sr0
      +  sr0 ;
t7  s =  t1* t2* t3* t4* t5* t6*/t7*/sr0 ;
    r =  t1* t2* t3* t4* t5* t6* t7*/sr0
      +  sr0 ;
t8  s =  t1* t2* t3* t4* t5* t6* t7*/t8*/sr0 ;
    r =  t1* t2* t3* t4* t5* t6* t7* t8*/sr0
      +  sr0 ;
t9  s =  t1* t2* t3* t4* t5* t6* t7* t8*/t9*/sr0 ;
    r =  t1* t2* t3* t4* t5* t6* t7* t8* t9*/sr0
      +  sr0 ;
t10 s =  t1* t2* t3* t4* t5* t6* t7* t8* t9*/t10*/sr0 ;
    r =  t1* t2* t3* t4* t5* t6* t7* t8* t9* t10*/sr0
      +  sr0 ;

    (Can be used later to define state equations)
```

Table 8-1. Logic Equations

Programmable logic design and application notes

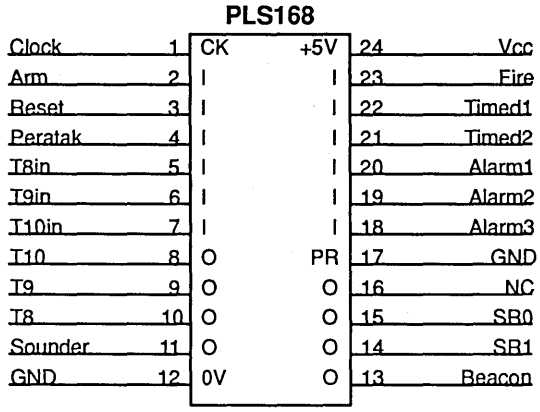


Figure 8-3. Pinning Information for the Alarm Controller

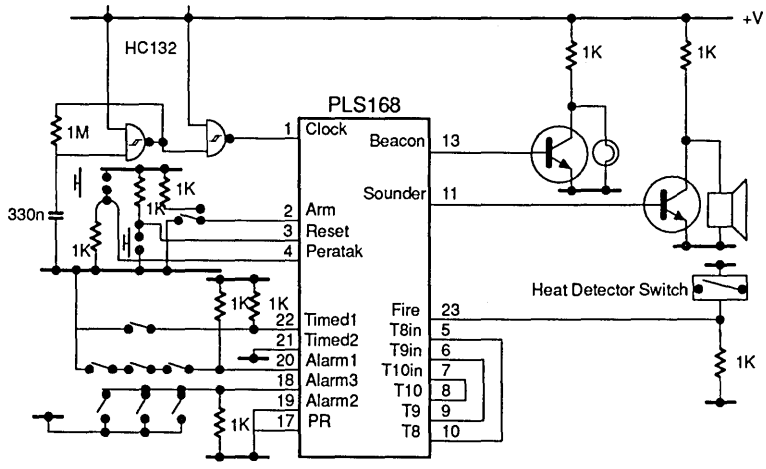


Figure 8-4. Alarm System based on the PLS168

With the system fully defined, simply assemble the design information during the AMAZE assembler to produce the fuse pattern for the desired device. Should any design changes meed to be made to a device, the fuse pattern may be modified directly using the program table editor of AMAZE. However, taking this action is not recommended since Boolean equation and state equation files are not altered correspondingly.

Functioning of the device can be verified with the AMAZE simulator, which can also be used to check A.C. timings before downloading the pattern to a device programmer, such as Stag ZL30A or Data I/O 29B, to program the device. Test vectors are produced either automatically or interactively by the simulator.

Programmable logic design and application notes

Programmability

The PLS168 device could now be used as the controller of an alarm system. As it stands, the device assumes that all the alarm inputs indicate an alarm condition when in the high state, logic one, and that the alarms are activated when the alarm outputs are active low (i.e., at logic zero).

Should an alarm input transducer be used which indicates an alarm condition as a low state, this can be catered for by altering the .SEE file. For example, consider a smoke detector which outputs logic zero on detection of an alarm condition and assume that this transducer is driving the "fire" input of the device. By changing all references to 'fire' in the .SEE file to '/fire' and all instances of '/fire' to 'fire' then the activation of the alarms will occur when logic zero is applied to this input and not when logic one is applied, as in the original case. Pinlist and .BEE files do not need to be altered.

Polarity of the output signals cannot be altered as easily, as the device will always power-up with the outputs at logic one. This should not prove to be a problem since the outputs simply drive output transistors and these can be used to produce the correct polarity signal for the beacon and sounder.

System Implementation

Figure 8-4 shows a typical alarm system based on this device. The system clock is produced by a relaxation oscillator built from 74HC132 Schmitt Triggers. Values of R_1 and C_1 shown result in a frequency of approximately 4Hz which will provide the desired entry/exit and sounder turn-off delays. These delays can be modified either by changing the external oscillator circuit or by decoding a different internal counter state. For example, to increase the entry/exit delay change all references to t7 in the .SEE file to t8. Both normally-closed and normally-open loop implementations are shown. Due to the distances involved in an alarm system, the open-loop configuration may cause problems, being driven by the positive supply. to avoid this problem, input-detect polarity of the open-loop circuit can be changed by altering the .SEE file.

Status indication can be provided by connecting LEDs as in Figure 8-5. When the reset button is pressed, any LED being lit will indicate an alarm condition for that input. This will not reset the alarm system unless the arm switch is off.

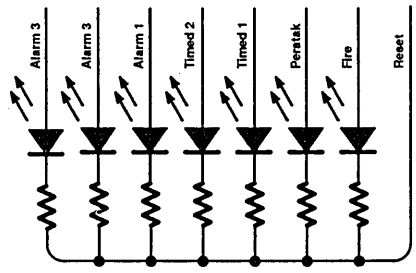


Figure 8-5. Status LEDs Connected to the alarm controller as shown provide status information

Programmable logic design and application notes

```
[sr0, sr1, beacon]

st_null = 111b;
st_0    = 001b;
st_1    = 101b;
st_2    = 011b;
st_3    = 010b;
st_4    = 100b;

@TRANSITIONS
While [st_null]
        case
          [arm* /fire * /peratak]           : : [st_0]
          [peratak]                         : : [st_4    with [/sounder']
          [fire]                            : : [st_4]   with [/sounder']
        encase

While [st_0]
        case
          [t7 * /fire * (arm + reset)]      : : [st_1]
          [/arm * /reset]                   : : [st_null] with [sounder']
          [fire]                            : : [st_4]   with [/sounder']
        encase

While [st_1]
        case
          [timed1 * /fire]                  : : [st_2]
          [timed2 * /fire]                  : : [st_2]
          [alarm1 * /fire]                  : : [st_3]   with [/sounder']
          [alarm2 * /fire]                  : : [st_3]   with [/sounder']
          [alarm3 * /fire]                  : : [st_3]   with [/sounder']
          [/arm * /reset]                   : : [st_null] with [sounder']
          [fire]                            : : [st_4]   with [/sounder']
        encase

While [st_2]
        case
          [t7 * /fire]                      : : [st_3]   with [/sounder']
          [alarm1 * /fire]                  : : [st_3]   with [/sounder']
          [alarm2 * /fire]                  : : [st_3]   with [/sounder']
          [alarm3 * /fire]                  : : [st_3]   with [/sounder']
          [/arm * /reset]                   : : [st_null] with [sounder']
          [fire]                            : : [st_4]   with [/sounder']
        encase

While [st_3]
        case
          [tl0in * /fire * (arm + reset)]   : : [st_4]   with [sounder']
          [[/arm * /reset]                  : : [st_null  with [sounder']
          [fire]                            : : [st_4]   with [/sounder']
        encase

While [st_4]
        case
         if    [/arm * /reset] then [st_null] with [sounder']
```

Table 8-2. State Equations can be used to define state transitions

CMOS sequencer solutions

Programmable Logic Devices

INTRODUCTION

Signetics invented the programmable logic sequencer with the 82S105 back in 1979. Since that time, additional parts were introduced, with a wide following of users who got programmable state machines into their designs. But, many potential users are still confused about what a sequencer is. Let us clarify that by simply stating that a sequencer is a programmable logic device capable of making user configurable state machines in a single chip. The first sequencers were configured with a programmable logic array (PLA) connected to a group of flip-flops. For state machine designs, the choice of either S-R or J-K flip-flops was appropriate because of logic efficiency. D flip-flops may be used, just as well, but are less efficient. Also, the restriction of using a PLA may not be mandatory. Many have used fixed-OR structures driving D flip-flops, and found the resulting solutions satisfactory. Hence, the broad definition of a sequencer is simply a programmable logic device with flip-flops. The inclusion of additional features beyond these basics can make all the difference in the world, and will be shown to be quite useful. Among the critical additional features are: A complement array, buried versus exposed flip-flops, independent flip-flop clocking and independent asynchronous set and reset capabilities. As well, something as simple as permitting some flip-flops to be clocked on rising clock edges and others on falling edges can have far-reaching performance implications.

This booklet will look at three parts that are sequencers, but which also have the additional property of being made from CMOS, so they have some low power capabilities that similar bipolar devices do not have. These devices are the PLC18V8Z, the PLC415 and the PLC42VA12. Each of these devices will be presented with a detailed application described that is appropriate to that device. After an initial description of the parts is given, a brief discussion of power-saving techniques is given. Then the example applications are detailed with complete design files which can be run on Signetics design software.

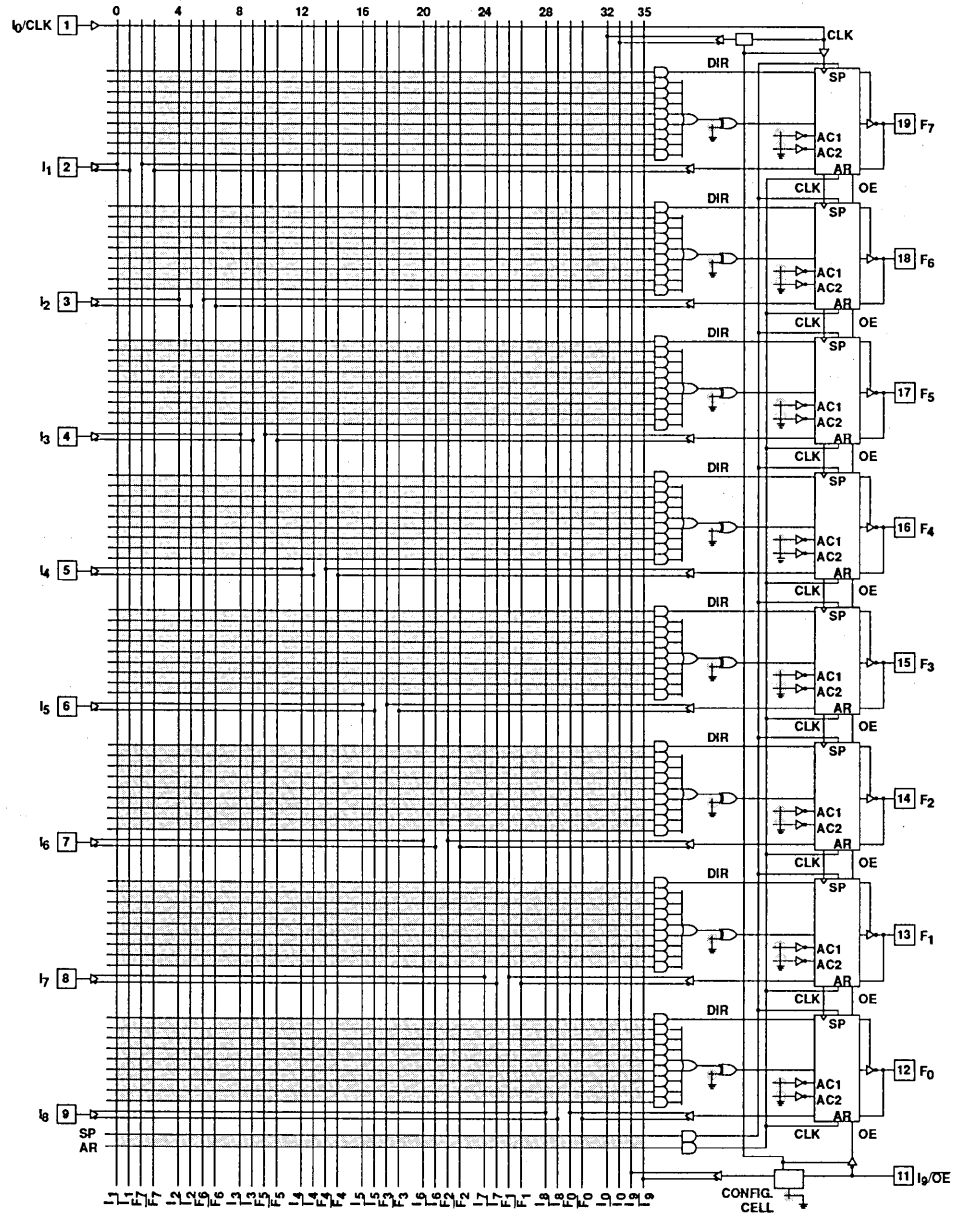
The PLC18V8Z

Figure 1 shows the PLC18V8Z logic diagram. As can be seen, there is a large programmable region that can interconnect input lines and feed back logic values to a region where they may be associated with the inputs to AND gates at small fixed-OR sites. This is termed Programmable Array Logic (PAL®). The outputs of the fixed-OR sites then drive into macrocells (the little boxes in Figure 1). The macrocell is detailed in Figure 2, where it is seen that the macrocell consists of one D flip-flop, three multiplexors, an exclusive-OR gate (for polarity control), configuration programmable sites and feedback paths. The output points of the macrocells have access to the chip's output pins. By configuring the macrocell multiplexors, it is possible for a signal coming into the macrocell to be routed (by MUX) to the output pins (from the combinational logic area) to take the flip-flop output to the pins, to feed back the flip-flop to the main logic array, or accept the "F" pin as in input. Because a large number of applications are "byte" oriented, eight macrocells fits in a data oriented system. The number of applications that require no more than 8 product terms per OR gate cluster, is also very large, and includes counters, shifters, pattern recognizers and handshakers. It should also be noted that a special output is dedicated on the macrocell to permit 3-State control of the output pins, from the programmable array.

Designers familiar with generic array logic (GAL™) will appreciate that the PLC18V8Z is intentionally pinned to be directly compatible with the 20-pin 16V8 device. This means that the PLC18V8Z can replace the long list of fixed-OR devices that includes the popular 16L8, 16R8, 16R6, 16R4, etc.

®PAL is a registered trademark of Monolithic Memories, Inc., a wholly-owned subsidiary of Advanced Micro Devices, Inc.
GAL is a trademark of Lattice Corp.

CMOS sequencer solutions



Denotes a programmable cell location.

Figure 1. PLC18V8Z

CMOS sequencer solutions

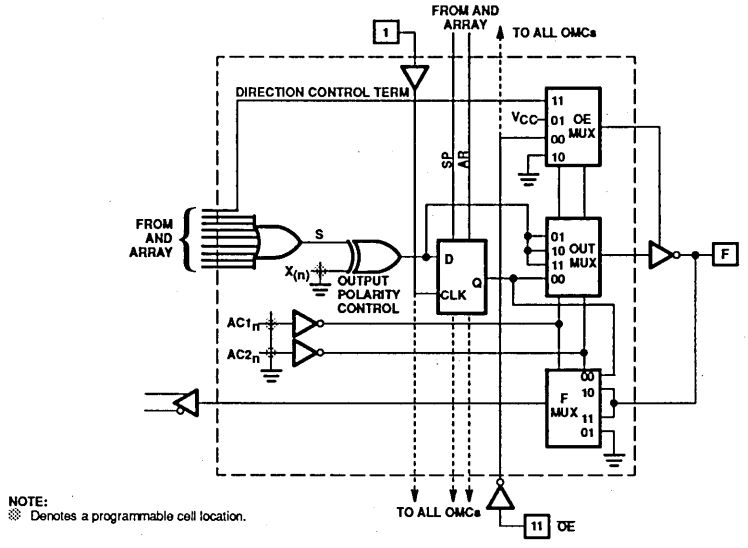


Figure 2. PLC18V8Z Macrocell

CMOS sequencer solutions

The PLC415

The PLC415 is shown in Figure 3, in a shorthand form. The actual part incorporates groups of J-K flip-flops with a programmable logic array. The flip-flops are logically grouped by virtue of association to specific output pins and association with particular clock inputs. The PLC415 illustrates both exposed flip-flops (Q outputs directly tied to the output pins) and buried flip-flops (Q outputs fed directly back to the programmable interconnect region). Sometimes, the outputs of the buried flip-flops are referred to as the state variable register because they capture next-state information to generate the transition signals for driving other flip-flop inputs. It should be noted that the PLC415 associates one group of four buried flip-flops, to a specific group of 4 exposed flip-flops by virtue of common clock inputs. Hence, this group can be thought of as a single, synchronous programmable state machine. The 64 product terms are available to be freely interconnected to any of the flip-flop OR gate inputs. There are no connection restrictions, and complete freedom of sharing. The second state machine can also use any of the AND gates as needed.

One of the powerful features of the PLC415, which is not available on registered fixed-OR devices, is multiple clock source availability. This, coupled with independent 3-State controls, permits some unique inherent output multiplexing capabilities.

Another feature that has been seldom understood, is the "complement term." Basically, the complement term is a NOR gate located in the PLC415 PLA, to permit efficient next-state transitioning. If flip-flop transitions are accomplished with Boolean products (asserted to logical "1" at their outputs), the state machine will transition accordingly. Usually, this is thought of by saying "If the machine is in state X, and an input of Z occurs, then assume the next state of W." What happens if input value Z doesn't occur? Usually, the machine will then stay in state X. But what if it is desired to move to another state if there are no asserted input conditions present? The logic designer is confronted with generating a function that provides the correct, positive asserted transition terms. This consumes lots of product terms, and the designer quickly depletes AND gates. However, by logically combining a product term that decodes the present state, with the missing input combination, a logical product will be generated permitting a next-state transition to occur when the input condition is absent. This product can be sent to the NOR term (i.e., complement term), which generates a logical "1" when all of its inputs are at a logical "0". This NOR output can force a transition from state X to state Y. The folks at Signetics refer to this as generating a logical "else" condition. Hence, using the complement term, a state transition may be described as "if state X and input Z, then state W, else state Y." A particularly slick feature of this attribute is that by combining the current state with the queried input into a product term, other product terms can be included in the complement NOR gate, which decode different states and input conditions. The state transition payoff is that only one complement term is really ever needed per state machine, to get the "else transitions" from all possible states. Because the PLC415 can build two independent machines, it includes two complement terms. The Signetics design software AMAZE and SNAP automatically use the complement term to perform "else state transitions." Because the complement term requires a signal to pass through the programming array twice before hitting a flip-flop input, there is a small speed penalty to use it. However, there is a major payoff in terms of AND gate usage. This feature has literally become a signature item in most Signetics sequencers.

CMOS sequencer solutions

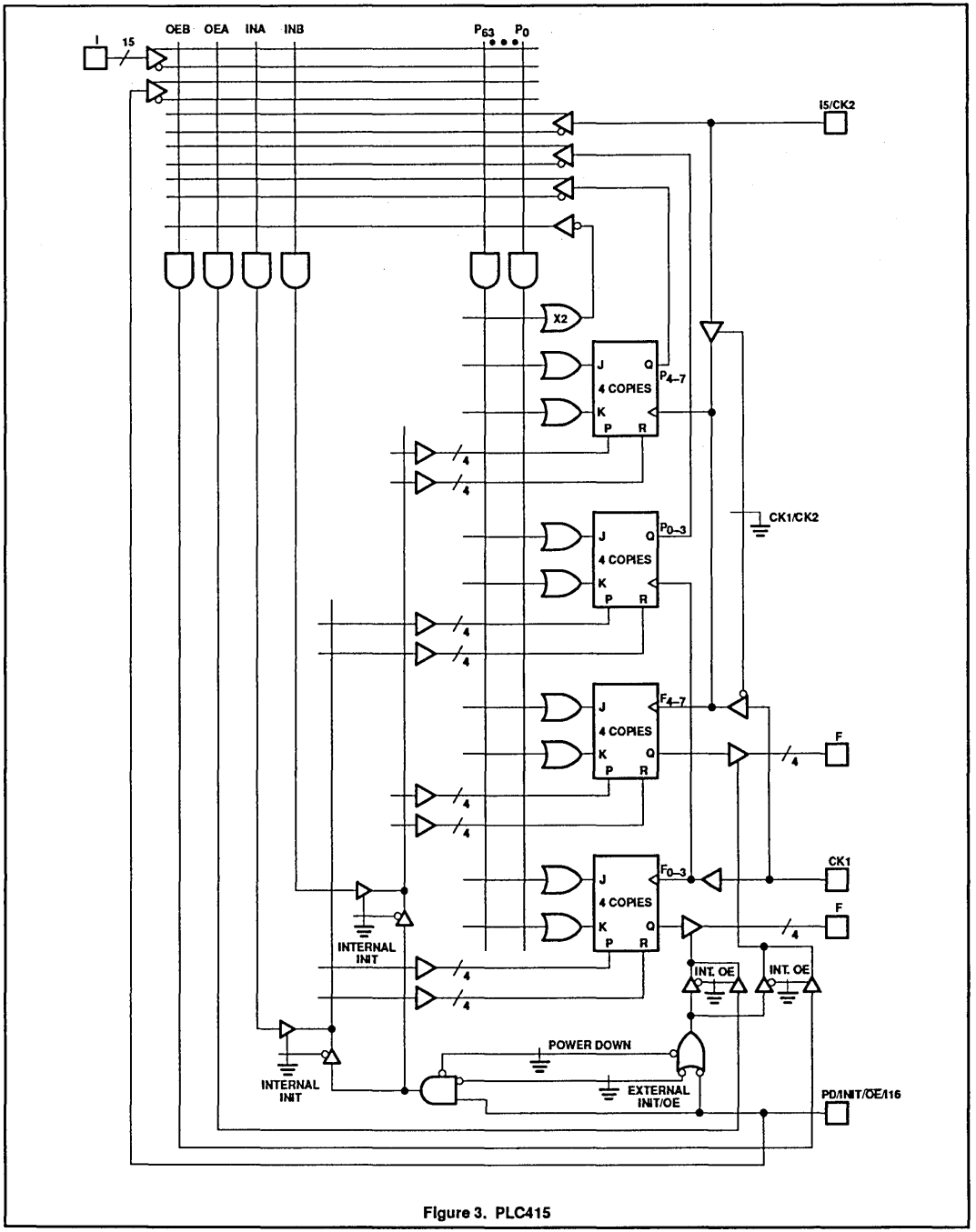


Figure 3. PLC415

CMOS sequencer solutions

The PLC42VA12

The PLC42VA12 was recently (December 1989) heralded as one of the top programmable logic devices of the year by *Electronic Design* magazine. Its architecture solves more standard design problems for a 24-pin device than any other comparable device. The PLC42VA12 is so flexible that designers are permitted nearly unrestricted design freedom. Incorporating 10 J-K flip-flops with a large PLA, the designer is able to clock each flip-flop from a distinct source. Each flip-flop has asynchronous set and clear, and flip-flop outputs either pass to an output pin, back into the programming array, neither, or both. Using an output pin does not force the associated flip-flop function to be lost, and outputs are 3-State controlled in small groups, from either dedicated pins, or from the PLA. The PLC42VA12 also has two combinational outputs. As can be seen in Figure 4, the combinational output points are driven from polarity controlling Exclusive-OR gates. It should be noted that the PLC42VA12 has been designed for compatibility with the popular 22V10 device. Designers that have enjoyed the 22V10, but needed greater flexibility, will appreciate additional freedom in designing with the PLC42VA12.

Functional independence is a key feature of the PLC42VA12. By having separate clocks for each flip-flop, the designer may treat each flip-flop as a separate element. Most PLDs assign a single clock to large groups of flip-flops, which forces the designer to restrict the applications to standard synchronous state machines. The PLC42VA12 permits a designer to build up to 10 (granted, simple) state machines in a single chip. But basically, design freedom is maximized in a PLC42VA12. Additionally, the asynchronous reset and set inputs are carefully partitioned among the flip-flops to minimize restrictive design practices. As usual, the complement term is available for efficient utilization of the "IF-THEN-ELSE" syntax.

CMOS sequencer solutions

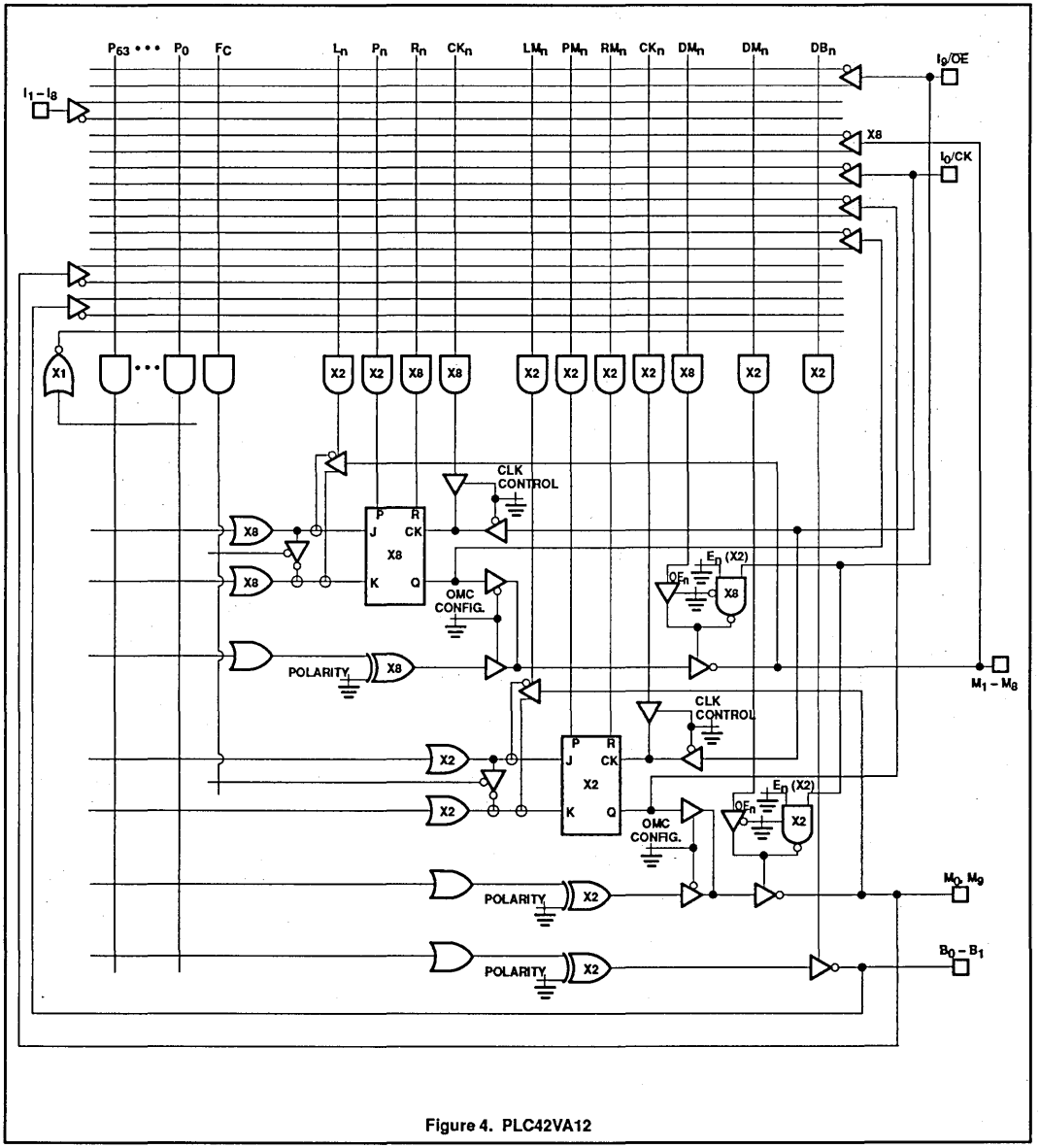


Figure 4. PLC42VA12

CMOS sequencer solutions

CMOS POWER IN PLDs

When one first thinks of CMOS, zero power is one concept that comes to mind. A CMOS device should consume not power in a DC state, and when switching the power should be proportional to the frequency. This idea seems to make the thought of a zero power CMOS PLD a natural one. Yet, when one examines the CMOS PLD marketplace, only a fraction of the devices claim the mantle of zero power. An understanding of the basic concepts can be vital.

With the exception of two architectures, all CMOS PLDs are constructed using an electrically erasable (EEPROM) or ultraviolet erasable (EPROM) cell similar to that shown in Figure 5. It should be pointed out that a true CMOS E^2 or EPROM memory cell does not exist. The core of almost all CMOS PLDs is an array of NMOS transistors. By wrapping the NMOS core with CMOS I/O cells, the illusion of a CMOS PLD is created (Figure 6). The only fallacy behind this is that NMOS devices consume power in a DC state. The array of NMOS devices are continually fed power to maintain optimum speed. This means that even in a DC state, where the device is not switching, the power level is in the 10's of mA. The main advantage here is that since only the array consumes power, the I_{CC} of the device will be much less than similar bipolar PLDs. If we could "eliminate" the power to the array, the power drops to levels expected from CMOS devices.

There are two classical electronic techniques used to eliminate the I_{CC} to the die core, and Signetics uses both of them depending on the part. Both techniques have inherent disadvantages, but lower I_{CC} dramatically. The first method is through the use of a special pin. When correctly asserted, a series blocking transistor(s) that supplies power to the core is turned off. This requires an external signal to control when I_{CC} is to be blocked and the core becomes "asleep" or inoperable.

To wake up the part, the power down condition must be released. Placing the part into and out of power down mode takes time and this impedes performance. Figure 7 illustrates the power saving approach of a dedicated control pin with the specific transaction for the PLC415. Note that PD (Power Down) must be asserted (i.e., logic "1") and released (logic "0") with a specific timing relationship to the clock. If the timing is maintained, the device will power down and power back up in the same state. Should the timing be altered, the internal state may be lost.

From a simple viewpoint, the other method, which is used in Signetics 18V8Z, is more convenient. This technique, called "Input Transition Detection" (ITD), has been adapted from MOS memory design. A diagram of the key circuitry is shown in Figure 8. If any input makes either a high-to-low or low-to-high transition, the ITD senses it and sends a pulse to the array supply module. The supply in turn provides power to the array for a period of time long enough that the contents of the array can be latched. This data is now available as inputs into the I/O buffers for further processing. Once the data has been latched, there is no need for the array to consume power. The width of the power pulse is designed so it only need fulfill this function. This period of time, approximately 20ns for the 18V8Z, is much shorter than the cycle time of the device (35ns for the 18V8Z). This means that the array only consumes power for 57% of each cycle. Obviously when longer cycles are used, the percentage is reduced. The rest of the circuitry obeys CMOS rules. The IDT feature gives a device more typical of a traditional CMOS technology. When all inputs are at static CMOS levels, power to the array is turned off so the device consumes a current of less than 100μA. Unlike some zero power PLDs, there is no surge in current once the device becomes active. When switching, the current of the 18V8Z rises at a linear rate which is typically 1mA per MHz.

The key to the technique is the Input Transition Detection circuit. Shown in Figure 9 is a simplified ITD implementation. When any of the inputs makes a transition, the output of the Ex-OR gate, as well as the OR gate, goes high for a period determined by the delay circuit. This "power signal" is what feeds the array supply circuit which eventually powers the array. The width of the delay varies across temperature to compensate for faster speeds at cold temperatures and slower speeds at hot temperatures.

Waveforms for the ITD circuit are shown in Figure 10. When input A makes a low-to-high transition, the power signal goes high for a period of t_{AP} after a delay of t_{DP}. From these waveforms, it is easy to see one of the disadvantages of this method. When several inputs are applied to the device in a system environment, there will be a finite amount of skew between the inputs. Since all inputs are tied to the ITD circuitry, pulses will be generated for each input. An example of this is shown in Figure 11. The power signal is initially triggered by input A. Input B, which is skewed from input A by the time t_{SKEW} generates its own power pulse, which is ORed to the pulse generated by input A. This makes the power pulse last longer in a system environment, which means the device will consume more power than originally anticipated.

A disadvantage of the ITD circuit is a speed penalty. Two factors in the design reduce the speed. Since the array is not continuously powered, some delay is incurred by the ITD circuit to provide array power. The other speed penalty is paid in the data latches. These latches are necessary to store the array contents.

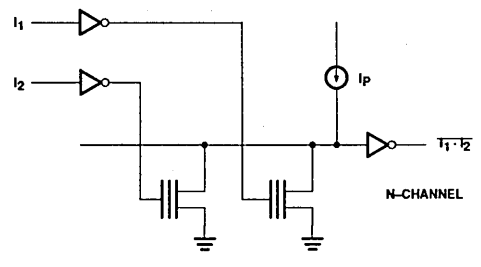


Figure 5. CMOS Programming – The EPLD

CMOS sequencer solutions

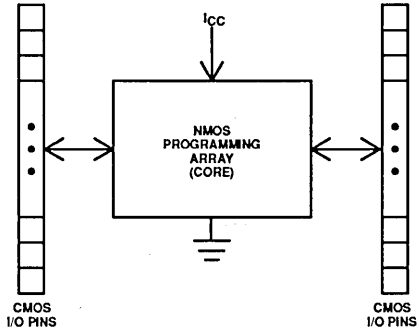


Figure 6. CMOS PLD Layout

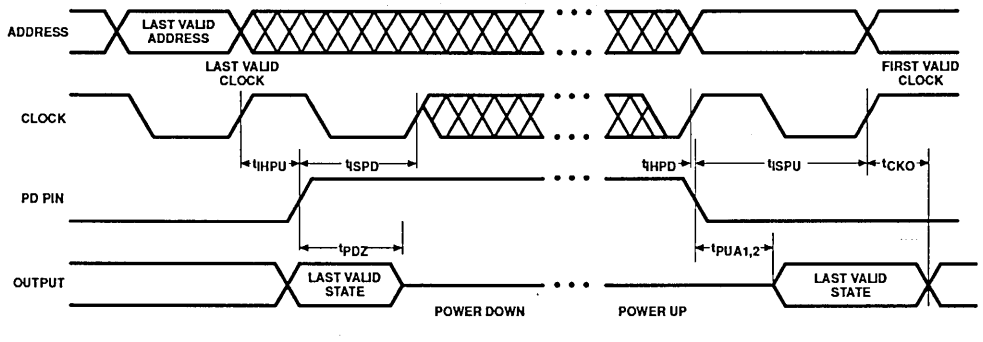


Figure 7. Power Down Enable and Disable

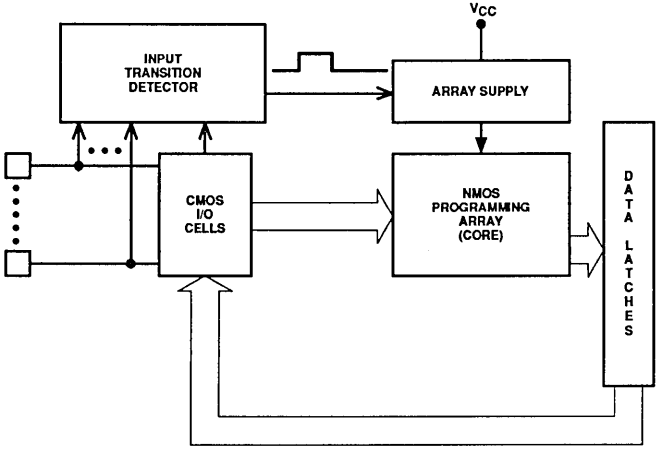


Figure 8. CMOS PLD with ITD

CMOS sequencer solutions

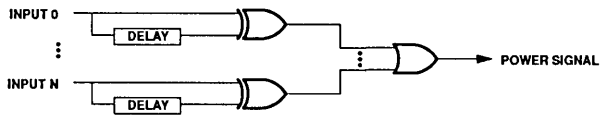


Figure 9. ITD Circuit

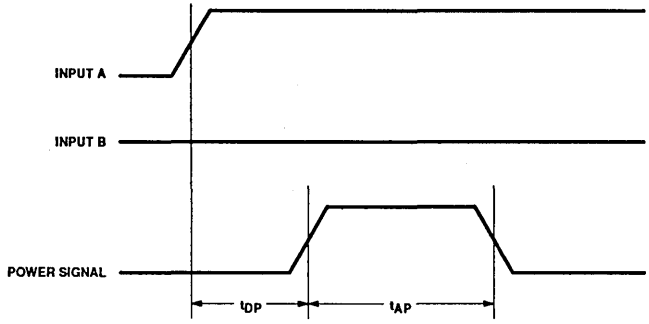


Figure 10. ITD Waveforms

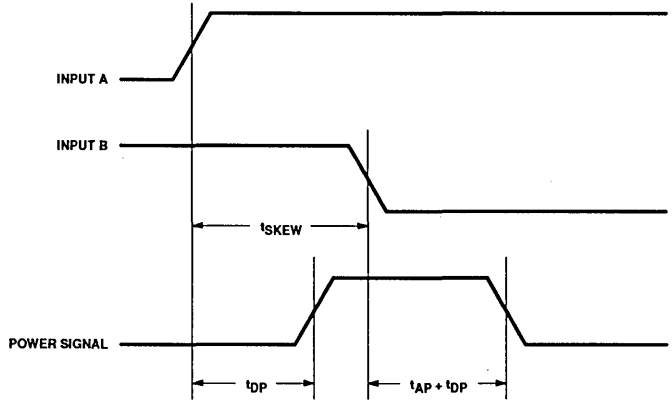


Figure 11. ITD with Skewed Inputs

CMOS sequencer solutions

PLC18V8Z APPLICATIONS

The 80C51 microcontroller and its CMOS derivatives have two power reducing modes, Idle and Power Down. The Power Down mode reduces the device's current to less than 50μA by only keeping the on-chip RAM and SFRs data intact. In order to resume operation while in the Power Down mode, it is necessary to apply a reset to the microcontroller.

The PLC18V8Z is in a low power mode whenever its inputs are not switching, drawing less than 100μA. An input transition causes the PLC18V8Z to power up its internal array for a short time, latch a valid output and then return to low power mode. Because of this transparent power reduction feature and its programmability, the PLC18V8Z is an excellent device to use in low power applications with an 80C51 microcontroller.

Two examples of using the PLC18V8Z with a SC87C751 microcontroller are presented. Both applications use the PLC18V8Z to detect events while the SC87C751 is in a Power Down mode and then reset (wake up) the microcontroller. The first example, shown in Figure 12, uses the PLC18V8Z as an 8-bit priority encoder. An AMAZE listing of the circuitry fused inside the PLC18V8Z is shown in Figure 13.

Whenever one of the inputs $I_7 - I_0$ goes LOW, a binary representation of its position is output on pins $A_3 - A_0$. If more than one input is Active-LOW, then the input with the highest priority is represented on the output, where I_7 has the highest priority. Another output, $\overline{EO}$, is not connected to the microcontroller but is used to control the RST output of the PLC18V8Z. $\overline{EO}$ is Active-LOW anytime all inputs are high. Actually, the PLC18V8Z could easily be reprogrammed to output the inverse of this signal which could be tied to the interrupt line of the microcontroller to generate an interrupt anytime one or more inputs were low.

Pin 16 of the PLC18V8Z, labeled RST, is the output of a 3-State buffer whose input is always high. The buffer's control line is tied internally to a product term which is enabled by $\overline{EO}$ and an input from the microcontroller labeled $\overline{RSTEN}$. The RST buffer may be in only two states, either driving a high (resetting the SC87C751) or 3-State (allowing C_3 to discharge), enabling normal operation of the microcontroller. Before entering the Power Down mode, the microcontroller should force $\overline{RSTEN}$ low. Then, any low on $I_7 - I_0$ will cause $\overline{EO}$ and also RST high, resetting the microcontroller. When the microcontroller is reset, it will force its ports to input mode and since P_1 and P_3 have internal pull-up resistors, $\overline{RSTEN}$ will go high forcing RST into the 3-State mode allowing C_3 to discharge.

The second example, shown in Figure 14, with AMAZE listing in Figure 15, uses the PLC18V8Z to monitor three microcontroller input lines ($I_{NC} - I_{NA}$) and reset the microcontroller upon any change. Three internal registers inside the PLC18V8Z are used to hold the states or levels of the input lines prior to entering Power Down mode. Before entering Power Down mode, the microcontroller should clock into the PLC18V8Z the states of $I_{NC} - I_{NA}$ with the LOAD signal. Comparator logic fused into the PLC18V8Z compares the output of the registers to the three input lines. The RST output of the PLC18V8Z operates in a similar manner to the first example to reset the microcontroller whenever $\overline{RSTEN}$ is low and the output of the comparator is false.

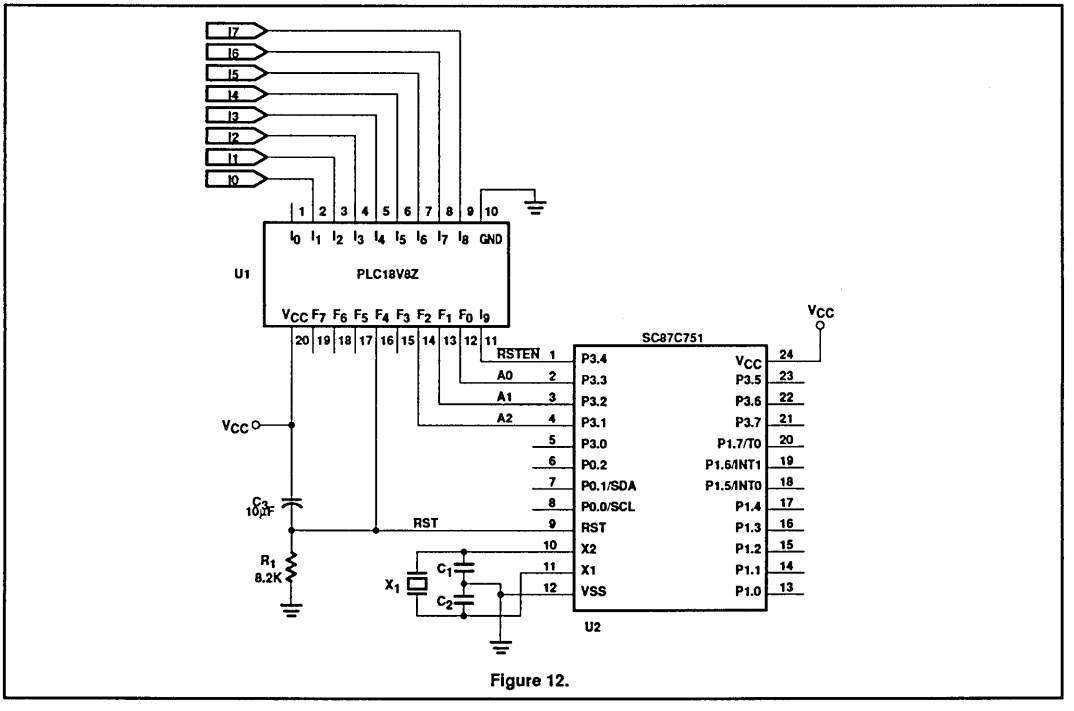


Figure 12.

CMOS sequencer solutions

```
File Name : ENCODER
Date : 3/29/1990
Time : 11:23:58

@DEVICE TYPE
PLC18V8Z
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST

"<--------FUNCTION-------->    <--REFERENCE-->"
"PINLABEL    PIN #  PIN_FCT    PIN_ID     OE_CTRL"
N/C           1       I          IO/CLK      -  ;
I0            2       I          I1          -  ;
I1            3       I          I2          -  ;
I2            4       I          I3          -  ;
I3            5       I          I4          -  ;
I4            6       I          I5          -  ;
I5            7       I          I6          -  ;
I6            8       I          I7          -  ;
I7            9       I          I8          -  ;
GND          10       0V         GND         -  ;
/RSTEN       11       O          I9//OE      -  ;
A0           12       O          B0          D0          ;
A1           13       O          B1          D1          ;
A2           14       O          B2          D2          ;
/EO          15       /B         B3          D3          ;
RST          16       B          B4          D4          ;
N/C          17       /B         B5          D5          ;
N/C          18       /B         B6          D6          ;
N/C          19       /B         B7          D7          ;
VCC          20       +5V        VCC         -  ;

@COMMON PRODUCT TERM  "CPT_label = (expression)"
@I/O DIRECTION  " D0 .. Dn "
" Enable: /EO "
 D3 = 1;
" Enable: RST "
 D4 = RSTEN * /EO;

@ASYNCHRONOUS PRESET/RESET " AR = <EXP>; "
@SYNCHRONOUS PRESET/RESET " SP = <EXP>;"
@LOGIC EQUATION
 A0   =  /I1*I2*I3*I4*I5*I6*I7
       +      /I3*I4*I5*I6*I7
       +           /I5*I6*I7
       +                /I7;
 A1   =     /I2*I3*I4*I5*I6*I7
       +      /I3*I4*I5*I6*I7
       +              /I6*I7
       +                 /I7;
 A2   =          /I4*I5*I6*I7
       +           /I5*I6*I7
       +              /I6*I7
       +                 /I7;
 /EO  =  /(I0*I1*I2*I3*I4*I5*I6*I7);
 RST  =  1;
```

Figure 13.

CMOS sequencer solutions

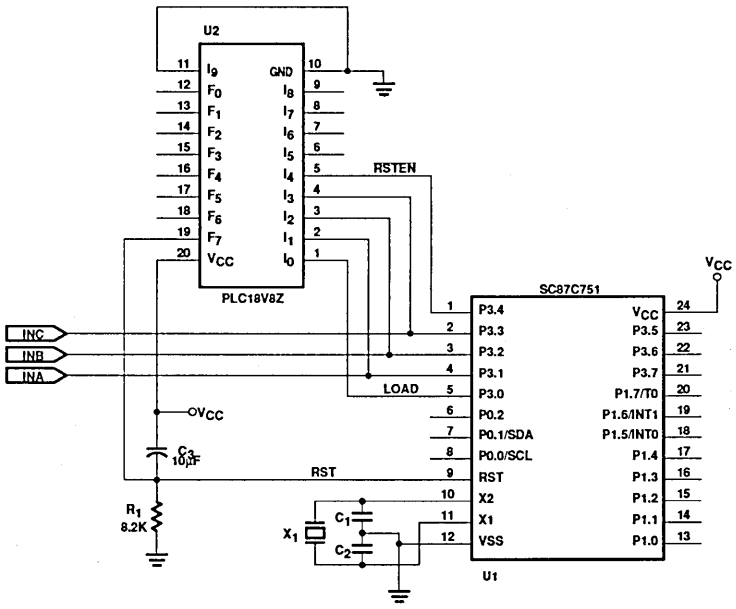


Figure 14.

CMOS sequencer solutions

```
File Name : WAKEUP
Date : 3/29/1990
Time : 12:22:57

@DEVICE TYPE
PLC18V8Z
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST

"<--------FUNCTION-------->    <--REFERENCE-->"
"PINLABEL    PIN #  PIN_FCT    PIN_ID      OE_CTRL"
LOAD          1       CLK        IO/CLK      -  ;
INA           2       I          I1          -  ;
INB           3       I          I2          -  ;
INC           4       I          I3          -  ;
/RSTEN        5       I          I4          -  ;
N/C           6       I          I5          -  ;
N/C           7       I          I6          -  ;
N/C           8       I          I7          -  ;
N/C           9       I          I8          -  ;
GND          10       0V         GND         -  ;
OE           11       /OE        I9//OE      -  ;
OUTA         12       B          B0          D0          ;
OUTB         13       B          B1          D1          ;
OUTC         14       B          B2          D2          ;
COMP         15       B          B3          D3          ;
DA           16       D          B4          D4          ;
DB           17       D          B5          D5          ;
DC           18       D          B6          D6          ;
RST          19       B          B7          D7          ;
VCC          20       +5V        VCC         -  ;

@COMMON PRODUCT TERM  "CPT_label = (expression)"
@I/O DIRECTION  " D0 .. Dn "
 D0 = 1;
 D1 = 1;
 D2 = 1;
 D3 = 1;
"Enable: RST "
 D7 = rsten * /comp;

@ASYNCHRONOUS PRESET/RESET " AR = <EXP>; "
@SYNCHRONOUS PRESET/RESET " SP = <EXP>;"
@LOGIC EQUATION
 RST  =  1;
 da :d = ina;
 db :d = inb;
 dc :d = inc;
 outa = da*ina + /da*/ina;
 outb = db*inb + /db*/inb;
 outc = dc*inc + /dc*/inc;
 comp = outa * outb * outc;
```

Figure 15.

CMOS sequencer solutions

PLC415 APPLICATIONS

This example places 2 independent stepper motor controllers in one PLC415. Each individually clocked controller includes a direction input as well as full and half step control. Individual set inputs force the internal state and output registers to state #1. Pin 19 is fused as a power-down input and may be used to reduce current consumption while the motors are stationary.

Suppose a stepper motor needs the sequence of data shown in Table 1. Clockwise rotation is performed by applying outputs associated with steps 1 through 8, while counter-clockwise rotation is achieved by applying outputs corresponding to 8 through 1. Each state or step in this table is actually one half step to the motor. A full step skips one state.

AMAZE listings for this example are shown in Figure 16 and Figure 17. The Boolean Equation Entry (BEE) file contains the pinlist, internal flip-flop labels, and flip-flop initialization data. The State Equation Entry (SEE) contains state equations describing the functions of each controller. In AMAZE 1.9, after compilation it is necessary to invoke the Program Table Editor (PTE) and change the polarity of the dual-clock fuse. The fuse shown under the CLK 1/2 header should be changed from 'L' to 'H'.

Table 1. Half Step Sequence

STATE	W1D	W1C	W1B	W1A
STEP 1	0	1	0	1
STEP 2	0	0	0	1
STEP 3	1	0	0	1
STEP 4	1	0	0	0
STEP 5	1	0	1	0
STEP 6	0	0	1	0
STEP 7	0	1	1	0
STEP 8	0	1	0	0

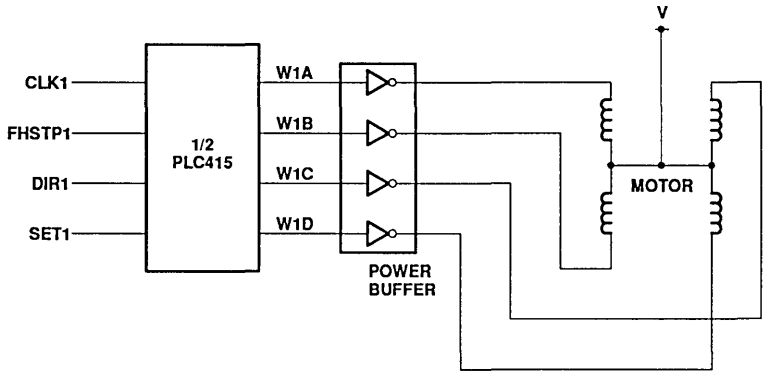
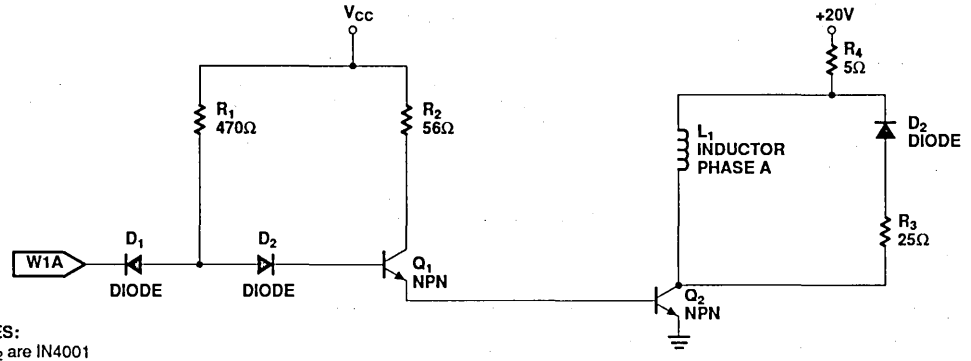


Figure 16. Stepper Control Circuit

CMOS sequencer solutions



NOTES:
D_1, D_2 are IN4001
Q_1 is a 2N2222
Q_2 is a 2N3055
D_3 is a BY127

Figure 17. Stepper Control Power Buffer
1 OF 4

CMOS sequencer solutions

```
@DEVICE TYPE
PLC415
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST

"<--------FUNCTION-------->   <--REFERENCE-->"
"PINLABEL    PIN #  PIN_FCT   PIN_ID     OE_CTRL"
CLK1          1       CK        CLK        -  ;
N/C           2       I         I7         -  ;
N/C           3       I         I6         -  ;
CLK2          4       CK2       I5/CK2     -  ;
FHSTP1        5       I         I4         -  ;
DIR1          6       I         I3         -  ;
SET1          7       I         I2         -  ;
N/C           8       I         I1         -  ;
N/C           9       I         I0         -  ;
W1D          10       O         F7         /OE        ;
W1C          11       O         F6         /OE /OEA   ;
W1B          12       O         F5         /OE /OEA   ;
W1A          13       O         F4         /OE /OEA   ;
GND          14       0V        GND        -  ;
W2A          15       O         F3         /OE        ;
W2B          16       O         F2         /OE        ;
W2C          17       O         F1         /OE        ;
W2D          18       O         F0         /OE        ;
PDOWN        19       PD        INIT/OE    -  ;
N/C          20       I         I15        -  ;
N/C          21       I         I14        -  ;
SET2         22       I         I13        -  ;
DIR2         23       I         I12        -  ;
FHSTP2       24       I         I11        -  ;
N/C          25       I         I10        -  ;
N/C          26       I         I9         -  ;
N/C          27       I         I8         -  ;
VCC          28       +5V       VCC        -  ;


@INTERNAL FLIP FLOP LABELS  "Label declared for FF labels."
p2a,p2b,p2c,p2d,p1a,p1b,p1c,p1d;

@COMMON PRODUCT TERM  "CPT_label = (expression)"
@COMPLEMENT ARRAY  "/C0 = /(---);/C1 = /(---);"
@INIT/OE  " INT_Pin_Label = R0 .. Rn;   or   R0 .. Rn; "
/r0, r1,/r2, r3,/r4, r5,/r6, r7,/r8, r9,/r10, r11,/r12, r13,/r14, r15;

@INIT/OE CONTROL TERM
 INA = set1;
 INB = set2;
@LOGIC EQUATION
```

Figure 18. Boolean Equation Entry File

CMOS sequencer solutions

```
@DEVICE SELECTION
415st/plc415
@STATE VECTORS
[p1d,p1c,p1b,p1a]
stp1 = 0101b;
stp2 = 0001b;
stp3 = 1001b;
stp4 = 1000b;
stp5 = 1010b;
stp6 = 0010b;
stp7 = 0110b;
stp8 = 0100b;

[p2d,p2c,p2b,p2a]
stp1a= 0101b;
stp2a= 0001b;
stp3a= 1001b;
stp4a= 1000b;
stp5a= 1010b;
stp6a= 0010b;
stp7a= 0110b;
stp8a= 0100b;

@INPUT VECTORS
@OUTPUT VECTORS
[w1d,w1c,w1b,w1a]
step1 = 0101b;
step2 = 0001b;
step3 = 1001b;
step4 = 1000b;
step5 = 1010b;
step6 = 0010b;
step7 = 0110b;
step8 = 0100b;

[w2d,w2c,w2b,w2a]
step1a= 0101b;
step2a= 0001b;
step3a= 1001b;
step4a= 1000b;
step5a= 1010b;
step6a= 0010b;
step7a= 0110b;
step8a= 0100b;
```

Figure 19.a State Equation Entry File

```
@TRANSITIONS

                        "motor comtroller #1"
while [stp1]
if [ fhstp1* dir1] then [stp3] with [step3']   "full step
forward"
if [/fhstp1* dir1] then [stp2] with [step2']   "half step
forward"
if [ fhstp1*/dir1] then [stp7] with [step7']   "full step
backward"
if [/fhstp1*/dir1] then [stp8] with [step8']   "half step
backward"

while [stp2]
if [ fhstp1* dir1] then [stp4] with [step4']
if [/fhstp1* dir1] then [stp3] with [step3']
if [ fhstp1*/dir1] then [stp8] with [step8']
if [/fhstp1*/dir1] then [stp1] with [step1']

while [stp3]
if [ fhstp1* dir1] then [stp5] with [step5']
if [/fhstp1* dir1] then [stp4] with [step4']
if [ fhstp1*/dir1] then [stp1] with [step1']
if [/fhstp1*/dir1] then [stp2] with [step2']

while [stp4]
if [ fhstp1* dir1] then [stp6] with [step6']
if [/fhstp1* dir1] then [stp5] with [step5']
if [ fhstp1*/dir1] then [stp2] with [step2']
if [/fhstp1*/dir1] then [stp3] with [step3']

while [stp5]
if [ fhstp1* dir1] then [stp7] with [step7']
if [/fhstp1* dir1] then [stp6] with [step6']
if [ fhstp1*/dir1] then [stp3] with [step3']
if [/fhstp1*/dir1] then [stp4] with [step4']

while [stp6]
if [ fhstp1* dir1] then [stp8] with [step8']
if [/fhstp1* dir1] then [stp7] with [step7']
if [ fhstp1*/dir1] then [stp4] with [step4']
if [/fhstp1*/dir1] then [stp5] with [step5']

while [stp7]
if [ fhstp1* dir1] then [stp1] with [step1']
if [/fhstp1* dir1] then [stp8] with [step8']
if [ fhstp1*/dir1] then [stp5] with [step5']
if [/fhstp1*/dir1] then [stp6] with [step6']

while [stp8]
if [ fhstp1* dir1] then [stp2] with [step2']
if [/fhstp1* dir1] then [stp1] with [step1']
if [ fhstp1*/dir1] then [stp6] with [step6']
if [/fhstp1*/dir1] then [stp7] with [step7']
```

Figure 19.b State Equation Entry File (Continued)

CMOS sequencer solutions

```
            "motor comtroller #2"
while [stp1a]
if [ fhstp2* dir2] then [stp3a] with [step3a']   "full step forward"
if [/fhstp2* dir2] then [stp2a] with [step2a']   "half step forward"
if [ fhstp2*/dir2] then [stp7a] with [step7a']   "full step backward"
if [/fhstp2*/dir2] then [stp8a] with [step8a']   "half step backward"

while [stp2a]
if [ fhstp2* dir2] then [stp4a] with [step4a']
if [/fhstp2* dir2] then [stp3a] with [step3a']
if [ fhstp2*/dir2] then [stp8a] with [step8a']
if [/fhstp2*/dir2] then [stp1a] with [step1a']

while [stp3a]
if [ fhstp2* dir2] then [stp5a] with [step5a']
if [/fhstp2* dir2] then [stp4a] with [step4a']
if [ fhstp2*/dir2] then [stp1a] with [step1a']
if [/fhstp2*/dir2] then [stp2a] with [step2a']

while [stp4a]
if [ fhstp2* dir2] then [stp6a] with [step6a']
if [/fhstp2* dir2] then [stp5a] with [step5a']
if [ fhstp2*/dir2] then [stp2a] with [step2a']
if [/fhstp2*/dir2] then [stp3a] with [step3a']

while [stp5a]
if [ fhstp2* dir2] then [stp7a] with [step7a']
if [/fhstp2* dir2] then [stp6a] with [step6a']
if [ fhstp2*/dir2] then [stp3a] with [step3a']
if [/fhstp2*/dir2] then [stp4a] with [step4a']

while [stp6a]
if [ fhstp2* dir2] then [stp8a] with [step8a']
if [/fhstp2* dir2] then [stp7a] with [step7a']
if [ fhstp2*/dir2] then [stp4a] with [step4a']
if [/fhstp2*/dir2] then [stp5a] with [step5a']

while [stp7a]
if [ fhstp2* dir2] then [stp1a] with [step1a']
if [/fhstp2* dir2] then [stp8a] with [step8a']
if [ fhstp2*/dir2] then [stp5a] with [step5a']
if [/fhstp2*/dir2] then [stp6a] with [step6a']

while [stp8a]
if [ fhstp2* dir2] then [stp2a] with [step2a']
if [/fhstp2* dir2] then [stp1a] with [step1a']
if [ fhstp2*/dir2] then [stp6a] with [step6a']
if [/fhstp2*/dir2] then [stp7a] with [step7a']
```

Figure 19.c State Equation Entry File (end)

CMOS sequencer solutions

PLC42VA12 DMA APPLICATIONS

The PLC42VA12 contains 10 flip-flops that may flexibly be configured to build counters, shifters or any customized state machine required. With today's 32-bit microprocessors, there is a need for user-designed, system-specific DMA controllers that can generate addresses or count nibbles, bytes, half-words or words. Applications for these controllers include I/O concentration and cache subsystem updating. Typically, these devices can be preset or cleared and count (up) by 1, 2, or 4 depending on the chosen circumstances. Two solutions for the same problem are presented in this section to illustrate solving the problem with Signetics AMAZE 1.9 or SNAP design software. Although these tools are similar, their approach and syntax are distinct. The AMAZE design files are presented in Figure 20 and the SNAP files in Figure 21.

```
@DEVICE TYPE
PLC42VA12
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@PINLIST

"<--------FUNCTION-------->    <--REFERENCE-->"
"PINLABEL    PIN #  PIN_FCT    PIN_ID     OE_CTRL"
CLK           1       CLK       IO/CLK      -  ;
MODE0         2       I         I1          -  ;
MODE1         3       I         I2          -  ;
RST           4       I         I3          -  ;
LOAD          5       I         I4          -  ;
N/C           6       I         I5          -  ;
N/C           7       I         I6          -  ;
N/C           8       I         I7          -  ;
N/C           9       I         I8          -  ;
N/C          10       /B        B0          DO ;
CO           11       O         B1          D1 ;
GND          12       0V        GND         -  ;
/OE          13       I         /OE/I9      -  ;
OUTA         14       /BJ       M0          EM0 DM0;
OUTB         15       /BJ       M1          EA  DM1;
OUTC         16       /BJ       M2          EA  DM2;
OUTD         17       /BJ       M3          EA  DM3;
OUTE         18       /BJ       M4          EA  DM4;
OUTF         19       /BJ       M5          EB  DM5;
OUTG         20       /BJ       M6          EB  DM6;
OUTH         21       /BJ       M7          EB  DM7;
OUTI         22       /BJ       M8          EB  DM8;
OUTJ         23       /BJ       M9          EM9 DM9;
VCC          24       +5V       VCC         -  ;
```

Figure 20.a Amaze Design Files

CMOS sequencer solutions

```
@INTERNAL FLIP FLOP LABELS  "Label declared for FF labels."
" P0  P1  P2  P3  P4  P5  P6  P7  P8  P9"
  N/C N/C N/C N/C N/C N/C N/C N/C N/C N/C
@REGISTER LOAD  " LA, LB, LM0, LM9 "
LA  = LOAD;
LB  = LOAD;
Lm0 = LOAD;
Lm9 = LOAD;
@COMMON PRODUCT TERM  "CPT_label = (expression)"
@COMPLEMENT ARRAY  "/C = /(---);"
@I/O DIRECTION  " D0,D1, DM0..DM9 "
@INTERNAL CLOCKS  "CK0 .. DK9 = EXPRESSION;"
@FLIP FLOP CONTROL " FC = --- ; "
FC=1;
@ASYNCHRONOUS PRESET/RESET  "PA, PB, PM0, PM9, RM0..RM9"
pm0=rst; pm9=rst;     "preset registers to force pin LOW"
pa=rst; pb=rst;
@FLIP FLOP MODE  " M0 .. MN "

@LOGIC EQUATION
                "  mode1  mode0    function
                     0      0     count by 1
                     0      1     count by 2
                     1      0     count by 4
                     1      1      illegal
                "
 /OUTA : J = 1*/load*/mode0*/model                    "load disables P-terms"
           + mode0*/load                              "force 0 for count by 2"
           + mode1*/load;                             "or count by 4"
         K = 1*/load*/mode0*/model;
 /OUTB : J = outa*/load*/mode0*/model
           + 1*/model*mode0*/load
           + model*/mode0*/load;                      "force 0 for count by four"
         K = outa*/load*/mode0*/model
           + 1*/model*mode0*/load;
 /OUTC : T = outa*outb*/load*/mode0*/model
           + outb*/model*mode0*/load
           + 1*model*/mode0*/load;
 /OUTD : T = outa*outb*outc*/load*/mode0*/model       "count by 1"
           + outb*outc*/model*mode0*/load             "count by 2"
           + outc*model*/mode0*/load;                 "count by 4"
 /OUTE : T = outa*outb*outc*outd*/load*/mode0*/model
           + outb*outc*outd*/model*mode0*/load
           + outc*outd*model*/mode0*/load;
 /OUTF : T = outa*outb*outc*outd*oute*/load*/mode0*/model
           + outb*outc*outd*oute*/model*mode0*/load
           + outc*outd*oute*model*/mode0*/load;
 /OUTG : T = outa*outb*outc*outd*oute*outf*/load*/mode0*/model
           + outb*outc*outd*oute*outf*/model*mode0*/load
           + outc*outd*oute*outf*model*/mode0*/load;
 /OUTH : T = outa*outb*outc*outd*oute*outf*outg*/load*/mode0*/model
           + outb*outc*outd*oute*outf*outg*/model*mode0*/load
           + outc*outd*oute*outf*outg*model*/mode0*/load;
 /OUTI : T = outa*outb*outc*outd*oute*outf*outg*outh*/load*/mode0*/model
           + outb*outc*outd*oute*outf*outg*outh*/model*mode0*/load
           + outc*outd*oute*outf*outg*outh*model*/mode0*/load;
 /OUTJ : T = outa*outb*outc*outd*oute*outf*outg*outh*outi*/load*/mode0*/model
           + outb*outc*outd*oute*outf*outg*outh*outi*/model*mode0*/load
           + outc*outd*oute*outf*outg*outh*outi*model*/mode0*/load;
        CO = outa*outb*outc*outd*oute*outf*outg*outh*outi*outj*/load*/mod0*/model
           + outb*outc*outd*oute*outf*outg*outh*outi*outj*/model*mode0*/load
           + outc*outd*oute*outf*outg*outh*outi*outj*model*/mode0*/load;
```

Figure 20.b AMAZE Design Files (Continued)

CMOS sequencer solutions

```
@PINLIST
CLK  I ;MODE0   I ;MODE1  I;RST I;LOAD I;CO  O;TOE  I;
OUTA O;OUTB  O;OUTC O;OUTD B;OUTE B;OUTF B;OUTG  G;OUTH B;OUTI B;OUTJ B;

@Logic Equations
               "  model  mode0    function
                    0      0     count by 1
                    0      1     count by 2
                    1      0     count by 4
                    1      1       illegal
               "
QUOTA.J = 1*/load*/mode0*/model1            "load disables P-terms"
        + mode0*/load                       "force 0 for count by 2"
        + model1*/load          + XOUTA;    "or count by 4"
QUOTA.K = 1*/load*/mode0*/model1  + YOUTA;  "XOUTA,YOUTA are outputs of "
                                            "tristate inputs "
QOUTB.J = outa*/load*/mode0*/model1
        + 1*/model1*mode0*/load
        + model1*/mode0*/load   + XOUTB;    "force 0 count by four"
QOUTB.K = outa*/load*/mode0*/model1
        + model1*/mode0*/load   + YOUTB;

DOUTC = outa*outb/load*/mode0*/model1
        + outb*/model1*mode0*/load;
QOUTC.J = DOUTC + XOUTC;
QOUTC.K = DOUTC + YOUTC;

DOUTD = outa*outb*outc*/load*/mode0*model1          "count by 1"
        + outb*outc*/model1*mode0*/load             "count by 2"
        + outc*model1*/mode0*/load;                 "count by 4"
QOUTD.J = DOUTD + XOUTD;
QOUTD.K = DOUTD + YOUTD;

DOUTE = outa*outb*outc*outd*/load*/mode0*/model1
        + outb*outc*outd*/model1*mode0*/load
        + outc*outd*model1*/mode0*/load;
QOUTE.J = DOUTE + XOUTE;
QOUTE.K = DOUTE + YOUTE;

DOUTF = outa*outb*outc*outd*oute*/load*/mode0*model1
        + outb*outc*outd*oute*/model1*mode0*/load
        + outc*outd*oute*model1*/mode0*/load;
QOUTF.J = DOUTF + XOUTF;
QOUTF.K = DOUTF + YOUTF;

DOUTG = outa*outb*outc*outd*oute*outf*/load*/mode0*/model1
        + outb*outc*outd*oute*outf*/model1*mode0*/load
        + outc*outd*oute*outf*model1*/mode0*/load;
QOUTG.J = DOUTG + XOUTG;
QOUTG.K = DOUTG + YOUTG;

DOUTH = outa*outb*outc*outd*oute*outf*outg*/load*/mode0*/model1
        + outb*outc*outd*oute*outf*outg*/model1*mode0*/load;
        + outc*outd*oute*outf*outg*model1*/mode0*/load;
QOUTH.J = DOUTH + XOUTH;
QOUTH.K = DOUTH + YOUTH;

DOUTI = outa*outb*outc*outd*oute*outf*outg*outh*/load*/mode0*/model1
        + outb*outc*outd*oute*outf*outg*outh*/model1*mode0*/load
        + outc*outd*oute*outf*outg*outh*model1*/mode0*/load;
QOUTI.J = DOUTI + XOUTI;
QOUTI.K = DOUTI + YOUTI;

DOUTJ = outa*outb*outc*outd*oute*outf*outg*outh*outi*/load*/mode0*/model1
        + outb*outc*outd*oute*outf*outg*outh*outi*/model1*mode0*/load
        + outc*coutd*oute*outf*outg*outh*outi*model1*/mode0*/load;
QOUTJ.J = DOUTJ + XOUTJ;
QOUTJ.K = DOUTJ + YOUTJ;
```

Figure 21.a SNAP Files

CMOS sequencer solutions

```
CO = outa*outb*outc*outd*oute*outf*outg*outh*outi*outj*/load*/mode0*/mode1
     + outb*outc*outd*oute*outf*outg*outh*outi*outj*/mode1*mode0*/load
     + outc*outd*oute*outf*outg*outh*outi*outj*mode1*/mode0*/load;

" Reset for all flip-flops "
QOUTA.RST = RST;
QOUTB.RST = RST;
QOUTC.RST = RST;
QOUTD.RST = RST;
QOUTE.RST = RST;
QOUTF.RST = RST;
QOUTG.RST = RST;
QOUTH.RST = RST;
QOUTI.RST = RST;
QOUTJ.RST = RST;

" Flip-flops are followed by tristate outputs which drive the pin "
OUTA = /QOUTA;
OUTB = /QOUTB;
OUTC = /QOUTC;
OUTD = /QOUTD;
OUTE = /QOUTE;
OUTF = /QOUTF;
OUTG = /QOUTG;
OUTH = /QOUTH;
OUTI = /QOUTI;
OUTJ = /QOUTJ;
OUTA.OE = TOE;
OUTB.OE = TOE;
OUTC.OE = TOE;
OUTD.OE = TOE;
OUTE.OE = TOE;
OUTF.OE = TOE;
OUTG.OE = TOE;
OUTH.OE = TOE;
OUTI.OE = TOE;
OUTJ.OE = TOE;
```

Figure 21.b SNAP Files (Continued)

CMOS sequencer solutions

```
"Pins are fed back to flip-flops using tristate inputs (FF load)"
XOUTA = /OUTA;       "feed.back to J is inverted"
YOUTA =  OUTA;       "feed-back to K IS NOT inverted"
XOUTB = /OUTB;
YOUTB =  OUTB;
XOUTC = /OUTC;
YOUTC =  OUTC;
XOUTD = /OUTD;
YOUTD =  OUTD;
XOUTE = /OUTE;
YOUTE =  OUTE;
XOUTF = /OUTF;
YOUTF =  OUTF;
XOUTG = /OUTG;
YOUTG =  OUTG;
XOUTH = /OUTH;
YOUTH =  OUTH;
XOUTI = /OUTI;
YOUTI =  OUTI;
XOUTJ = /OUTJ;
YOUTJ =  OUTJ;
XOUTA.LD = LOAD;
XOUTB.LD = LOAD;
XOUTC.LD = LOAD;
XOUTD.LD = LOAD;
XOUTE.LD = LOAD;
XOUTF.LD = LOAD;
XOUTG.LD = LOAD;
XOUTH.LD = LOAD;
XOUTI.LD = LOAD;
XOUTJ.LD = LOAD;
YOUTA.LD = LOAD;
YOUTB.LD = LOAD;
YOUTC.LD = LOAD;
YOUTD.LD = LOAD;
YOUTE.LD = LOAD;
YOUTF.LD = LOAD;
YOUTG.LD = LOAD;
YOUTH.LD = LOAD;
YOUTI.LD = LOAD;
YOUTJ.LD = LOAD;
```

Figure 21.c SNAP Files (End)

Programmable High-Speed Decoders (PHD) Application Note

INTRODUCTION

A recent Philips Components–Signetics Application Note described how to perform microprocessor "High Speed Address Decode". This document focused on the use of Philips Components–Signetics D-speed PLS devices and 7.5 nanosecond PAL®-type devices to drive the chip-selects, output enables, read and write strobes for static RAM, dynamic RAM, EPROM and I/O in a microprocessor system. In the meantime, microprocessors got faster and so did memories. Several customers had requested faster parts, but when Philips Components–Signetics new product definition engineers examined the customer designs, it was noted that the logical OR function was seldom used. In fact the OR had become a time bottleneck for their system throughput. With this in mind, Philips Components–Signetics resurrected an older architecture on a blazing fast high-speed bipolar process and began a new family of parts called the Programmable High-Speed Decoders (PHD).

The current offering consists of the 20-pin PHD16N8 and the 68-pin PHD48N22. Figure 1 tells 90% of the story. As shown, the PHD16N8–5 delivers a decoded output in less than 5 nanoseconds. The PHD48N22–7 has two basic delay path times—one delay at 6.5 nanoseconds, and one delay at 7.0 nanoseconds. Remember, this is the worst-case maximum time delay for the PHD16N8 with the output delivering 24mA of I_{OL}.

It may be noted that the PHD16N8–5 contains no OR function at all. It is targeted to help designers milk the maximum performance from high-speed cache systems and get full performance at the best system cost. If an OR-type function is needed, outputs may be wire-ANDed generating a composite "AND–OR–INVERT" function similar to the 16L8-type part. This requires an external pull-up resistor and will incur an "RC" time delay speed penalty. It should also be noted that the PHD16N8–5 is pin-compatible to the TIBPAD part, but is much faster.

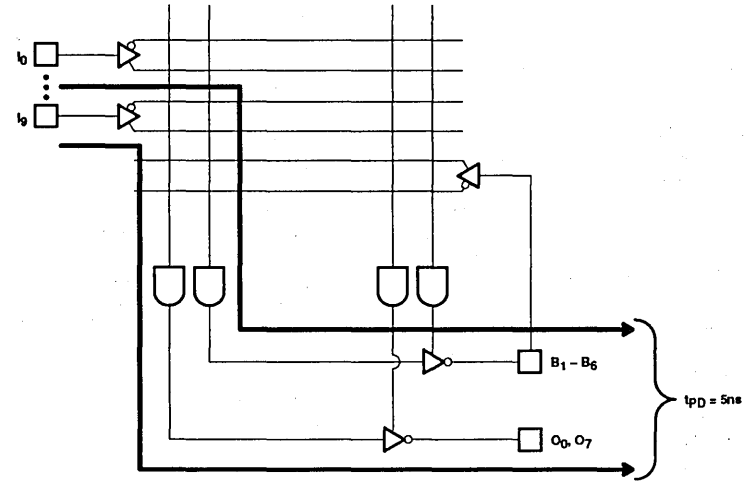
The PHD48N22–7, on the other hand, has enough inputs and outputs to form the complete system decode needs of a whole microprocessor system. This includes the I/O as well as the memory space, in general. The PHD48N22–7 does include four sites of programmable array logic where three OR patches accommodate 7 product terms and one patch handles 12 product terms. These were intended to handle such microprocessor pins as the wait line or interrupt lines where sums of products might be needed. Other uses are possible.

In either part, bidirectionals can be configured as outputs or inputs to select more devices or resolve more address inputs. It is interesting to note that the PHD48N22 device can resolve a single byte address from a 4 gigabyte address space and distinguish memory from I/O and distinguish input from output data directions. There is no need for fractured memory / I/O spaces with big gaps. RISC design can cleanly have embedded cache in the middle of DRAM or EPROM space with registers sprinkled throughout as needed.

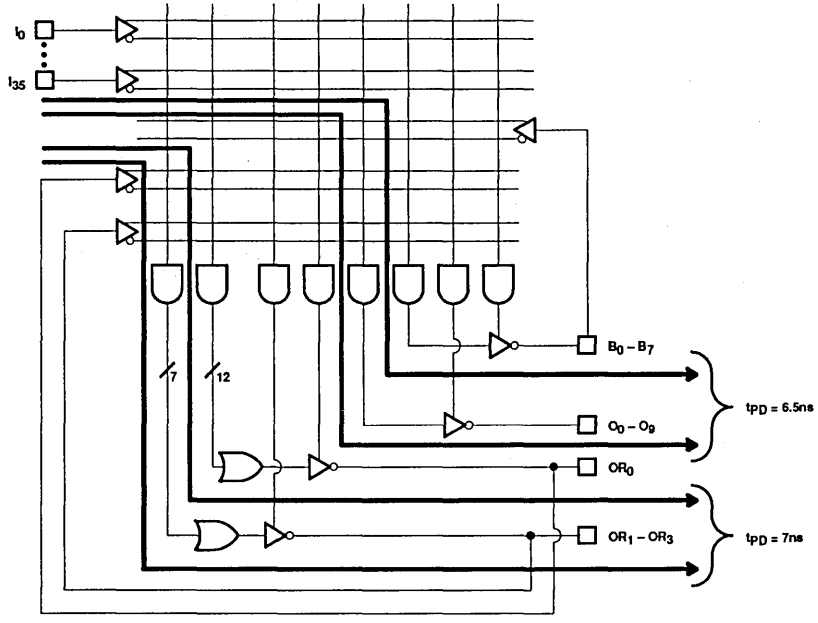
To maintain full performance while driving a memory array, the current drive has been maintained at I_{OL} = 24mA. However, there is a restriction in that full speed is guaranteed for a single output asserted at a time. This is due more to the way things are specified and measured than anything else. As more outputs are driven Low (at full rated current), the ground pins must accommodate the Low-going output current which results in possible ground rise. This affects the apparent speed of the part as measured, so derating of the speed must be done with additional loading. The derating curves for the PHD16N8 and PHD48N22 are shown in Figure 2.

The complete logic diagrams for the PHD16N8–5 and PHD48N22–7 are shown in Figure 3 and Figure 4.

Programmable high-speed decoders (PHD)



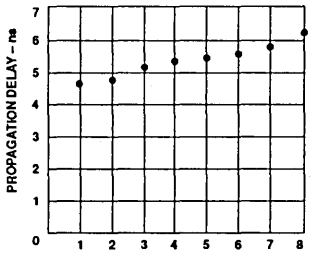
(a) The PHD16N8–5 Architecture



(b) The PHD48N22–7 Architecture

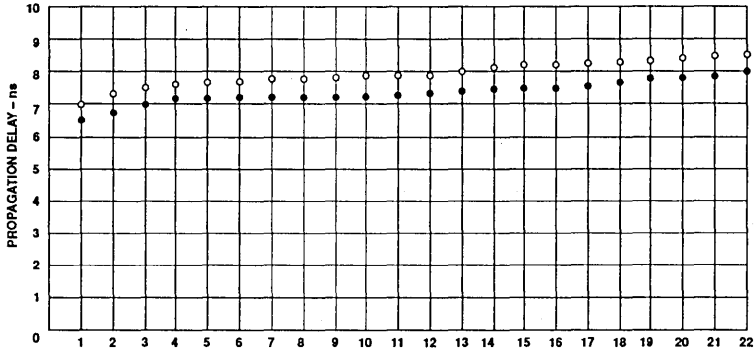
Figure 1. The Current PHD Family

Programmable high-speed decoders (PHD)



TEST CONDITIONS: T_{amb} = 75°C;
V_{CC} = 4.75V; C_L = 50pF;
R_1 = 200Ω; R_2 = 390Ω

PHD16N8–5 Propagation Delay Values



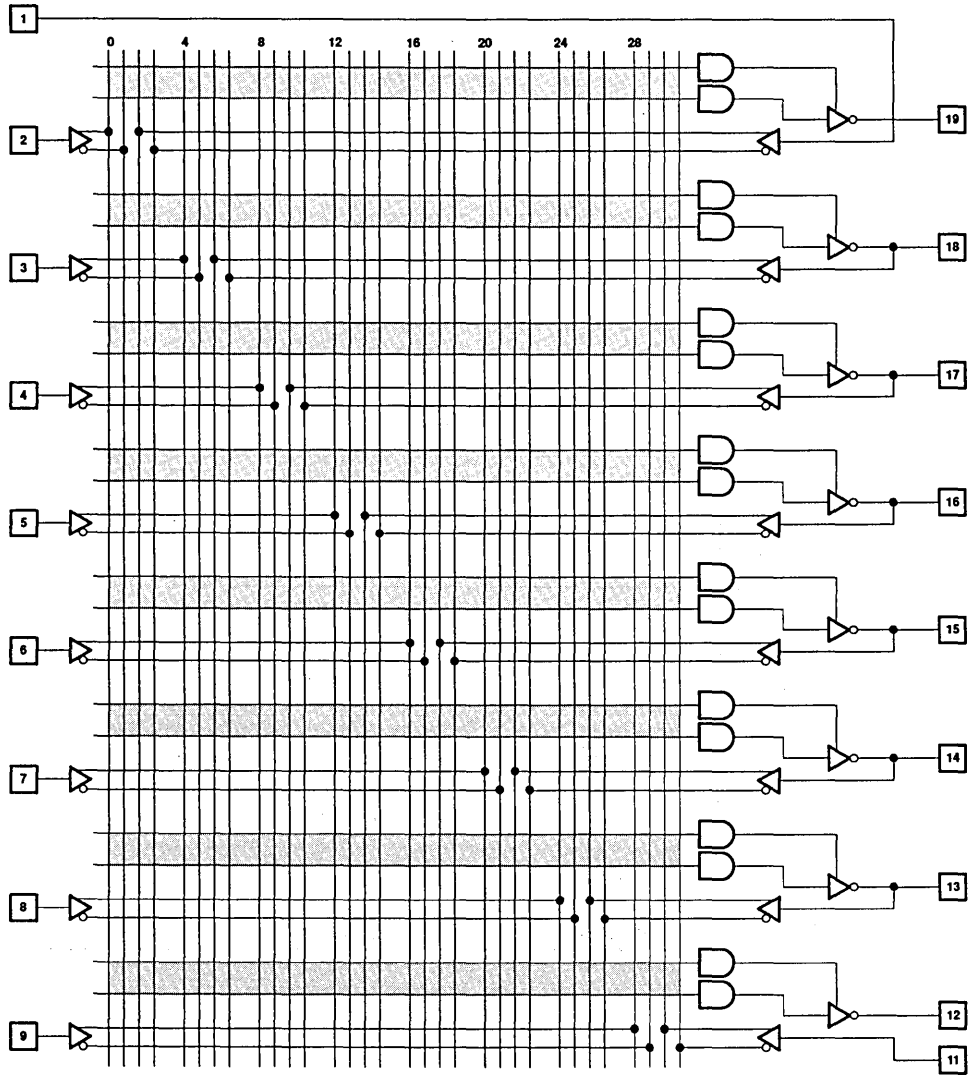
NOTE: ○ = OR Outputs
● = B/O Outputs

TEST CONDITIONS: T_{amb} = 75°C;
V_{CC} = 4.75V; C_L = 50pF;
R_1 = 200Ω; R_2 = 390Ω

PHD48N22–7 Propagation Delay Values

Figure 2. Propagation Delay vs. Number of Outputs Switching

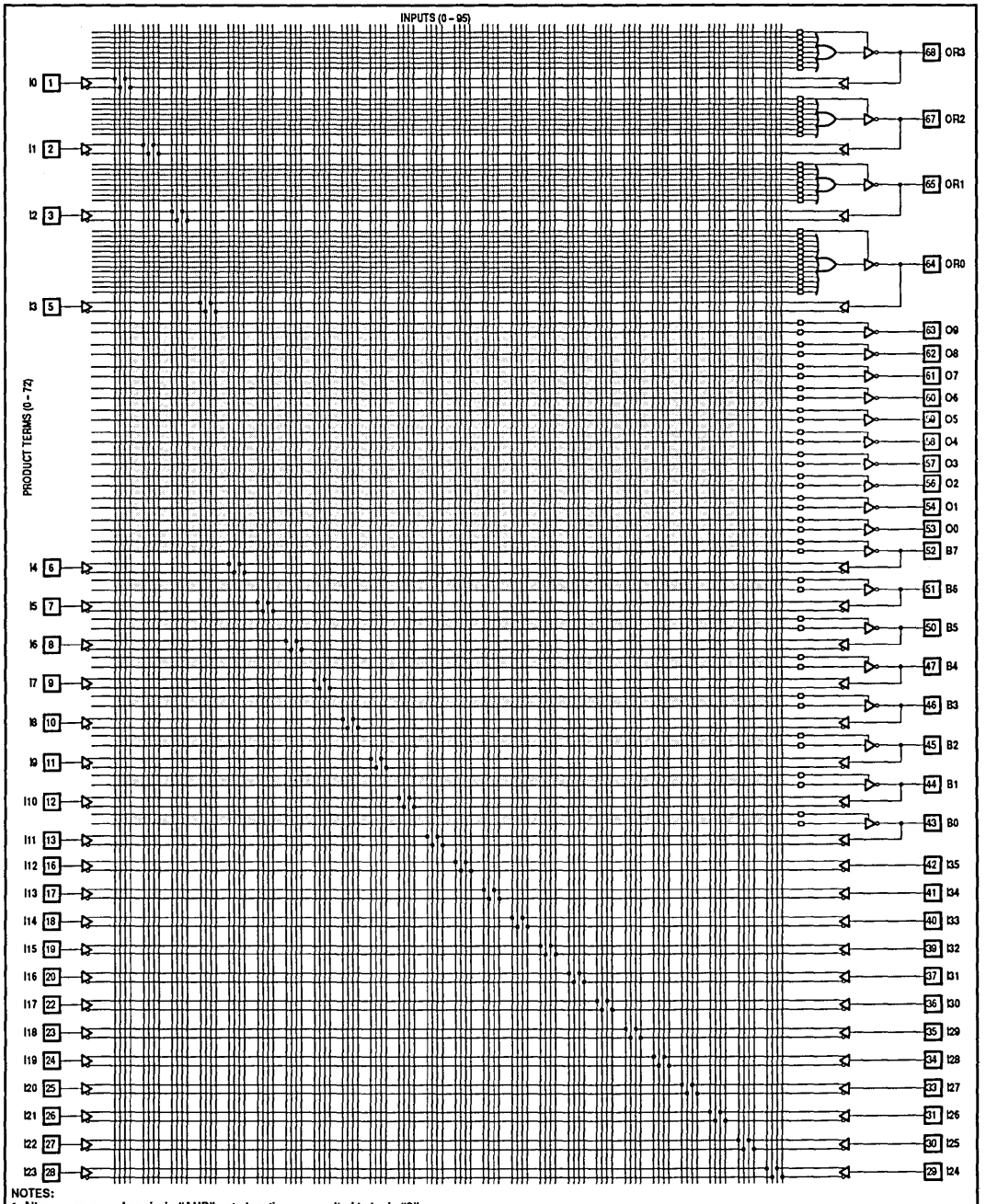
Programmable high-speed decoders (PHD)



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. Programmable connections

Figure 3. PHD16N8–5 Logic Diagram

Programmable high-speed decoders (PHD)



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ⊙ Programmable connections

Figure 4. PHD48N22–7 Logic Diagram

Programmable high-speed decoders (PHD)

SUPPORT

The PHD family is targeted for support on Philips Components–Signetics AMAZE and SNAP products. Currently, the PHD16N8–5 is supported on AMAZE 1.8 and higher, and the PHD48N22–7 supported on AMAZE 1.9. SNAP support will occur at a later date. Because of the architectural simplicity, basic logic equation and compilation is the net development support required, in the AMAZE environment.

Programming support changes very rapidly as time proceeds. Current support for the PHD16N8–5 includes the Data I/O Model 29B (Version 11), the Unisite (Version 2.8), and the Model 60 (Version 15). Support on the STAG ZL30A (Version 36) is also available. Current support for the PHD48N22–7 is on the Strebor Low Cost Programmer, but other manufacturers will be qualified soon.

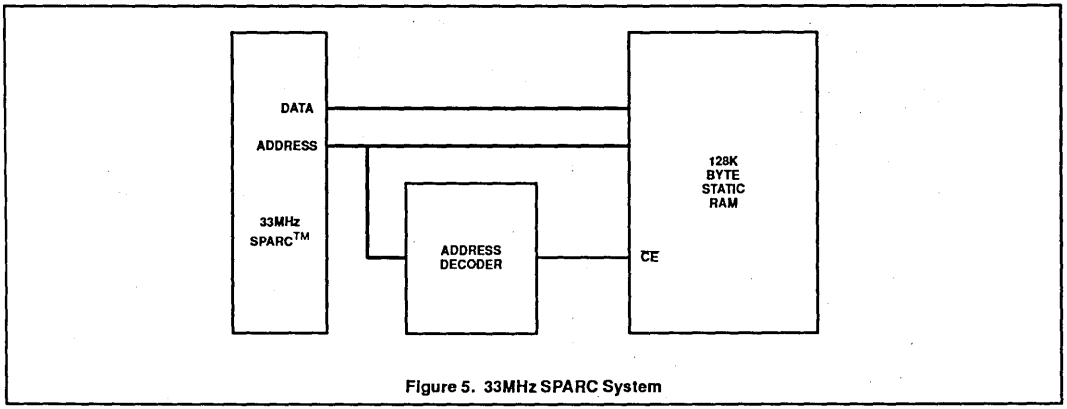


Figure 5. 33MHz SPARC System

Programmable high-speed decoders (PHD)

Table 1. Cost Savings of PHD16N8–5

33MHz SPARC SOLUTION		
DEVICE	RAM SPEED	RAM COST
PHD16N8–5	25ns	$576
7.5ns 16L8	20ns	$1,216

PHD vs. PAL COST ANALYSIS

The PHD16N8 is an ideal device when interfacing high-speed microprocessors to memory. Figure 5 is a diagram showing a 33MHz SPARC™ processor, an address decoder, and 128K bytes of memory. The goal of this circuit is to develop a system that has zero wait states and analyze the trade-offs between using 7.5ns 16L8s and the PHD16N8–5.

In order to have zero wait states, the access time of the RAMs plus the decoder must be less than 31ns. With a 5ns PHD16N8, 25ns static RAMs can be used. With a 7.5ns 16L8 PAL, you must use RAMs with a maximum access time of 22.5ns. This means that the designer must use either 15ns or 20ns RAMs. Cypress Semiconductor makes 64Kx1 static RAMs that have speeds of 20ns and 25ns. We will use these devices in our analysis.

We will need 64 static RAMs to implement a 128K-byte memory using 64Kx1 devices. Using the PHD16N8 and 25ns static RAMs, we obtain an approximate cost of $576. If we were to use a 7.5ns 16L8 and 20ns static RAMs, we would obtain an approximate cost of $1,216. This shows that one can save $640 by using the PHD16N8. This is a considerable cost savings and in many cases can give designers a substantial edge over those competitors who use the PAL solution. A summary of these results is shown in Table 1.

It should be noted that this system is quite simple, but easily illustrates the potential value of the PHD16N8. If one wanted more margin in one's design, the cost differential would be much greater. As systems approach 40MHz, the 16L8 designs will be forced into using 15ns static RAMs to have zero wait states. The PHD16N8 can probably stay with 20ns RAMs. This will make the savings in the cost of RAM in the thousands of dollars. This savings might even buy the processor!

68030 ADDRESS DECODING

Philips Components–Signetics PHD48N22 decoder is not only very high speed but also has a very wide input structure. This combination allows high-performance systems to be constructed in a minimal space without sacrificing decoding resolution. The following example demonstrates the use of the PHD48N22 as well as the PHD16N8 in a 33MHz 68030 system.

This example uses one each 48N22 and 16N8 PHD decoders to interface together a 33MHz 68030 processor, 33MHz 68882 Floating-Point Coprocessor, 128K bytes of 35ns static RAM, 64K bytes of 200ns EPROM and a SCN2692 Dual port UART. Additional logic required amounts to only one 16-pin and one 24-pin device!

The schematic is shown in Figure 6 with AMAZE listings of the PHD devices in Figures 7 and 8. As can be seen from the schematic, the PHD48N22 handles all of the RAM, PROM and DUART decoding. Notice the number of 68030 address lines input into the 48N22: A31 through A8 and A1 and A0. Depending upon the application, the 48N22 can decode down to the byte level in 6.5ns! The PHD16N8 decodes an early chip-select signal for the 33MHz 68882 FPCP in less that the required 5ns.

Accesses to the RAM produce an immediate acknowledge (DSACK) from the 48N22 to the 68030, allowing it to run full speed in asynchronous mode. Since the EPROM and DUART operate at a much slower speed than the 68030, two additional devices, a 74F191 counter and a 74F646 transceiver/register were used in this example. If either the EPROM or DUART are accessed, the counter will count CPU clock cycles and delay the 48N22's assertion of a DSACK signal. The 'F646 is used to quickly 3–State signals from the EPROM and DUART. Also, in conjunction with the counter and 48N22, a read of a DUART register first causes information to be read into a register of the 'F646 which is then read by the 68030 while the DUART's output is disabled. This was done since a DUART specification (t_{RWD}) requires 200ns between reads or writes. Therefore, accesses to the DUART are controlled by the counter, 48N22, and 'F646. Software restrictions are not required.

Programmable high-speed decoders (PHD)

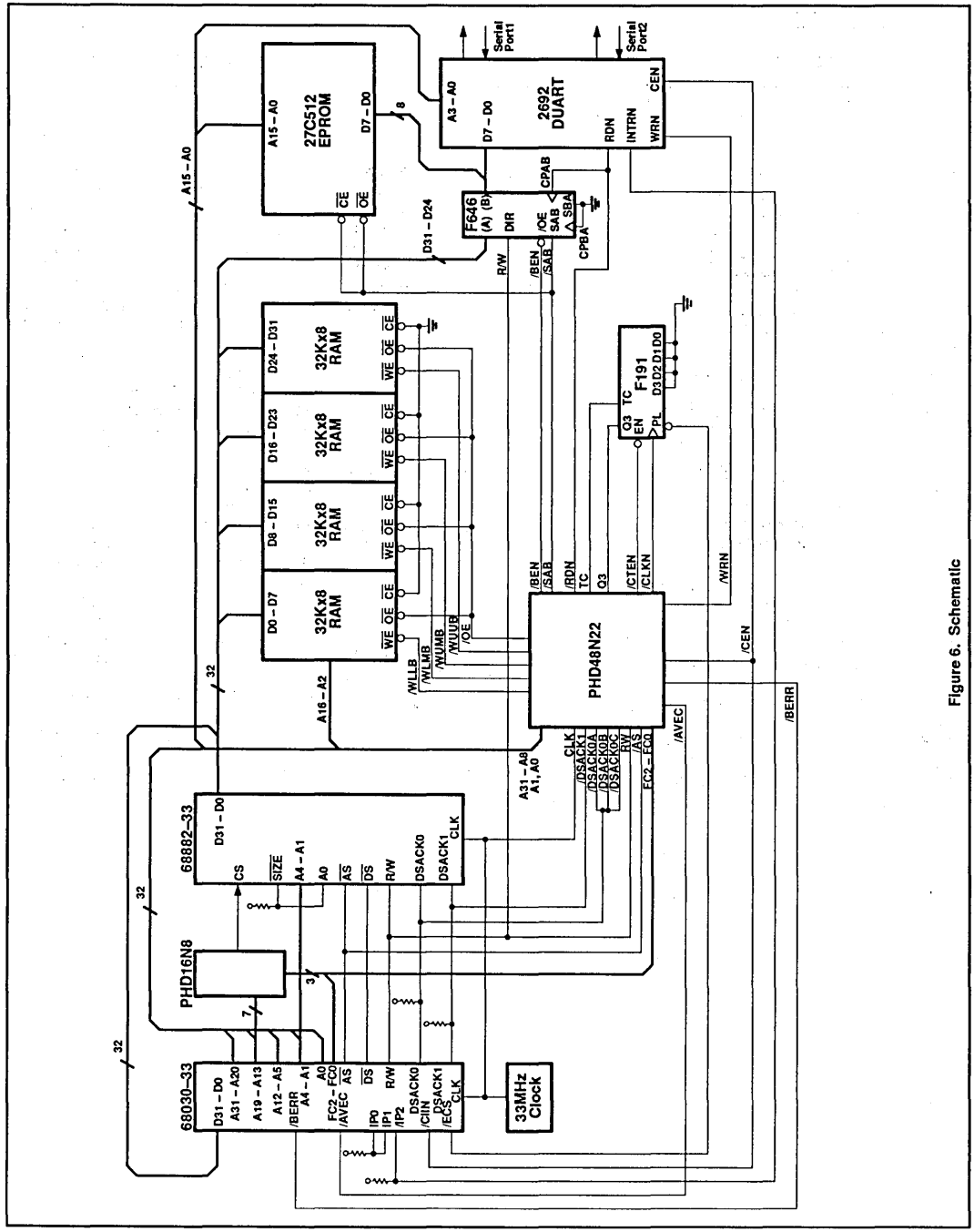


Figure 6. Schematic

Programmable high-speed decoders (PHD)

```
@DEVICE TYPE
PHD48N22
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
 33MHz 68030 system deocder example
@PINLIST

"<--------FUNCTION-------->    <--REFERENCE-->"
"PINLABEL    PIN #  PIN_FCT    PIN_ID      OE_CTRL"
A0            1       I          I0          -  ;
A1            2       I          I1          -  ;
/AS           3       I          I2          -  ;
VCC3          4       PWR        VCC3        -  ;
Q3            5       I          I3          -  ;
TC            6       I          I4          -  ;
CLK           7       I          I5          -  ;
A8            8       I          I6          -  ;
A9            9       I          I7          -  ;
A10          10       I          I8          -  ;
A11          11       I          I9          -  ;
A12          12       I          I10         -  ;
A13          13       I          I11         -  ;
GND5         14       PWR        GND5        -  ;
GND1         15       PWR        GND1        -  ;
A14          16       I          I12         -  ;
A15          17       I          I13         -  ;
A16          18       I          I14         -  ;
A17          19       I          I15         -  ;
A18          20       I          I16         -  ;
VCC2         21       PWR        VCC2        -  ;
A19          22       I          I17         -  ;
A20          23       I          I18         -  ;
A21          24       I          I19         -  ;
A22          25       I          I20         -  ;
A23          26       I          I21         -  ;
A24          27       I          I22         -  ;
A25          28       I          I23         -  ;
A26          29       I          I24         -  ;
A27          30       I          I25         -  ;
A28          31       I          I26         -  ;
GND3         32       PWR        GND3        -  ;
A29          33       I          I27         -  ;
A30          34       I          I28         -  ;
A31          35       I          I29         -  ;
FC0          36       I          I30         -  ;
FC1          37       I          I31         -  ;
VCC4         38       PWR        VCC4        -  ;
FC2          39       I          I32         -  ;
SIZ0         40       I          I33         -  ;
SIZ1         41       I          I34         -  ;
RW           42       I          I35         -  ;
N/C          43       /B         B0          D0 ;
/OE          44       /B         B1          D1 ;
/CPU         45       /B         B2          D2 ;
N/C          46       /B         B3          D3 ;
/SAB         47       /B         B4          D4 ;
GND6         48       PWR        GND6        -  ;
GND2         49       PWR        GND2        -  ;
/ASN         50       /B         B5          D5 ;
/WUUB        51       /B         B6          D6 ;
/CEN         52       /B         B7          D7 ;
```

Figure 7. PHD48N22–7 AMAZE Listing (begins)

Programmable high-speed decoders (PHD)

```
/BEN          53        /O          O0          OE0  ;
/RDN          54        /O          O1          OE1  ;
VCC1          55        PWR         VCC1        -  ;
/WRN          56        /O          O2          OE2  ;
/DSACK0a      57        /O          O3          OE3  ;
/DSACK0b      58        /O          O4          OE4  ;
/DSACK0c      59        /O          O5          OE5  ;
/DSACK1       60        /O          O6          OE6  ;
/BERR         61        /O          O7          OE7  ;
/AVEC         62        /O          O8          OE8  ;
/CLKN         63        /O          O9          OE9  ;
/CTEN         64        /B          OR0         DO0  ;
/WUMB         65        /B          OR1         DO1  ;
GND4          66        PWR         GND4        -  ;
/WLMB         67        /B          OR2         DO2  ;
/WLLB         68        /B          OR3         DO3  ;

@ COMMON PRODUCT TERM  "CPT_label = <expression>;"

        "RAM, PROM, and USART address definitions"
PROMADR = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A18*/A18*/A17*/A16;
RAMADDR = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A18*/A18*/A17;
UARTADR = //A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A18*/A18*/A17*A16*/A15*/A14*/A13*/A12*/A11*/A10*/A9*/A8;
UORPROM = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A18*/A18*/A17;
ASD     =  AS * ASN;  "eliminate hazard ANDing AS with /CPU"

@I/O DIRECTION  " D0 .. D7, OE0 .. OE9, DO0 .. DO3 "

D1=1; D2=1; D4=1; D5=1; D6=1; D7=1;
DO0=1; D01=1; D02=1; D03=1;

        "enables for DSACK0 and DSACK1
OE3= PROMADR*/CPU*ASD*Q3 ;  "acknowledge PROM  8 bits"
OE4= UARTADR*/CPU*ASD*TC ;  "acknowledge USART 8 bits"
OE5= RAMADDR*/CPU*ASD ;     "acknowledge RAM 32 bits"
OE6= RAMADDR*/CPU*ASD ;     "acknowledge RAM 32 bits"

@LOGIC EQUATION

/ASN = /(AS); "delay AS"

        "these outputs are controlled by OE3, OE4, OE5, OE6"
/DSACK0a = /(PROMADR*/CPU*ASD*Q3);  "drive high before 3-state"
/DSACK0b = /(UARTADR*/CPU*ASD*TC);
/DSACK0c = /(RAMADDR*/CPU*ASD);
/DSACK1  = /(RAMADDR*/CPU*ASD);

        "CPU address space decoding"
/CPU  = /(FC0*FC1*FC2);
/AVEC = /(FC0*FC1*FC0*A19*A18*A17*A16); "interrupt ack forces autovector"

        "RAM output enable"
/OE   = /(RAMADDR*/CPU*RW);              "enable all RAM outputs upon read"
```

Figure 7. PHD48N22–7 AMAZE Listing (continued)

Programmable high-speed decoders (PHD)

```
        "RAM write strobes"
/WLLB = /(RAMADDR*/RW*A1*A0*/CPU*ASD          "directly addressed, any size"
        + RAMADDR*/RW*A0*SIZ1*SIZ0*/CPU*ASD   "odd alignment, three byte size"
        + RAMADDR*/RW*/SIZ1*/SIZ0*/CPU*ASD    "any RAM address, long word size"
        + RAMADDR*/RW*A1*SIZ1*/CPU*ASD);      "word aligned, word or 3 byte size"

/WLMB = /(RAMADDR*/RW*A1*/A0*/CPU*ASD         "directly addressed, any size"
        + RAMADDR*/RW*/A1*/SIZ1*/SIZ0*/CPU*ASD "word aligned, long word size"
        + RAMADDR*/RW*/A1*SIZ1*SIZ0*/CPU*ASD  "word aligned, three byte size"
        + RAMADDR*/RW*/A1*A0*SIZ0*/CPU*ASD);  "word aligned, word or long word"

/WUMB = /(RAMADDR*/RW*/A1*A0*/CPU*ASD         "directly addressed, any size"
        + RAMADDR*/RW*/A1*/SIZ0*/CPU*ASD      "word aligned, byte or 3 byte size"
        + RAMADDR*/RW*/A1*SIZ1*/CPU*ASD);     "word aligned, word or long word"

/WUUB = /(RAMADDR*/RW*/A1*/A0*/CPU*ASD);      "directly addressed, any size"


        "DUART and PROM address decoding"
/CEN = /(UARTADR*/CPU*ASD);          "this signal is also /CIIN"
/RDN = /(UARTADR*/CPU*ASD*/Q3*RW);
/WRN = /(UARTADR*/CPU*ASD*/Q3*/RW);
/BEN = /(UORPROM*/CPU*ASD);          "enable F646 for USART or PROM"
/SAB = /(PROMADR*/CPU*ASD*RW);       "F646 transparent for PROM, registered for USART"

/CTEN = /(UARTADR*/CPU*ASD*/TC       "hold count upon TC for USART or"
        + PROMADR*/CPU*ASD*/Q3);     "Q3 for PROM access"
/CLKN = /(CLK*AS);                   "clock for F191"
/BERR = /(/CEN*/WUUB*/WUMB*/WLMB*/WLLB*/OE*/CPU*/SAB*ASD); "/BERR if no access"
```

Figure 7. PHD48N22–7 AMAZE Listing (end)

Programmable high-speed decoders (PHD)

```
@DEVICE TYPE
PHD16N8
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
 33MHz 68030/68882 /CS decode example
@PINLIST

"<--------FUNCTION-------->    <--REFERENCE-->"
"PINLABEL    PIN #  PIN_FCT    PIN_ID     OE_CTRL"
FC2           1       I          I0         -  ;
FC1           2       I          I1         -  ;
FC0           3       I          I2         -  ;
A19           4       PWR        I3         -  ;
A18           5       I          I4         -  ;
A17           6       I          I5         -  ;
A16           7       I          I6         -  ;
A15           8       I          I7         -  ;
A14           9       I          I8         -  ;
GND          10       OV         GND        -  ;
A13          11       I          I9         -  ;
/CS          12       /O         O0         EA  ;
N/C          13       /B         B1         D1  ;
N/C          14       /B         B2         D2  ;
N/C          15       /B         B3         D3  ;
N/C          16       /B         B4         D4  ;
N/C          17       /B         B5         D5  ;
N/C          18       /B         B6         D6  ;
N/C          19       /O         O7         EB  ;
VCC          20       +5V        VCC        -  ;

@COMMON PRODUCT TERM  "CPT_label = (expression)"

CPU         = FC2 * FC1 * FC0;         "cpu space"
COPROCESSOR =/A19 */A18 * A17 */A16;  "coprocessor communications"
CP_ID       =/A15 */A14 * A13;         "cp-id one"

@OUTPUT ENABLE  " EA, EB "

 EA = 1;

@I/O DIRECTION  " D0 .. Dn"
@LOGIC EQUATION

/CS = /(CPU * COPROCESSOR * CP_ID);
```

Figure 8. PHD16N8–5 AMAZE Listing

Minimize metastability in 50MHz state machines

Programmable Logic Devices

By Bob Kelly, Senior Field Applications Engineer, Philips Components–Signetics

Engineers are excited to discover the PLUS405–55, a PLD state machine IC rated for a maximum operating frequency of 55MHz. It has a flexible architecture offering 65 product terms, and a programmable OR array driving 16 S-R flip-flops, 8 of which are buried (see Figure 1.). This design allows the 64 product terms to realize 64 state transitions in a general state machine implementations. (State machines based on a counter will be implemented much more efficiently.) In order to estimate if a particular state machine will fit in the PLUS405, one need only count the state transitions and assure that there are fewer than 65! There are the remaining issues of number of states, inputs and outputs. The PLUS405 has 8 buried registers, allowing representation of 256 unique states. A dual complement array is available for the "ELSE" condition of state equations, and along with dual clocking capabilities allows two independent state machines to be synthesized on one IC.

Ease of design is further enhanced by AMAZE, the PC-based PLD development tool. AMAZE supports Boolean and State Equation entry of the design, simulation, and downloading of the programming information to a programmer. A more powerful deisgn tool that supports the PLUS405 and the Programmable Macro Logic Family is SNAP 1.4. SNAP allows an abstract approach to design with PLDs, as the target device is not specified by the engineer until he is done fully integrating and simulating his efforts. After device selection, SNAP can back-annotate the design files with target silicon characteristics, allowing simulation of the actual device. SNAP is old with or without a schematic capture package and AMAZE is distributed at no cost after a registration form has been filled out.

The Engineer sets out to solve all his high speed state machine design problems armed with this new silicon and software, only to discover all this performance has its price. Studying the data sheet on the PLUS405–55 shows the following performance:

f_{MAX1}	55.6MHz minimum
Input Setup time	10ns minimum
Input Hold time	0ns minimum

Minimize metatability in 50MHz state machines

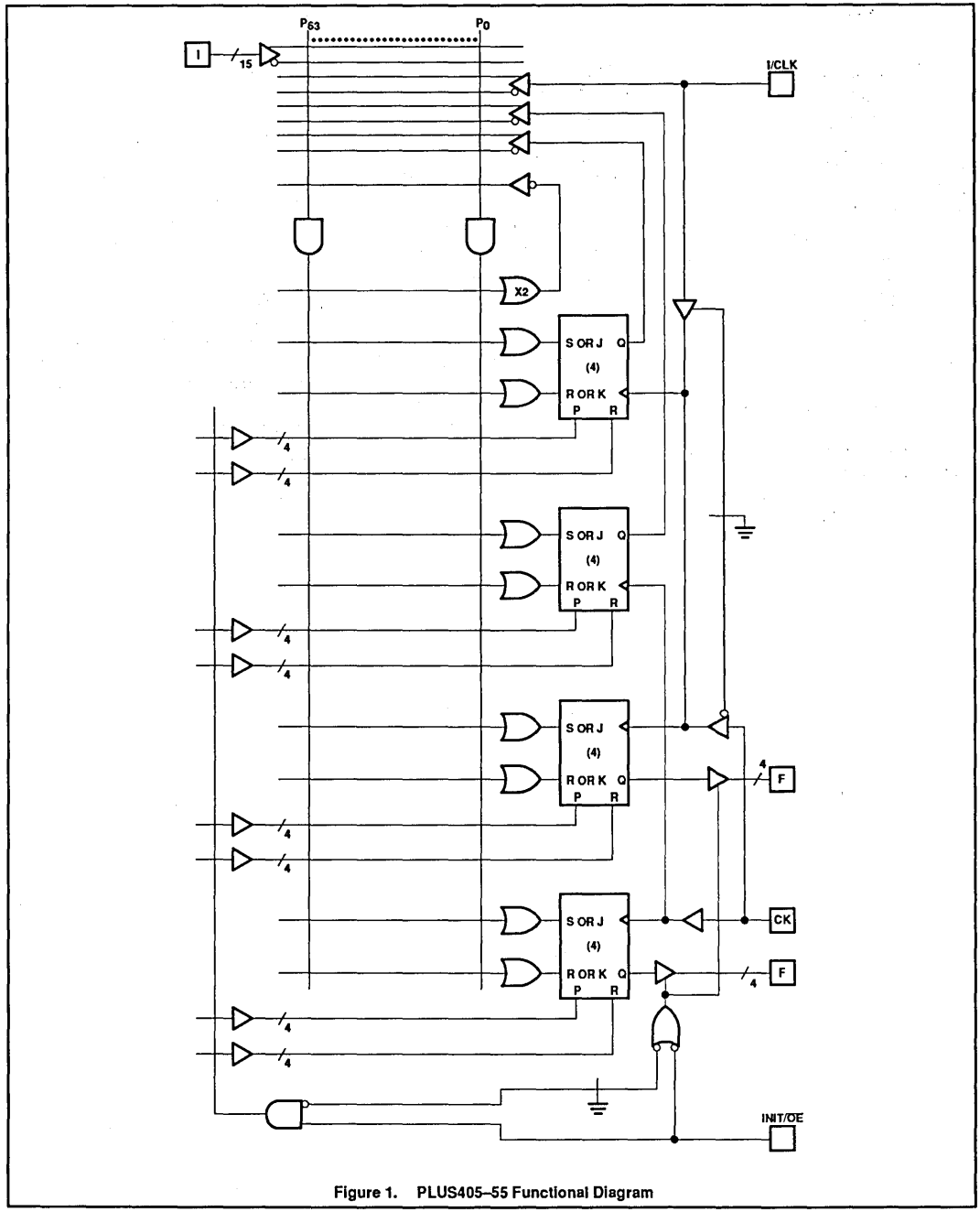


Figure 1. PLUS405–55 Functional Diagram

Minimize metatability in 50MHz state machines

The cycle time at 55MHz is roughly 18.2 nsec. The window during which data must be stable to guarantee no metastability is 10 nsec long. The difference between the setup and hold time, and the cycle time, is the allowed time interval for changes to occur. this example leaves 8.2ns for any changes.

From a system standpoint, this means the design engineer must be extremely careful in implementing his system, or he will violate the setup and hold specifications of the PLUS405–55. This can lead to metastable conditions in the state machine with several negative effects:

1. Jumps to undefined states. (May hang up system!)
2. Lengthened clock to Q times (slows down!).
3. Jumps to states out of proper sequence.

All of the above problems will yield a system that is unreliable, unpredictable and expensive in terms of servicing elusive bugs in the field.

The preceding analysis said nothing regarding asynchronicity. It is feasible to design the above system in a fully synchronous manner and have acceptable results. What about the system where known asynchronous inputs will be used in the state machine? Examples of common asynchronous signals are refresh request in DRAM controller applications and interrupts in a real-time control system. One approach to managing asynchronous inputs is to precede the state machine with a D-type flip-flop. This can serve as a synchronizing stage . . . or can it??? A simple analysis will explore the feasibility of using a simple synchronizing flip-flop.

A common Dual-D flip-flop frequently selected for this application in TTL high-performance systems is the FAST 74F74. The asynchronous data is fed into the D input of the flip-flop, and the Q output is fed into the logic input of the PLUS405 state machine. A common clock is used for both parts (see Figure 2.). Based on current published data sheets, the 74F74 has a clock-to-Q time of 9.2ns maximum. The worst case setup time on the PLUS405–55 is 10ns. The minimum cycle time of the combined system is (9.2 + 10) ns, yielding a maximum clock frequency of 52MHz. Let's assume for this example a desired system clock is 50MHz.

Minimize metatability in 50MHz state machines

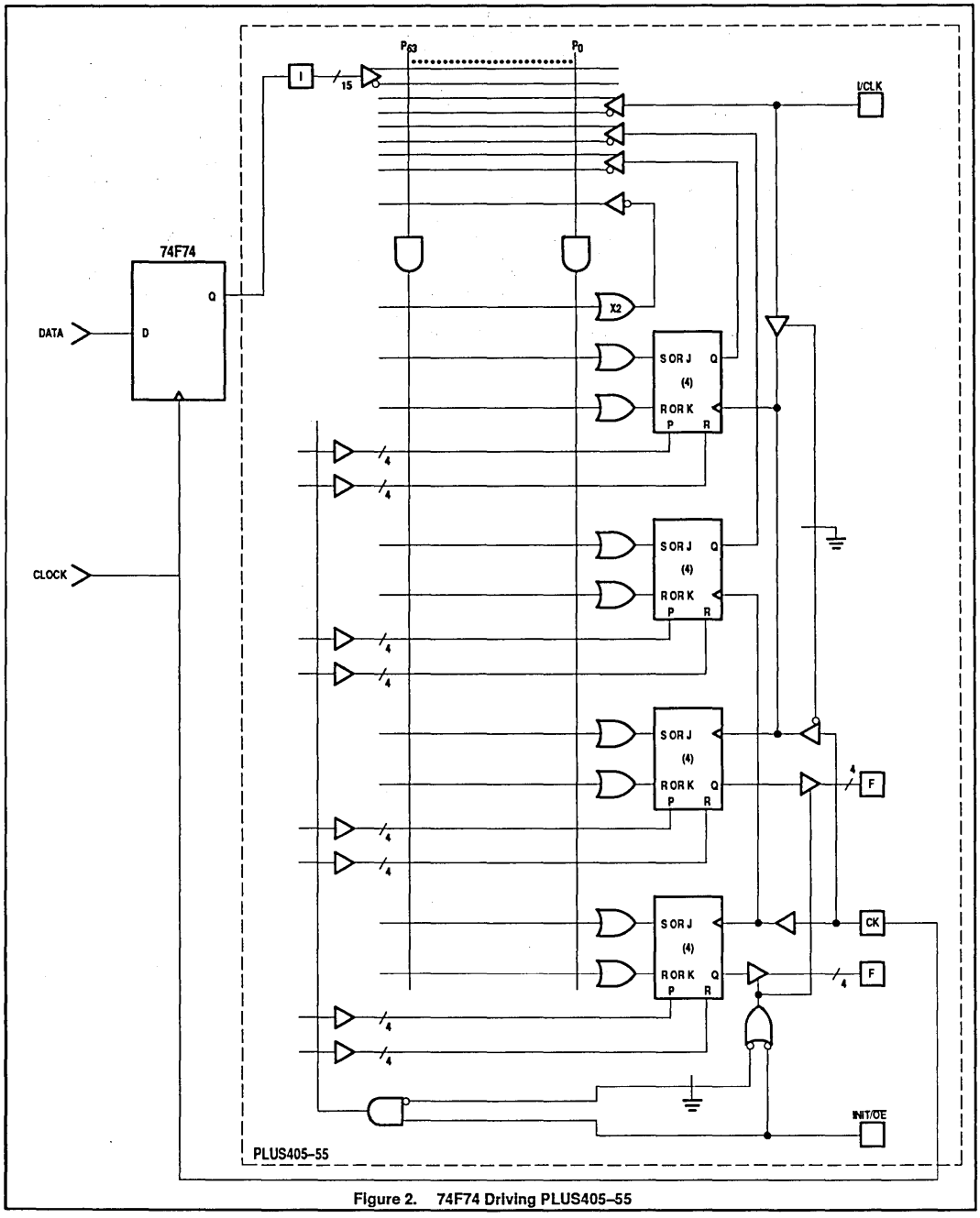


Figure 2. 74F74 Driving PLUS405–55

Minimize metatability in 50MHz state machines

The important issue to examine in the timing diagram is the time that elapses between the end of the 74F74 clock-to-Q interval and the beginning of the PLUS405's setup time (see Figure 3.). This is 20ns minus 10ns minus 9.2ns, which equals .8ns! At 50MHz, this combination is just able to work reliably on a worst case basis, assuming no instances of metastability. If metastable operation is encountered, the 800 picosecond window is the only time left in the clock cycle to resolve the situation. The next issue to examine is the mean time between failures (MTBF) for this system. From the work of Mr. Chaney, an equation which models metastable behavior is:

EQUATION 1. MTBF = exp($\tau r/\tau$) / (T0 * f * a)

(Explanation of above symbols)

MTBF is mean time between failures, in seconds.

τr is the elapsed time before sampling the process

or

the time allotted for metastability to resolve.

τ is the "Metastability Time Constant".

T0 in seconds, the zero intercept of aperture time versus propagation delay. T0 indicates the propensity of a device to enter the metastable state.

f is the clock speed, in Hertz.

a is the transition rate of data being sampled (i.e., edges per second) in Hertz.

Assume for this discussion that the asynchronous input data is roughly 2MHz, meaning the edges that can cause metastability occur at a 4MHz rate. The system clock is assumed to be 50MHz, and the elapsed time before sampling is 10ns. (The sample time is calculated from the difference between the cycle time (1/50MHz = 20ns) and the setup time of the PLUS405–55 (10ns). The other parameters can be determined from measurements of an 'F74, or can be found in Mr. Chaney's paper. τ was found to be .4ns and T0 .2 milliseconds. Armed with a calculator and Equation #1, the MTBF for this particular scenario is calculated:

MTBF = exp(10/.4) / (.2e–3* 50e6 * 4e6) =
1.8 seconds

Clearly, this level of failure in any system is unacceptable. A better solution for this class of problem must be found!

Signetics has recently introduced a new family of parts designed with metastability performance in mind. The first four members of this family are the 74F5074, 74F50109, 74F50728 and 74F50729. These are excellent general purpose flip-flops, but special attention has been paid to short setup and hold times, and fast clock-to-Q times. The output stage has also been designed with a balanced drive characteristic, leading to tight matching between rise and fall propagation delays, and matching of skews between other outputs. this makes them useful in clock driver applications also. Let's repeat the former calculation using the measured τ and T0 values for the 74F5074 used as a synchronizer (see Figure 4.) ahead of the PLUS405–55.

τ = .135ns
T0 = 9.8 E 6 seconds

MTBF = exp(10/.135) / (9.8e6 * 50e6 * 4e6) =
75.46 e9 seconds

NOTE: For the reader's reference, a century is 3.154 e9 seconds.

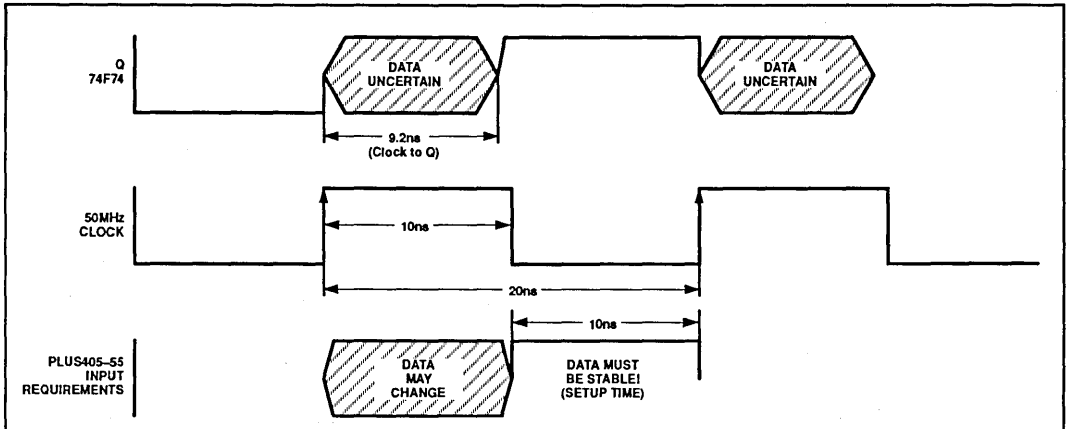


Figure 3. Timing Diagram of 74F74 and PLUS405–55

Minimize metatability in 50MHz state machines

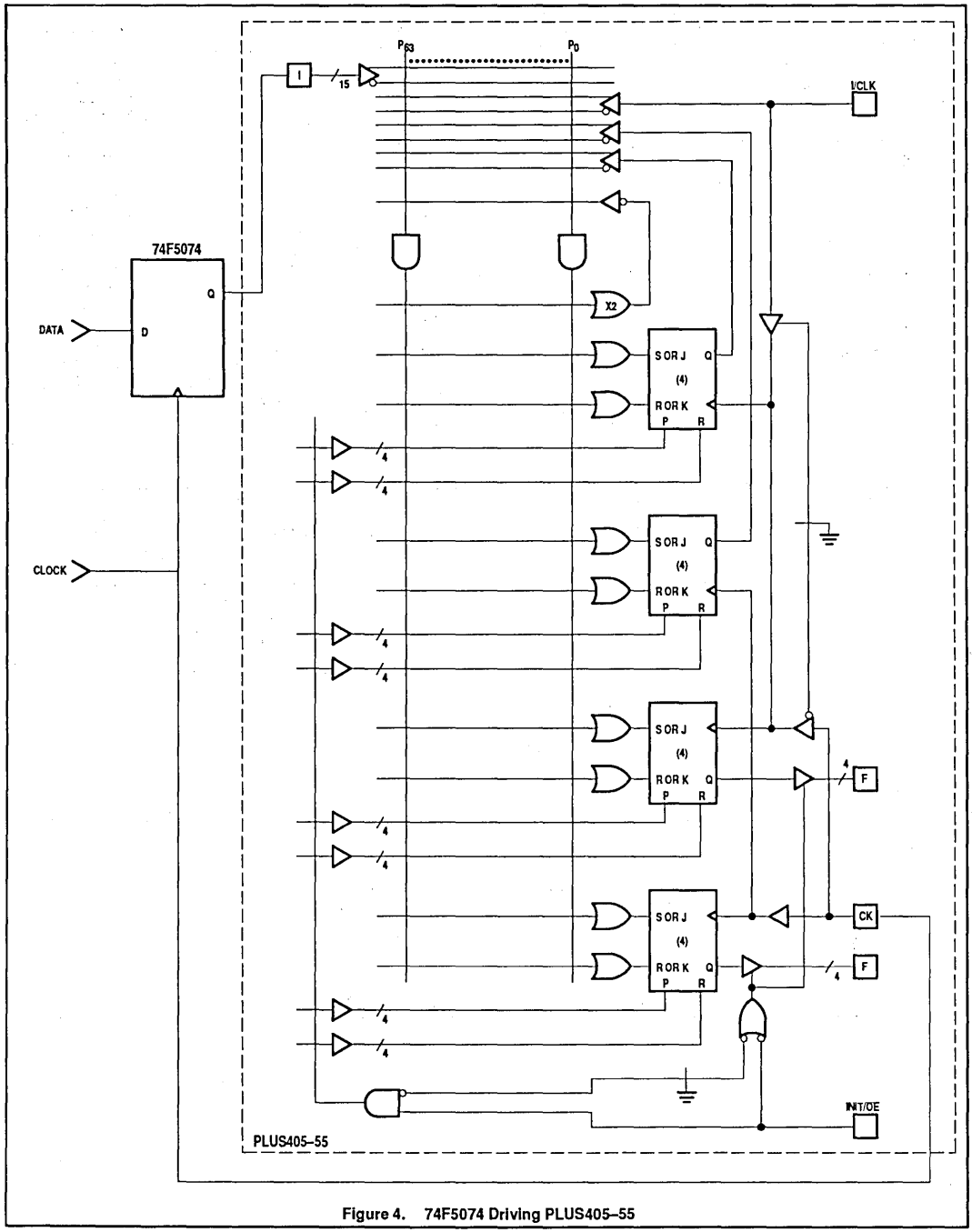


Figure 4. 74F5074 Driving PLUS405–55

Minimize metatability in 50MHz state machines

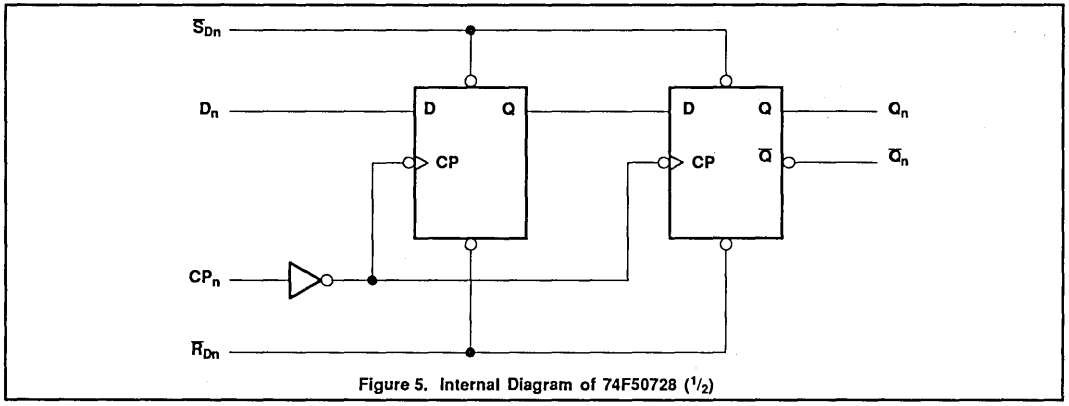


Figure 5. Internal Diagram of 74F50728 ($^1/_2$)

A system that was unreliable is now found to be quite acceptable by using the 74F5074. The major drawback to the synchronizing flip-flop solution is the added delay on the asynchronous signal before it enters the state machine. In the case of the 74F5074, this will amount to one clock cycle delay. For designs that demand the maximum in freedom from metastability, Signetics has developed a product with cascaded D flip-flops for synchronizing applications. The 74F50728 (see Figure 5.) will therefore introduce a two clock cycle delay into the system. It is pin compatible with the 74F5074 and 74F74 to allow retrofits on existing systems.

Calculation for the MTBF of a system using the 74F50728 is similar to the technique used earlier. In this case though, at least one entire clock cycle is used to resolve any metastability.

EQUATION 2. MTBF = $\exp(\tau r/\tau) / (T0 * f * a)$

{Explanation of above symbols}

All symbols are the same as EQUATION 1 with the exception of τr.

τr is the elapsed time before sampling the process or the time allotted for metastability to resolve. In the case of 74F50728, one entire clock cycle.

The flip-flops embedded in the 'F50728 are essentially the same as the flip-flops used for the 'F5074, therefore the same "Metastability Time Constant" τ, and T0, can be used in the calculation.

MTBF = exp(20/.135)/ (9.8e6 * 50e6 * 4e6) =
1.12 e43 seconds!

Now that the designer is comfortable with handling metastability, it is feasible to begin approaching the design of the system by stating a goal for MTBF and adjusting the state machine's clock to meet the desired failure level.

Let's assume out system is to have an MTBF of 5 years from metastability induced anomalies. The calculations would proceed as follows, assuming the same 2MHz data rate from our previous example:

MTBF = 5 years * (31.54 e6 seconds/year) =
157.7 e6 seconds

Setting up the equation to find the roots yields:

EQUATION 3. $T(setup)/\tau - 1/(f * \tau) +$
$1n(T0*a*MTBF*f) = 0$

(T(setup) is the setup time on the PLUS405)

Minimize metatability in 50MHz state machines

Equation 3 is not solved using algebra, but simple numerical methods will allow easy solution, especially since we have a good initial guess for the value of f. (50 to 55MHz!) An HP 32S calculator was used to find the root of this equation by the following program:

				COMMENTS
	PRGM			start program entry
	GTO . .			go to top of memory
B01	LBL/RTN {LBL}		B	label program as B
B02	INPUT	A		a, Data rate, Edges/Sec.
B03	INPUT	F		Clock frequency, Hertz
B04	INPUT	J		τ, seconds
B05	INPUT	M		MTBF, seconds
B06	INPUT	T		T0, seconds
B07	INPUT	U		T(setup), seconds
B08	RCL	A		begin calculation of 1n argument
B09	RCL x	T		
B10	RCL x	M		
B11	RCL x	F		
B12	LN			
B13	RCL	F		
B14	RCL x	J		
B15	1/x			
B16	+/–			change sign
B17	+			add
B18	RCL	U		
B19	RCL +	J		divide
B20	+			
B21	LBL/RTN {RTN}			end, return from routine

Minimize metatability in 50MHz state machines

To execute this program we must use the SOLVE capability on the calculator.

SOLVE {FN}	FN=		Prompt for label of function
B			
50 E 6	STO	F	load initial guess 50MHz
SOLVE {SOLVE}	SOLVE		prompt for unknown variable
F			frequency in this case!
A?	4.0 E 6		set edge rate
R/S			run
J?	135 E –12		set τ
R/S			
M?	157.7 E 6		set MTBF
R/S			
T?	9.8 E 6		set T0
R/S			
U?	10. E –9		setup time
R/S			

At this point the calculator will set off to find the root based on the initial guess and the desired conditions entered. The system clock speed determined from this technique is 52.16MHz.

Designers who are forced to deal with an uncertain system for the first time are uncomfortable with the idea that is is possible for the system to fail. Lower speed systems have been traditionally designed using worst case data sheet numbers to guarantee that the system will always work. As system clock speeds cross over 50MHz, meeting the setup and hold times becomes very difficult for TTL-based designs. The allowed time to resolve metastability gets shorter and the data stream edges become much more frequent, increasing the incidence of metastability. Faster systems demand that a design methodology based on statistics be used and the burden is now on the Engineer to manage likelihood of failure to acceptable levels. Persons defining high performance systems will need to specify goals for MTBF due to metastability. Usage of parts that have been characterized for metastability behavior will become mandatory in future systems. New parts, such as the 74F5074 and 74F50728 from Signetics, which have published metastable traits and are pin compatible with other industry standard ICs, can make solving these problems as easy as plugging in a new part!

Dike, Charles, *"AN219, A Metastability Primer"*, Signetics

Chaney, Thomas J., *"Measured Flip-Flop Responses to Marginal Triggering"* [*IEEE Transactions on Computing*, Vol. C-32, No. 12, December 1983, pp. 1207-1209]

Wakerly, John, *"Designers Guide to Synchronizers and Metastability, Part 1 and 2"* September 1987, *VLSI Design*

Philips Components–Signetics

Application Note	
Date of Issue	June 1988
Rev. Date	
Programmable Logic Devices	

AN7
Single chip multiprocessor arbiter

INTRODUCTION

In multiprocessor environments there is considerable savings to be made through sharing system resources. If each processor must support its own bus structure, I/O devices, and bulk storage medium, system cost could be very high. In the configuration shown in Figure 1, all processors share a common communication bus, and a number of system resources.

Since every processor must use the common system bus to communicate with its peripherals, a priority structure that resolves simultaneous processor bus requests into a single bus grant must be integrated into the system. In addition to making request-grant transactions, transient bus contention due to grant switching must be removed by inserting precise guard band times between bus grants.

Philips Components–Signetics' Programmable Logic Sequencer provides a convenient and cost effective means for implementing a synchronous arbiter to perform these tasks within a single chip.

ARBITER STRUCTURE

Within a multiprocessor system, two general classes of processors can be recognized: Priority A and Priority b. Priority A processors have the highest request priority and must only compete with other Priority A processors for bus control. The arbiter must issue "A" grants in a manner that prevents any high priority "A" processor from locking out another Priority A processor. To enable this, the Priority A rules implemented here use a Last Granted Lowest Priority (LGLP) ring structure. After an "A" processor has completed a bus-related task, its next arbitrated request priority will be lowest in the "A" request group. The previously second highest priority "A" processor will then become highest priority requester. The net effect of the "round robin" exchange is that every Priority A processor will have a turn at being highest priority processor. Priority A processors are typically ones that perform real-time operations or vital system tasks.

Priority b processors are lower in priority than the "A's" and may only be granted system control when no "A" requests are pending. "b" processors usually perform background tasks. Within the Priority b group, further priority ordering exists such that each "b" processor has a fixed priority position.

Plumber[1], Pearce[2], and Hojberg[3] present asynchronous techniques of arbiter implementation. These methods all have hard-wired priority rules and imprecise guard band times during grant switching. As pointed out by Hojberg, a synchronous state machine can be configured as a Mealy-type controller to provide not only precise guard band times and programmable priority rules, but also programmable input/output polarity. The state machine in Figure 2 is made from a control PROM array and an edge-triggered latch. The "A" and "b" requests and the machine's present state are used by the control PROM to determine the next "A" and "b" grants and the next state.

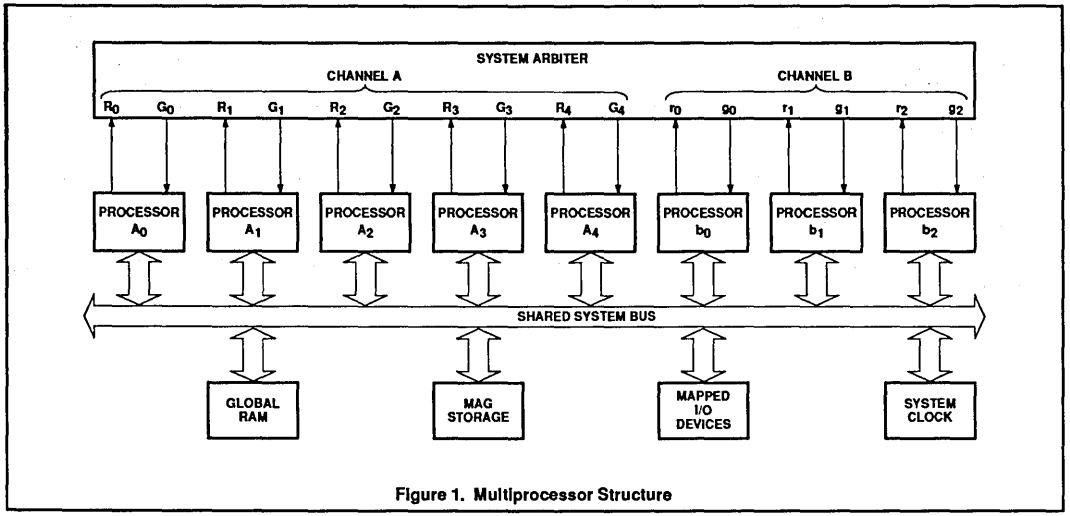
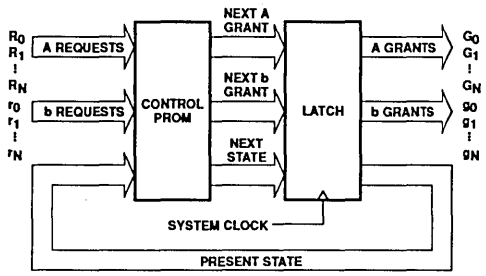


Figure 1. Multiprocessor Structure

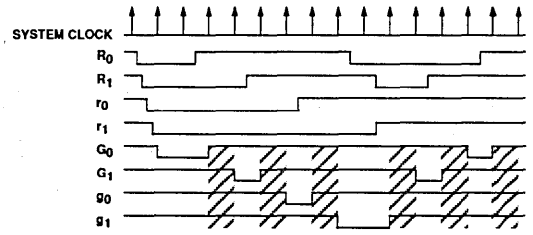
Single chip multiprocessor arbiter AN7

SYSTEM OPERATION

Two machine states can be identified by inspection: a wait state and a grant state. The state machine enters a grant state as a response to a system request on either R_N or r_N. The machine will remain in this state with a single grant line asserted as long as the request remains asserted. Upon releasing the request line, the machine will pass through a single wait state before considering other pending requests. This provides a single state guard band time. The requests received must meet the set-up requirement of the edge-triggered latch after propagating through the control PROM. If these time considerations do not fit within a given multiprocessor structure, an input latch may be added such that the R_N and r_N lines are clocked through the latch by the system clock, thereby removing asynchronous set-up time considerations. On the basis of a state machine approach, two techniques of implementation are feasible: 1) using an architecturally advanced single IC controller, the PLS, and, 2) a traditional PROM/LATCH configuration.



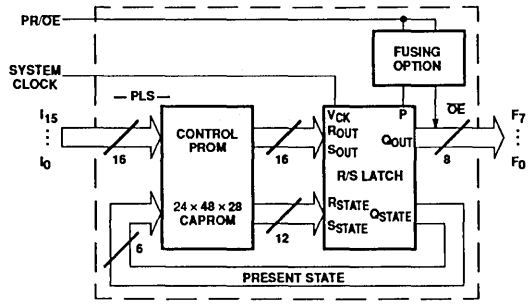
a. A and b service requests (R_N, r_N) plus the present state determine, through the control PROM, the next state and the next grant outputs (G_N, g_N).



b. Requests R_0, R_1, r_0, and r_1 are asserted low in the same clock sample period. The priority rules that determine the order in which the grants are issued and the shaded guard-band areas are programmed into the control PROM. Note that the A and b request lines and the present state input to the PROM must have a set-up time equal or greater than the latch set-up time plus the PROM access time.

Figure 2. Arbiter Constructed from a Mealy-Type State Machine

Single chip multiprocessor arbiter AN7



a. PLS Single Chip System

		TRANSITION TERM																OPTION (P/E)													
																		OUTPUT TERM													
NO.	C	INPUT VARIABLE (Im)														PRESENT STATE (Ps)						NEXT STATE (Ns)						OUTPUT FUNCTION (Fn)						
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0																															
1																															
2																															
3																															
4																															
5																															
46																															
47																															

b. PLS Standard Program Table

CK	S	R	Q
↑	0	0	Q
↑	0	1	0
↑	1	0	1
↑	1	1	Undefined

NOTE:
(↑) denotes positive clock transition.
S = R = 1 is an illegal input condition.

c. Transition Table of Clocked S/R Flip-Flop

Figure 3

PLS ARBITER IMPLEMENTATION

A five Priority "A" and three Priority "b" arbiter will be constructed such that all grant outputs will be asserted low for grants and all request inputs will be asserted low for system requests.

Brief PLS Description

The PLS block diagram shown in Figure 3(a) consists of a control PLA and 14 clocked S/R flip-flops. The control PLA is actually an AND-OR logic array that functions as a Content Addressable PROM. The PLA is organized as 48 words of 28 bits with 16 external input lines, and six internal inputs fed back from the State Register. The 28 PLA outputs drive the S/R inputs of the six-bit State Register and eight-bit Output Register. Note that the state feedback path is made inside the PLS.

I_N and present state inputs, P_S, represent 2^{22} possible input codes; 48 of these codes may be mapped in the PLA to provide a 14 bit register control word. As shown in Figure 3(b) each input code may be specified by assigning to the variables either Low "L", High "H", or Don't Care "–" logic states. If any input code falls logically outside the programmed codes, the PLA asserts a Low on all its 28 internal outputs, thereby issuing a "no change" command to the R/S flip-flops.

This is an important architectural feature because it requires that only state or output transition terms be programmed. Looping terms that change neither state nor output need not be programmed in the PLS, owing to the functional characteristics of S/R flip-flops tabulated in Figure 3(c). An example of this is shown in Figure 4.

The S/R inputs of both state and output registers are specified by using PLA outputs ("AND" functions of request inputs and present state) in the program table of Figure 3(c).

Single chip multiprocessor arbiter AN7

The corresponding next state of each bit will be set to 0 for "L", 1 for "H", and No Change for "–". The PLS's PR/OE line may be assigned either Asynchronous Preset or Output Enable functions, via a user programmable option.

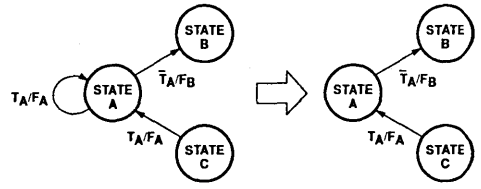
The entire function is integrated into a single 28-pin package designated as PLUS105.

State Algorithm

Figure 5(a) displays the circular state form and all possible state transitions of the LGLP priority structure. Hex states 3F, 3E, 3D, 3C, and 3B are arbiter wait states W_{0-4}. In these states, processor "A" and "b" requests are monitored. Figure 5(b) illustrates a typical grant to processor A_1 in hex state 07. As long as A_1 asserts its request line low, the next state will be 07_{16} and the next output will remain with G_1 asserted low and all the other grant outputs asserted high. Since no change in state or grant output results from this transition, no PLA resources are required.

As soon as processor A_1 returns its request line, R_1, to 1, a state transition is made to 3D, and an output transition is made to set all grant outputs to 1. Since processor A_1 was the last to be granted system resources, it will now have the lowest A level request priority (LGLP). In wait state W_2, the highest priority processor will be A_2, second A_3, third A_4, and fourth A_0. To maintain the LGLP rule, grant transitions must follow the state rule $G_N \rightarrow W_{(N+1)}$, and wait states, W_M, must set their "A" priorities so that processor A_M is highest priority. Priority decreases as one proceeds clockwise around the state ring to the lowest priority processor, $A_{(m-1)}$.

When no "A" requests are pending, "b" requests may be granted. To avoid upsetting the LGLP priority rule, a "b" grant must leave and return to the same wait state. Since the "b" priority structure is the same regardless of the wait state, only a single set of "b" transition terms are required.



NOTE:
T_N/F_N = Inputs/outputs.
The PLS requires that only two out of the three transition terms be programmed.

Figure 4. PLS State and Output Transitions

For example, a grant transition to g_2 (Hex 20-25) can be issued only if there are no "A", "b_0", or "b_1" requests pending. Given the binary wait state code 111XXX, where "X's" represent Don't Cares, a request code of 01111111 will transfer the arbiter to the grant state g_2 from any of the wait states, W_{0-4}.

It is important to realize that in making this transition, the lower 3-State bits will not be changed—they provide the wait state return address. When r_2 returns high, 1XXXXXXX, a transition back to the previously exited wait state is made by forcing a "1" in the three most significant state bits and leaving the lower 3-State bits unchanged.

All output and state bits are initially preset to "1" through the use of the optional preset function. Grant output lines are only forced low when transitions are made to grant states and are returned to "1" when jumping back to a wait state.

Table 1 provides the complete arbiter program. The complete arbiter circuit diagram is shown in Figure 6. The AMAZE equations are shown in Figure 7.

PROM/LATCH IMPLEMENTATION

The same five "A" processor and three "b" processor arbiter can be implemented with discrete PROM's and Latches using the same state diagrams for the PLS, except that now looping transition terms must be programmed. Coding of all state and output transitions requires programming of two memory fields: the "A" request PROM's (2KX6) and the "b" request PROM (64×3). The complete circuit diagram is shown in Figure 6(b).

The "A" request PROM's determine the next machine state (N_{0-5}) at all times, except when there are no "A" requests pending and there is a "b" request, or if the machine is presently in a "b" grant state. In these cases, the "b" request PROM controls the machine's next state.

The grant control lines are decoded from the next state lines and latched in two quad output latches. This PROM/LATCH organization is conceptually the same as that shown in Figure 2.

Single chip multiprocessor arbiter AN7

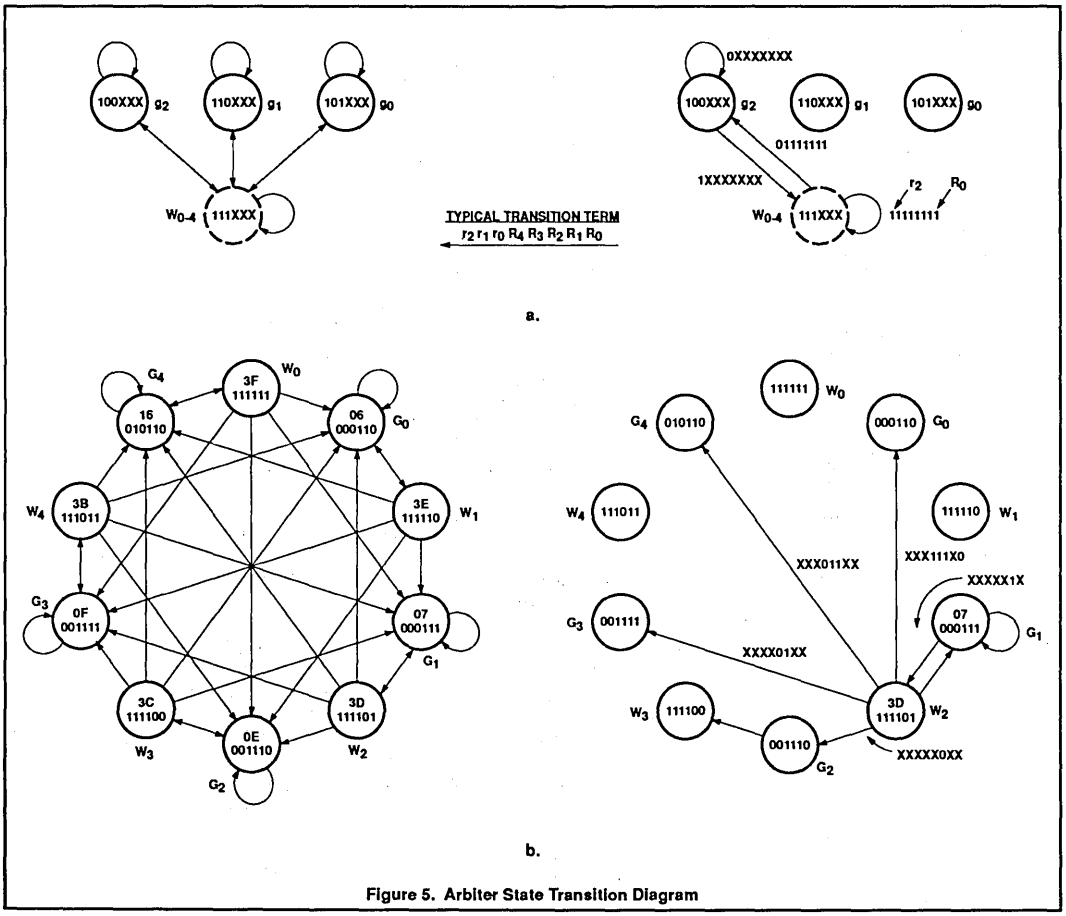


Figure 5. Arbiter State Transition Diagram

Single chip multiprocessor arbiter AN7

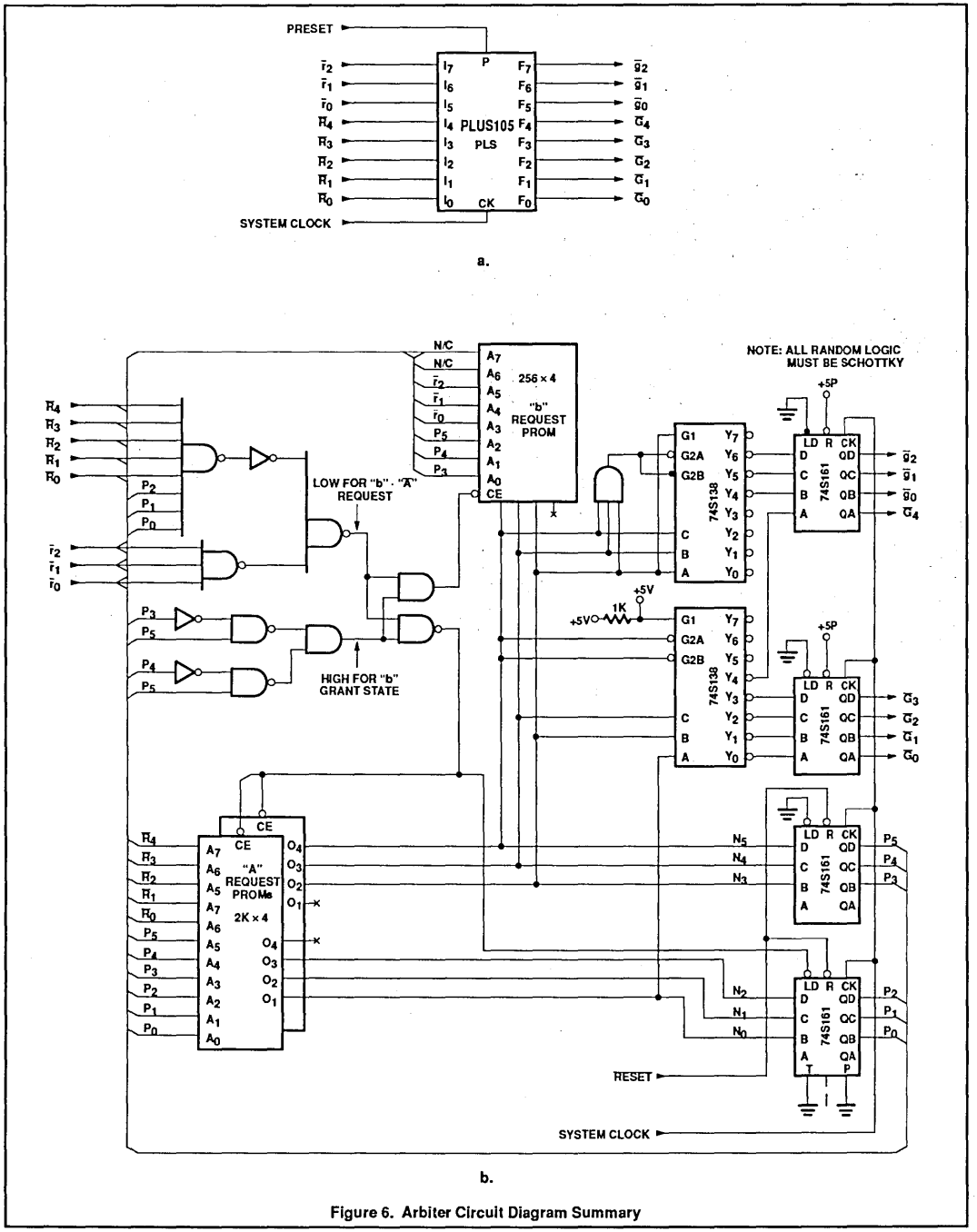


Figure 6. Arbiter Circuit Diagram Summary

Single chip multiprocessor arbiter AN7

Table 1. PLS Program Table for Priority Arbiter

NO.	TRANSITION TERM: INPUT VARIABLE (Im) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 5	4	3	2	1	0	OUTPUT TERM: NEXT STATE (Ns) 5	4	3	2	1	0	OUTPUT FUNCTION (Fn) 7	6	5	4	3	2	1	0	
0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	L	L	L	H	H	L	H	H	H	H	H	H	H	L	
1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H	
2	—	—	—	—	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	H	L	L	H	H	H	L	H	H	H	H	H	L	H	H	W_0
3	—	—	—	—	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	H	L	L	H	H	H	H	H	H	H	H	H	L	H	H	
4	—	—	—	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	H	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	
5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L	—	H	H	H	H	H	L	L	L	L	H	H	H	H	H	H	H	H	L	H	H	
6	—	—	—	—	—	—	—	—	—	—	—	—	—	L	H	—	H	H	H	H	H	L	L	L	H	H	H	L	H	H	H	H	H	H	H	L	
7	—	—	—	—	—	—	—	—	—	—	—	—	L	H	H	—	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	W_1
8	—	—	—	—	—	—	—	—	—	—	—	L	H	H	H	—	H	H	H	H	H	L	L	H	L	H	H	L	H	H	H	H	L	H	H	H	
9	—	—	—	—	—	—	—	—	—	—	—	H	H	H	H	L	H	H	H	H	H	L	L	L	H	H	H	L	H	H	H	H	H	H	H	L	
10	—	—	—	—	—	—	—	—	—	—	—	—	—	L	—	—	H	H	H	H	L	H	L	L	H	H	H	L	H	H	H	H	L	H	H	H	
11	—	—	—	—	—	—	—	—	—	—	—	—	L	H	—	—	H	H	H	H	L	H	L	L	H	H	H	H	H	H	H	H	L	H	H	H	
12	—	—	—	—	—	—	—	—	—	—	—	L	H	H	—	—	H	H	H	H	L	H	L	H	L	H	H	L	H	H	H	H	L	H	H	H	
13	—	—	—	—	—	—	—	—	—	—	—	H	H	H	—	L	H	H	H	H	L	H	L	L	L	H	H	L	H	H	H	H	H	H	H	L	
14	—	—	—	—	—	—	—	—	—	—	—	H	H	H	L	H	H	H	H	H	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	
15	—	—	—	—	—	—	—	—	—	—	—	—	L	—	—	—	H	H	H	H	L	L	L	L	H	H	H	L	H	H	H	H	L	H	H	H	
16	—	—	—	—	—	—	—	—	—	—	—	L	H	—	—	—	H	H	H	H	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	
17	—	—	—	—	—	—	—	—	—	—	—	H	H	—	—	L	H	H	H	H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L	W_3
18	—	—	—	—	—	—	—	—	—	—	—	H	H	—	L	H	H	H	H	H	L	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	
19	—	—	—	—	—	—	—	—	—	—	—	H	H	L	H	H	H	H	H	H	L	L	L	L	H	H	H	L	H	H	H	H	H	H	H	H	
20	—	—	—	—	—	—	—	—	—	—	—	L	—	—	—	—	H	H	H	L	H	H	L	L	H	L	H	L	H	H	H	H	H	H	H	H	
21	—	—	—	—	—	—	—	—	—	—	—	H	—	—	—	L	H	H	H	L	H	H	L	L	L	H	H	H	H	H	H	H	L	H	L	H	
22	—	—	—	—	—	—	—	—	—	—	—	H	—	—	L	H	H	H	H	L	H	H	L	L	L	L	H	H	H	H	H	H	H	L	H	H	W_4
23	—	—	—	—	—	—	—	—	—	—	—	H	—	L	H	H	H	H	H	L	H	H	L	L	H	H	H	H	H	H	H	H	L	H	H	H	
24	—	—	—	—	—	—	—	—	—	—	—	H	L	H	H	H	H	H	H	L	H	H	L	L	H	H	H	H	H	H	H	H	L	H	H	H	
25	—	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	H	—	—	—	H	H	L	H	H	H	H	H	H	H	—	—	—		
26	—	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	H	H	—	—	—	H	L	H	H	H	H	H	H	H	H	—	—	—		W_{0-4}
27	—	—	—	—	—	—	—	—	L	H	H	H	H	H	H	H	H	H	H	—	—	—	L	H	H	H	H	H	H	H	H	H	—	—	—		
28	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	G_0
29	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H	—	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	G_1
30	—	—	—	—	—	—	—	—	—	—	—	—	—	H	—	—	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	G_2
31	—	—	—	—	—	—	—	—	—	—	—	—	H	—	—	—	L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	G_3
32	—	—	—	—	—	—	—	—	—	—	—	H	—	—	—	—	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	G_4
33	—	—	—	—	—	—	—	—	—	—	H	—	—	—	—	—	H	L	H	—	—	—	H	H	H	H	H	H	H	H	H	H	H	H	H	H	g_0
34	—	—	—	—	—	—	—	—	—	H	—	—	—	—	—	—	H	H	L	—	—	—	H	H	H	H	H	H	H	H	H	H	H	H	H	H	g_1
35	—	—	—	—	—	—	—	—	H	—	—	—	—	—	—	—	H	L	L	—	—	—	H	H	H	H	H	H	H	H	H	H	H	H	H	H	g_2
36																																					
									$\bar{r}_2$	$\bar{r}_1$	$\bar{r}_0$	R_4	R_3	R_2	R_1	R_0													$\bar{g}_2$	$\bar{g}_1$	$\bar{g}_0$	G_4	G_3	G_2	G_1	G_0			

Single chip multiprocessor arbiter AN7

```
 ******************** ARBITERS ********************
@DEVICE SELECTION
ARBITERB/PLUS105
@STATE VECTORS
[ FF5, FF4, FF3, FF2, FF1, FF0 ]
W0  = 03Fh ;
W1  = 03Eh ;
W2  = 03Dh ;
W3  = 03Ch ;
W4  = 03Bh ;
W04 = 111---b ;
GA0 = 06h ;
GA1 = 07h ;
GA2 = 0Eh ;
GA3 = 0Fh ;
GA4 = 16h ;
GB0 = 101---b ;
GB1 = 110---b ;
GB2 = 100---b ;

@INPUT VECTORS
@OUTPUT VECTORS
[OB2, OB1, OB0, OA4, OA3, OA2, OA1, OA0]
QA0' = FEh ;
QA1' = FDh ;
QA2' = FBh ;
QA3' = F7h ;
QA4' = EFh ;
QB0' = DFh ;
QB1' = BFh ;
QB2' = 7Fh ;
NOGRANT' = FFh ;

@TRANSITIONS
WHILE [W0]
  CASE
    [/RA0] :: [GA0] WITH [QA0']
    [/RA1 * RA0] :: [GA1] WITH [QA1']
    [/RA2 * RA1 * RA0] :: [GA2] WITH [QA2']
    [/RA3 * RA2 * RA1 * RA0] :: [GA3] WITH [QA3']
    [/RA4 * RA3 * RA2 * RA1 * RA0] :: [GA4] WITH [QA4']
  ENDCASE

WHILE [W1]
  CASE
    [/RA1] :: [GA1] WITH [QA1']
    [/RA2 * RA1] :: [GA2] WITH [QA2']
    [/RA3 * RA2 * RA1] :: [GA3] WITH [QA3']
    [/RA4 * RA3 * RA2 * RA1] :: [GA4] WITH [QA4']
    [/RA0 * RA4 * RA3 * RA2 * RA1] :: [GA0] WITH [QA0']
  ENCASE
```

a. Arbiter State Equations

Figure 7

```
WHILE [W2]
  CASE
    [/RA2] :: [GA2] WITH [QA2']
    [/RA3 * RA2] :: [GA3] WITH [QA3']
    [/RA4 * RA3 * RA2] :: [GA4] WITH [QA4']
    [/RA0 * RA4 * RA3 * RA2] :: [GA0] WITH [QA0']
    [/RA1 * RA0 * RA4 * RA3 * RA2] :: [GA1] WITH [QA1']
  ENDCASE

WHILE [W3]
  CASE
    [/RA3] :: [GA3] WITH [QA3']
    [/RA4 * RA3] :: [GA4] WITH [QA4']
    [/RA0 * RA4 * RA3] :: [GA0] WITH [QA0']
    [/RA1 * RA0 * RA4 * RA3] :: [GA1] WITH [QA1']
    [/RA2 * RA1 * RA0 * RA4 * RA3] :: [GA2] WITH [QA2']
  ENDCASE

WHILE [W4]
  CASE
    [/RA4] :: [GA4] WITH [QA4']
    [/RA0 * RA4] :: [GA0] WITH [QA0']
    [/RA1 * RA0 * RA4] :: [GA1] WITH [QA1']
    [/RA2 * RA1 * RA0 * RA4] :: [GA2] WITH [QA2']
    [/RA3 * RA2 * RA1 * RA1 * RA4] :: [GA3] WITH [QA3']
  ENDCASE

WHILE [W04]
  CASE
    [/RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB0] WITH [QB0']
    [/RB1 * RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB1] WITH [QB1']
    [/RB2 * RB1 * RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB2] WITH [QB2']
  ENDCASE

WHILE [GA0]
  IF [RA0] THEN [W1] WITH [NOGRANT']

WHILE [GA1]
  IF [RA1] THEN [W2] WITH [NOGRANT']

WHILE [GA2]
  IF [RA2] THEN [W3] WITH [NOGRANT']

WHILE [GA3]
  IF [RA3] THEN [W4] WITH [NOGRANT']

WHILE [GA4]
  IF [RA4] THEN [W0] WITH [NOGRANT']

WHILE [GB0]
  IF [RB0] THEN [GB1] WITH [NOGRANT']

WHILE [GB1]
  IF [RB1] THEN [GB2] WITH [NOGRANT']

WHILE [GB2]
  IF [RB2] THEN [GB0] WITH [NOGRANT']
```

a. Arbiter State Equations (Continued)

Figure 7 (Continued)

Single chip multiprocessor arbiter AN7

```
 ******************** ARBITERB ********************
####################### P I N   L I S T ######################

   LABEL     ** FNC **PIN --------- PIN** FNC **    LABEL
CLOCK        ** CK  **  1-|        |-28 ** +5V **VCC
RB2          ** I   **  2-|        |-27 ** I   **N/C
RB1          ** I   **  3-|   P    |-26 ** I   **N/C
RB0          ** I   **  4-|   L    |-25 ** I   **N/C
RA4          ** I   **  5-|   U    |-24 ** I   **N/C
RA3          ** I   **  6-|   S    |-23 ** I   **N/C
RA2          ** I   **  7-|   1    |-22 ** I   **N/C
RA1          ** I   **  8-|   0    |-21 ** I   **N/C
RA0          ** I   **  9-|   5    |-20 ** I   **N/C
OB2          ** O   ** 10-|        |-19 ** PR  **PRESET
OB1          ** O   ** 11-|        |-18 ** O   **OA0
OB0          ** O   ** 12-|        |-17 ** O   **OA1
OA4          ** O   ** 13-|        |-16 ** O   **OA2
GND          ** 0V  ** 14-|        |-15 ** O   **OA3
                          ---------
```

b. Arbiter Pin List

```
 ******************** ARBITERB ********************
@DEVICE TYPE
PLUS105
@DRAWING
************ MULTI-PROCESSOR BUS ARBITOR
@REVISION
************ ARBITERB REV. 0
@DATE
************ JULY 26, 1985
@SYMBOL
************ ARBITERB
@COMPANY
************ SIGNETICS
@NAME
************ DAVID K. WONG
@DESCRIPTION
@INTERNAL SR FLIP FLOP LABELS
FF0 FF1 FF2 FF3 FF4 FF5

@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@LOGIC EQUATION
```

c. Arbiter Boolean Equations

Figure 7 (Continued)

Single chip multiprocessor arbiter AN7

Table 2. Arbiter Program Table

```
******************** ARBITERB ********************
Cust/Project - ************ DAVID K. WONG
Date         - ************ JULY 26, 1985
Rev/I. D.    - ************ ARBITERB REV. 0

PLUS105

---                                                   ---------------------------
 T !                                                  !         OPTION P/E    !H!
 E !------------------------------------------------------------------------------
 R ! !             INPUT VARIABLE      !PRESENT ST ! NEXT ST    !    OUTPUT      !
 M !C!1 1 1 1 1 1 ----------------------------------------------------------------
___!_!5_4_3_2_1_0_9_8_7_6_5_4_3_2_1_0!5_4_3_2_1_0!5_4_3_2_1_0!7_6_5_4_3_2_1_0!
  0!A!- - - -,- - - -,- - - -,- - - L!H H,H H H H!L L,L H H L!H H H H,H H H L!
  1!A!- - - -,- - - -,- - - -,- - L H!H H,H H H H!L L,L H H H!H H H H,H H L H!
  2!A!- - - -,- - - -,- - - -,- L H H!H H,H H H H!L L,H H H L!H H H H,H L H H!
  3!A!- - - -,- - - -,- - - -,L H H H!H H,H H H H!L L,H H H H!H H H H,L H H H!
  4!A!- - - -,- - - -,- - - L,H H H H!H H,H H H H!L H,L H H L!H H H L,H H H H!
  5!A!- - - -,- - - -,- - - -,- - L -!H H,H H H L!L L,L H H H!H H H H,H H L H!
  6!A!- - - -,- - - -,- - - -,- L H -!H H,H H H L!L L,H H H L!H H H H,H L H H!
  7!A!- - - -,- - - -,- - - -,L H H -!H H,H H H L!L L,H H H H!H H H H,L H H H!
  8!A!- - - -,- - - -,- - - L,H H H -!H H,H H H L!L H,L H H L!H H H L,H H H H!
  9!A!- - - -,- - - -,- - - H,H H H L!H H,H H H L!L L,L H H L!H H H H,H H H L!
 10!A!- - - -,- - - -,- - - -,- L - -!H H,H H L H!L L,H H H L!H H H H,H L H H!
 11!A!- - - -,- - - -,- - - -,L H - -!H H,H H L H!L L,H H H H!H H H H,L H H H!
 12!A!- - - -,- - - -,- - - L,H H - -!H H,H H L H!L H,L H H L!H H H L,H H H H!
 13!A!- - - -,- - - -,- - - H,H H - L!H H,H H L H!L L,L H H L!H H H H,H H H L!
 14!A!- - - -,- - - -,- - - H,H H L H!H H,H H L H!L L,L H H H!H H H H,H H L H!
 15!A!- - - -,- - - -,- - - -,L - - -!H H,H H L L!L L,H H H H!H H H H,L H H H!
 16!A!- - - -,- - - -,- - - L,H - - -!H H,H H L L!L H,L H H L!H H H L,H H H H!
 17!A!- - - -,- - - -,- - - H,H - - L!H H,H H L L!L L,L H H L!H H H H,H H H L!
 18!A!- - - -,- - - -,- - - H,H - L H!H H,H H L L!L L,L H H H!H H H H,H H L H!
 19!A!- - - -,- - - -,- - - H,H L H H!H H,H H L L!L L,H H H L!H H H H,H L H H!
 20!A!- - - -,- - - -,- - - L,- - - -!H H,H L H H!L H,L H H L!H H H L,H H H H!
 21!A!- - - -,- - - -,- - - H,- - - L!H H,H L H H!L L,L H H L!H H H H,H H H L!
 22!A!- - - -,- - - -,- - - H,- - L H!H H,H L H H!L L,L H H H!H H H H,H H L H!
 23!A!- - - -,- - - -,- - - H,- L H H!H H,H L H H!L L,H H H L!H H H H,H L H H!
 24!A!- - - -,- - - -,- - - H,L H H -!H H,H L H H!L L,H H H H!H H H H,L H H H!
 25!A!- - - -,- - - -,- - L H,H H H H!H H,H - - -!H L,H - - -!H H L H,H H H H!
 26!A!- - - -,- - - -,- L H H,H H H H!H H,H - - -!H H,L - - -!H L H H,H H H H!
 27!A!- - - -,- - - -,L H H H,H H H H!H H,H - - -!H L,L - - -!L H H H,H H H H!
 28!A!- - - -,- - - -,- - - -,- - - H!L L,L H H L!H H,H H H L!H H H H,H H H H!
 29!A!- - - -,- - - -,- - - -,- - H -!L L,L H H H!H H,H H L H!H H H H,H H H H!
 30!A!- - - -,- - - -,- - - -,- H - -!L L,H H H L!H H,H H L L!H H H H,H H H H!
 31!A!- - - -,- - - -,- - - -,H - - -!L L,H H H H!H H,H L H H!H H H H,H H H H!
 32!A!- - - -,- - - -,- - - H,- - - -!L H,L H H L!H H,H H H H!H H H H,H H H H!
 33!A!- - - -,- - - -,- - H -,- - - -!H L,H - - -!H H,L - - -!H H H H,H H H H!
 34!A!- - - -,- - - -,- H - -,- - - -!H H,L - - -!H L,L - - -!H H H H,H H H H!
 35!A!- - - -,- - - -,H - - -,- - - -!H L,L - - -!H L,H - - -!H H H H,H H H H!
 36!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 37!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 38!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 39!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 40!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 41!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 42!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 43!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 44!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 45!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 46!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!
 47!0!0 0 0 0,0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0!0 0,0 0 0 0!0 0 0 0,0 0 0 0!

       N N N N N N N N R R R R R R R R F F F F F F F F F F F F O O O O O O O O
       / / / / / / / / B B A A A A A A F F F F F F F F F F F F B B B A A A A A
       C C C C C C C C 2 1 0 4 3 2 1 0 5 4 3 2 1 0 5 4 3 2 1 0 2 1 0 4 3 2 1 0
```

Single chip multiprocessor arbiter AN7

Table 3. Design Alternatives for the Priority Arbiter

PARAMETER	SEQUENCER	PROM/LATCH
Parts count	1 IC	≈19IC's
PCB space	.84 in^2	7.92 in^2
Power	.65W	2.85W
Voltage	+5V	+5V

SUMMARY

As can be seen from the circuit diagrams, the PLS can offer significant advantages over discrete MSI arrays in the design of state machines. The tradeoff in both design alternatives for the Priority Arbiter is shown in Table 2. Clearly, the PLS approach uses fewer parts, with savings in PC board space and power requirements.

REFERENCES

1. W.W. Plumber: "Asynchronous Arbiters"; IEEE Transactions on Computers, January 1972, pp. 37–42.
2. R.C. Pearce, J.A. Field, and W.D. Little: "Asynchronous Arbiter Module"; IEEE Transactions on Computers, September 1975, pp. 931–933.
3. K. Soe Hojberg: "An Asynchronous Arbiter Resolves Resource Allocation Conflicts on a Random Priority Basis"; Computer Design, August 1977, pp. 120–123.
4. K. Soe Hojberg: "One–Step Programmable Arbiter for Multiprocessors"; Computer Design, April 1978, pp. 154–158.

Application Note	
Date of Issue	November 1986
Rev. Date	
Programmable Logic Devices	

AN11

PLD programmable retriggerable one-shot

Author: David Wong

FEATURES

- Programmable pulse-width/delay
- Maximum 256 clock cycles
- Asynchronous TRIGGER input
- Active-High and Active-Low outputs
- Asynchronous RESET
- 20-pin package

THEORY OF OPERATION

The one-shot consists of an PLS PLS159A and an external clock which may be part of the system in which this one-shot is to work. As shown in Figure 1 and Table 1 the PLS is configured to have a latch and an eight-bit binary up counter which is presettable by input data to any number less than 256. Since the input data is inverted before it is loaded into the registers, counting from the complements of the input to FF will give the correct number of counts as counting from the input down to 00.

Pulse-width/delay inputs may be the outputs of another device or switches. When /RESET goes Low, flip-flops are set to all 1's (terms PB and PA). At the rising edge of the next clock, data is latched into the registers (terms LB and LA). When /TRIG goes Low, it is latched into the input latch formed by term # 0, 1, 2 and 13. The output O_1 of the latch goes High and O_2 goes Low which enables the 8-bit counting cycle. The O_1 and $/O_1$ will maintain their output levels until the end of the counting cycle at which time the counter reaches the count FF, resets the latch by term # 13, and sets O_2 High. At the rising edge of the next clock, terms LA and LB cause data to be loaded again into the registers, and the device is ready for another /TRIG input. The output waveforms are illustrated in Figure 2.

If the /TRIG pulse-width is longer than the desired pulse-width of the one-shot, the device will react as mentioned above, and at the end of the count cycle new data will be loaded, another count cycle begins while the outputs remain set by the /TRIG input without changing throughout the change-over of one count cycle to another. O_{1a}, on the other hand, will go Low for one clock period at the change-over. As long as the /TRIG is Low, O_{1a} will continue to pulse Low for one clock period at the change-over of one count cycle to another. The output O_2 will pulse High for one clock cycle at the change-over. Figure 2 illustrates output wave-forms for both cases. The output wave-forms are as illustrated in Figure 2.

The one-shot is implemented by programming the PLS159A as shown in Table 1. The logic representation of the program is shown in Figure 3.

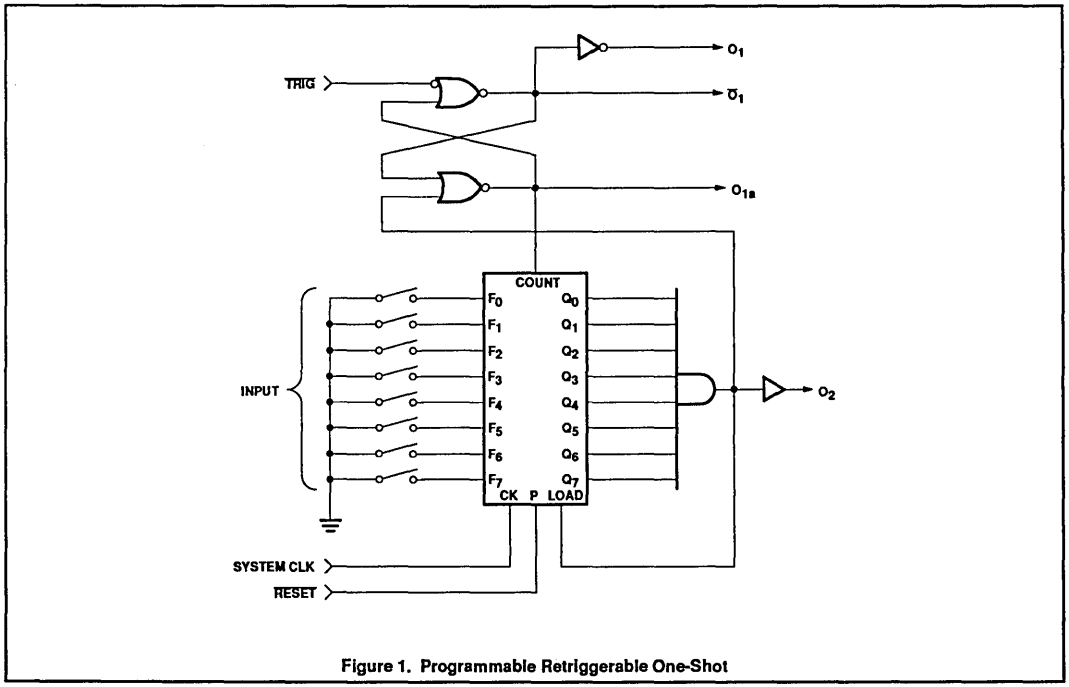


Figure 1. Programmable Retriggerable One-Shot

PLD programmable retriggerable one-shot AN11

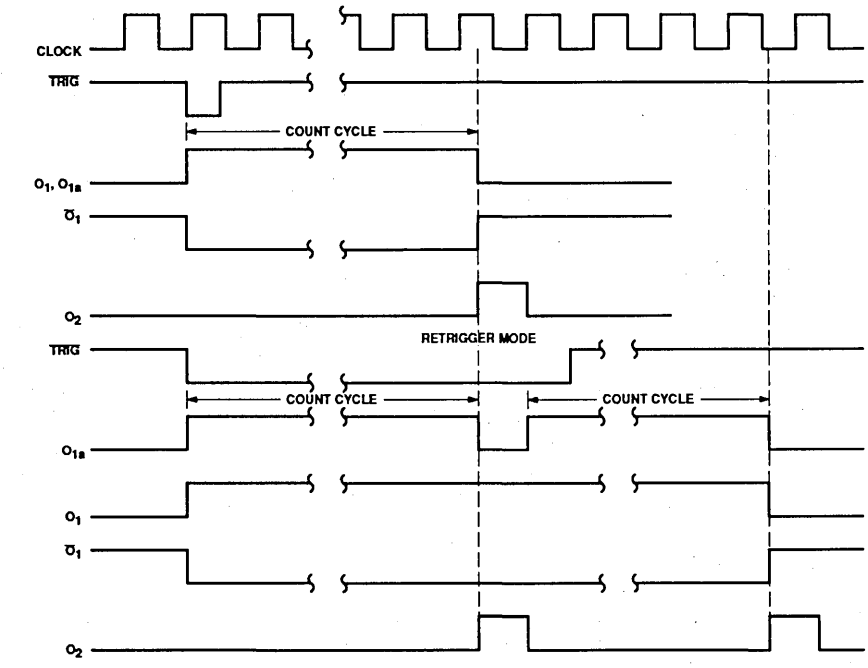
Table 1. PLS159A PLS Program Table

PROGRAMMABLE RETRIGGERABLE ONE-SHOT

TERM	C	I 3	I 2	I 1	I 0	B(I) 3	B(I) 2	B(I) 1	B(I) 0	Q(P) 7	Q(P) 6	Q(P) 5	Q(P) 4	Q(P) 3	Q(P) 2	Q(P) 1	Q(P) 0	REMARK	E_B —	E_A —	Q(N) 7	Q(N) 6	Q(N) 5	Q(N) 4	Q(N) 3	Q(N) 2	Q(N) 1	Q(N) 0	POLARITY L	POLARITY L	POLARITY H	POLARITY H	B(O) 3	B(O) 2	B(O) 1	B(O) 0	REMARK
		AND								F/F MODE A	A	A	A	A	A	A	A				(OR)																
0	—	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATCH	—	—	—	—	—	—	—	—	—	—	A	•	•	A	$B_0 = \overline{B}_3$
1	—	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	"	—	—	—	—	—	—	—	—	—	—	A	•	•	A		
2	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	"	—	—	—	—	—	—	—	—	—	—	•	A	•			
3																																				
4	—	—	—	H	—	—	H	—	—	—	—	—	—	—	—	—	—	COUNT CYCLE	—	—	—	—	—	—	—	—	—	0	•	•	•	•				
5	—	—	—	H	—	—	H	—	—	—	—	—	—	—	—	—	H	↓	—	—	—	—	—	—	—	—	0	—	•	•	•	•				
6	—	—	—	H	—	—	H	—	—	—	—	—	—	—	—	H	H		—	—	—	—	—	—	—	0	—	—	•	•	•	•				
7	—	—	—	H	—	—	H	—	—	—	—	—	—	—	H	H	H		—	—	—	—	—	—	0	—	—	—	•	•	•	•				
8	—	—	—	H	—	—	H	—	—	—	—	—	—	H	H	H	H		—	—	—	—	—	0	—	—	—	—	•	•	•	•				
9	—	—	—	H	—	—	H	—	—	—	—	—	H	H	H	H	H		—	—	—	0	—	—	—	—	—	—	•	•	•	•				
10	—	—	—	H	—	—	H	—	—	—	—	H	H	H	H	H	H		—	0	—	—	—	—	—	—	—	—	•	•	•	•				
11	—	—	—	H	—	—	H	—	—	—	H	H	H	H	H	H	H		0	—	—	—	—	—	—	—	—	—	•	•	•	•				
12																																				
13	—	—	—	—	—	—	—	—	—	H	H	H	H	H	H	H	H	LATCH	—	—	—	—	—	—	—	—	—	—	•	A	A	•				
14																																				
15																																				
16																																				
17																																				
18																																				
19																																				
20																																				
21																																				
22																																				
23																																				
24																																				
25																																				
26																																				
27																																				
28																																				
29																																				
30																																				
31																																				
F_C																																				
P_B	—	—	—	L	—	—	—	—	—	—	—	—	—	—	—	—	—	SET Q_4 TO Q_7 HIGH																			
R_B																																				
L_B	—	—	—	—	—	—	—	—	—	H	H	H	H	H	H	H	H	LOAD DATA AT NEXT CK																		
P_A	—	—	—	L	—	—	—	—	—	—	—	—	—	—	—	—	—	SET Q_0 TO Q_3 HIGH																		
R_A																																				
L_A	—	—	—	—	—	—	—	—	—	H	H	H	H	H	H	H	H	LOAD DATA AT NEXT CK																		
D3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																			
D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																			
D1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																			
D0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																			
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12																			
NAMES		1 TRIG		1 RESET		O_{1a}	/O_1	O_2	O_1																											

COMMENTS:
This one-shot will load data at the end of the count cycle. If TRIG pulse-width is longer than the count cycle, output B3 will go Low for one clock period and go High again for another count cycle. Outputs B_2 and B_0 stay Low and High respectively until TRIG goes High and count cycle is completed without interruption.

PLD programmable retriggerable one-shot AN11



PLS159A Programmable One-Shot

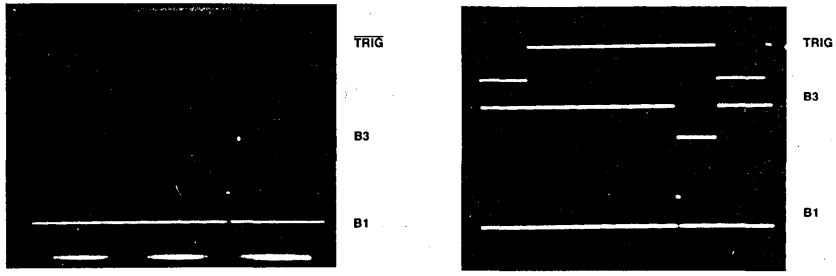


Figure 2. Timing Diagram of Programmable Retriggerable One-Shot

PLD programmable retriggerable one-shot AN11

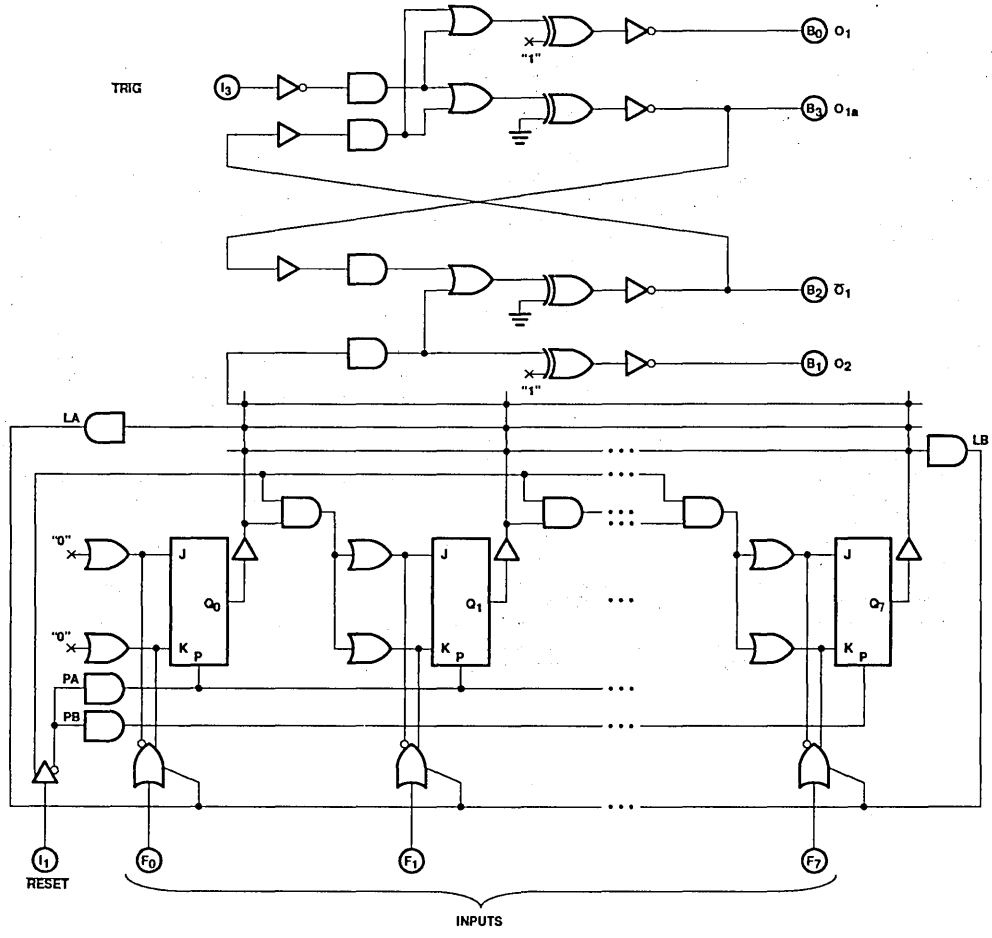


Figure 3. Logic Representation of Table 1

Philips Components-Signetics

Application Note	
Date of Issue	June 1988
Rev. Date	
Programmable Logic Devices	

AN14
Latches and flip-flops with PLS153

DESCRIPTION

Using the simple AND, OR and INVERT logic functions of the PLS153, memory functions such as latches and edge-triggered flip-flops may be implemented with a relatively small part of the chip and without external wiring. In this application note, we will discuss the implementation of two R-S latches, a D-latch, an edge-triggered R-S flip-flop, and an edge-triggered D flip-flop.

INTRODUCTION TO PLS153

To implement this function, let's first take a look at the PLS153 logic diagram and its programming table as shown in Appendices A and B. On the left side of the logic diagram (Appendix A) are eight dedicated inputs, I_0 to I_7, each of which has a true and a complement output. Each output is connected to the inputs of 32 AND functions (we will call them AND-terms from now on), the outputs of which are, in turn, connected to the inputs of ten OR functions. The output of each OR function is connected to one input of an Exclusive-OR function, which is in turn connected to a non-inverting output buffer. The function of the XOR is to control the output polarity. The output, in its virgin state, is non-inverting, since one side of the XOR is connected to ground by the fuse X_n, where n=0, 1, 3...9. To have the output inverted, one needs only to blow fuse X_n open so that the X_n input is unconditionally High. The output buffers are all 3-State buffers which may be enabled or disabled by their corresponding AND gates. The output buffers are disabled in their virgin state. All pins labeled "B" are bidirectional. Input buffers of the "B" pins are identical to those of the "I" pins.

The programming table shown in Appendix B emulates a truth table. All the inputs to the device are positioned on the left side, and all the outputs are on the right side. Each row in the table corresponds to an 18-input AND-term with up to ten outputs. On the left side, or the input side of the table, each column represents an input. The 18 columns represent input buffers I_0 to I_7, B_0 to B_9. To distinguish between inputs and outputs of the bidirectional pin, B(I) is used for input and B(O) is used for outputs as shown in the programming table. On the right side of the table, each column represents an output circuit ($B(O)_{0-9}$) which consists of an OR gate, an XOR, and a non-inverting 3-State buffer. The output buffers are controlled by AND-terms D_0 to D_9, the inputs of which may be connected to any number of the 18 inputs.

The polarity of the outputs is defined by the POLARITY entries which are on the upper right corner of the programming table.

To program the inputs to the AND-terms, an "H" will cause the fuse of the inverting input buffer to be blown, leaving the non-inverting buffer connected to the AND-term; an "L" will do the opposite. A "–" will cause both fuses to be blown, and therefore the programmed input is a "Don't care". A "0", the virgin state of the device, has both fuses intact, which causes the output of the AND-term to be unconditionally Low.

To program the outputs, a "." causes the fuse that connects the output of AND-term to the input of an OR to be blown and thus renders the output inactive. An "A" causes the fuse to remain intact and thus the output is active.

The output polarity of each output buffer may be programmed by entering an "H" or an "L" in the POLARITY section. An "L" causes the XOR to blow its grounding fuse and become inverted, whereas an "H" leaves the fuse intact and the output is non-inverted.

To AND several inputs, we put them in a row; to OR several inputs, we put them in different rows, as shown in illustrations in Appendix B.

Latches and flip-flops with PLS153 AN14

SIMPLE R-S LATCH

A simple R-S latch may be formed by cross-coupling two NAND functions together as shown in Figure 1.

As an illustration, let's assign the input R to I_0 of the PLS153, input S to I_1, output Q to B_0, and output $\overline{Q}$ to B_1. As shown in Table 1, to form the NAND gates we need to program the POLARITY Low on $B(O)_0$ and $B(O)_1$. To unconditionally enable the output buffers, we "dash" out all inputs to D_0 and D_1. As for the inputs, we put an "H" on I_0, term-0 for the input R, non-inverted; another "H" on $B(I)_1$, term-0 for the feedback from Q. In the same manner, we program I_1, term-1 and $B(O)_0$ "H". The POLARITY, rows 0, 1, D_0 and D_1, forms a "truth table" with which one can analyze his own or someone else's design. The program in Table 1 may be illustrated as shown in Figure 2.

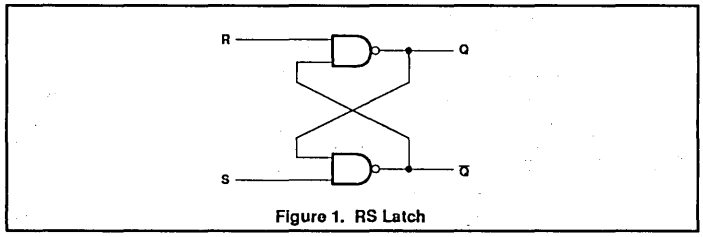


Figure 1. RS Latch

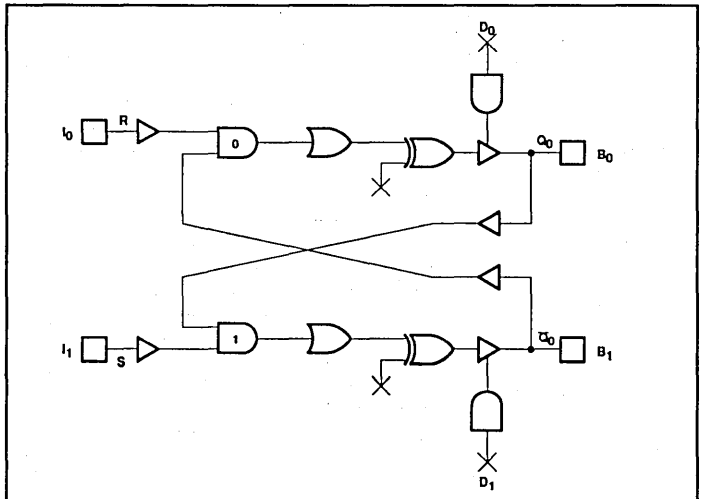


Figure 2. RS Latch

Latches and flip-flops with PLS153 AN14

Table 1. PLS153/153A Programming Table

CODE NO.

TERM	AND I 7	6	5	4	3	2	1	0	AND B(I) 9	8	7	6	5	4	3	2	1	0	REMARKS	POLARITY	OR B(O) 9	8	7	6	5	4	3	2	1	0	REMARKS
																													L	L	
0	—	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—	H	—			•	•	•	•	•	•	•	•	•	A	$Q = /(R \cdot \overline{Q})$
1	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	H			•	•	•	•	•	•	•	•	A	•	$\overline{Q} = /(S \cdot Q)$
2																															
3																															
4																															
5																															
6																															
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
D9																															
D8																															
D7																															
D6																															
D5																															
D4																															
D3																															
D2																															
D1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
PIN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9			19	18	17	16	15	14	13	12	11	9	
VARIABLE NAME							S	R									$\overline{Q}_0$	Q_0											$B(O)_1$ ($\overline{Q}_0$)	$B(O)_0$ (Q_0)	

PLS153

Latches and flip-flops with PLS153 AN14

ANOTHER SIMPLE R-S LATCH

Another way to implement a simple latch is shown in Figure 3, in which two NOR functions are cross-coupled to form a latch.

As with the previous example, we first define the input and output pins. For this example, we use I_2 for the R input, I_3 for the S input, B_2 for the Q output, and B_3 for the $\overline{Q}$ output. We program B_2 and B_3 to have inverted outputs by programming POLARITY of B_2 and B_3 Low, as shown in Table 2. Terms 6 and 7 are ORed together by $B(O)_2$, rows 6 and 7. In the same manner $B(O)_3$ ORs Terms 8 and 9. The programmed table of this design may be represented as shown in Figure 4.

Since each AND-term of the PLS153 can accommodate up to 18 inputs (true or inverting inputs of eight from I_0 to I_7 and ten from B_0 to B_9), and each OR circuit can be connected to up to thirty-two AND-terms, we can add additional features such as those shown in Figure 5.

The programming of this design is left to the reader as an exercise.

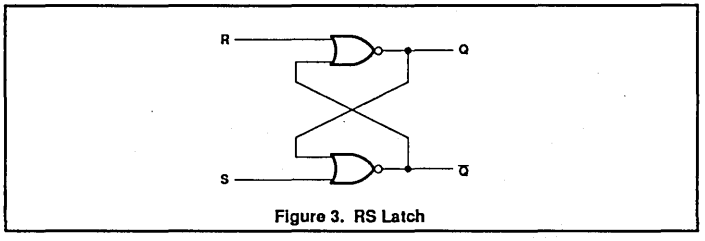


Figure 3. RS Latch

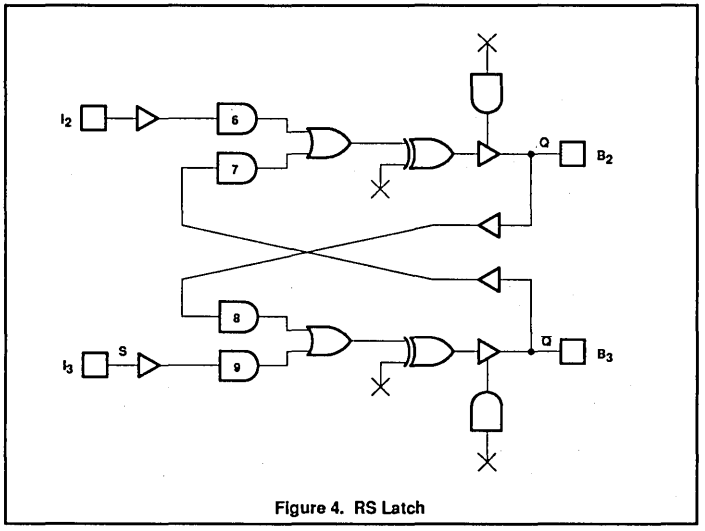


Figure 4. RS Latch

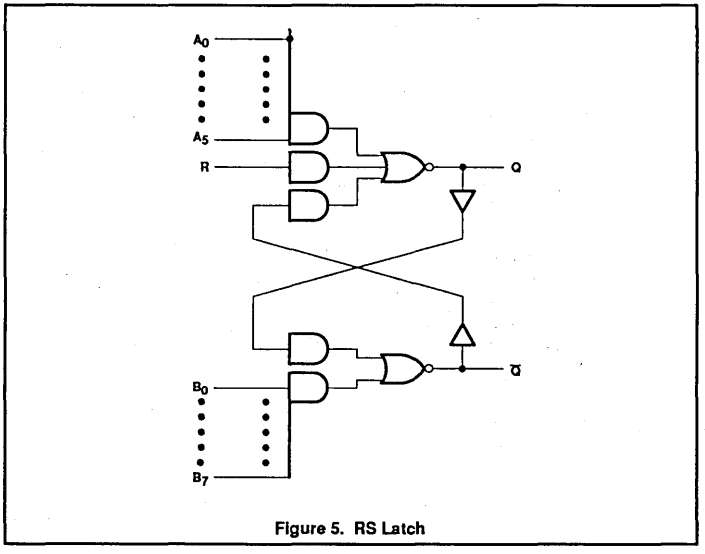


Figure 5. RS Latch

Latches and flip-flops with PLS153 AN14

Table 2. PLS153/153A Programming Table

CODE NO.

TERM	AND I 7	6	5	4	3	2	1	0	AND B(I) 9	8	7	6	5	4	3	2	1	0	REMARKS	POLARITY	OR B(O) 9	8	7	6	5	4	3	2	1	0	REMARKS
																					L	L	L	L							
0	—	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—	H	—	R·$\overline{Q}$		•	•	•	•	•	•	•	•	•	A	Q = /(R·$\overline{Q}$)
1	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	H	S·Q		•	•	•	•	•	•	•	•	A	•	$\overline{Q}$ = /(S·Q)
2																															
3																															
4																															
5																															
6	—	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	R		•	•	•	•	•	•	•	A	•	•	Q = /(R + $\overline{Q}$)
7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H	—	—	—	$\overline{Q}$		•	•	•	•	•	•	•	A	•	•	Q = /(R + $\overline{Q}$)
8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H	—	—	Q		•	•	•	•	•	•	A	•	•	•	$\overline{Q}$ = /(Q + S)
9	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S		•	•	•	•	•	•	A	•	•	•	$\overline{Q}$ = /(Q + S)
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
D9																															
D8																															
D7																															
D6																															
D5																															
D4																															
D3																															
D2																															
D1																															
D0																															
PIN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9			19	18	17	16	15	14	13	12	11	9	
VARIABLE NAME					S	R	S	R							$\overline{Q}$	Q	$\overline{Q}$	Q									B_3 ($\overline{Q}_1$)	B_2 (Q_1)	B_1	B_0			

PLS153

Latches and flip-flops with PLS153 AN14

D-LATCH

A simple D-latch can be constructed with a PLS153 as shown in Figure 6.

This circuit may be easily programmed into the PLS153 as shown in Table 3. The program may, in turn, be represented as shown in Figure 7.

This circuit may be expanded to have multiple D-latches using the same latch enable (LE).

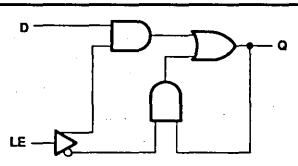


Figure 6. D Latch

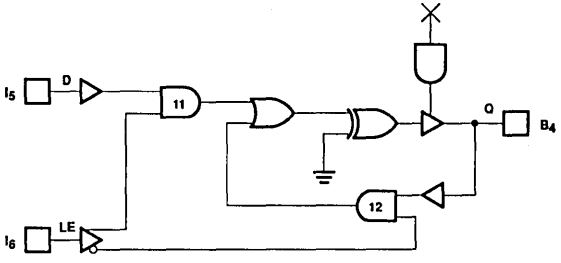


Figure 7. D-Latch

R-S FLIP-FLOP

Two R-S latches may be combined to form a master-save flip-flop that is triggered at the rising-edge of the clock (or the falling-edge of the clock, if the designer so desires). Figure 8 shows a combination of two sets of cross-coupled NOR gates concatenated to form the flip-flop. The implementation of this circuit using PLS153 is as illustrated in Table 4 and Figure 9.

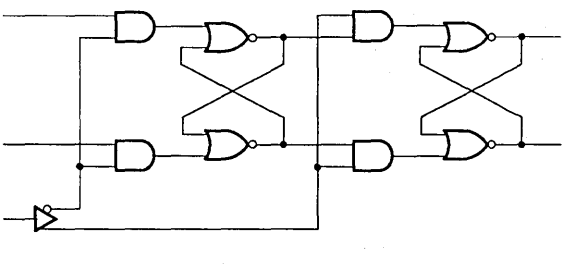


Figure 8. RS Flip-Flop

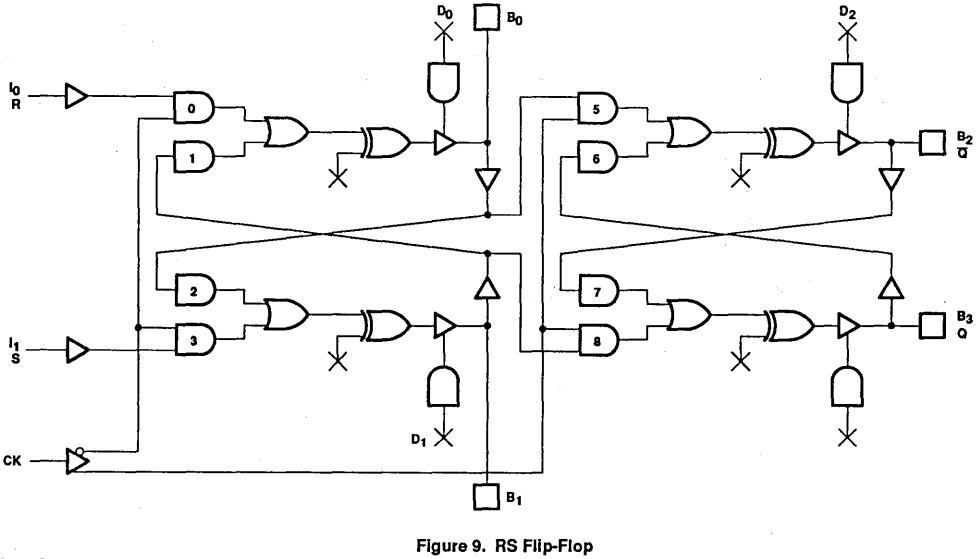
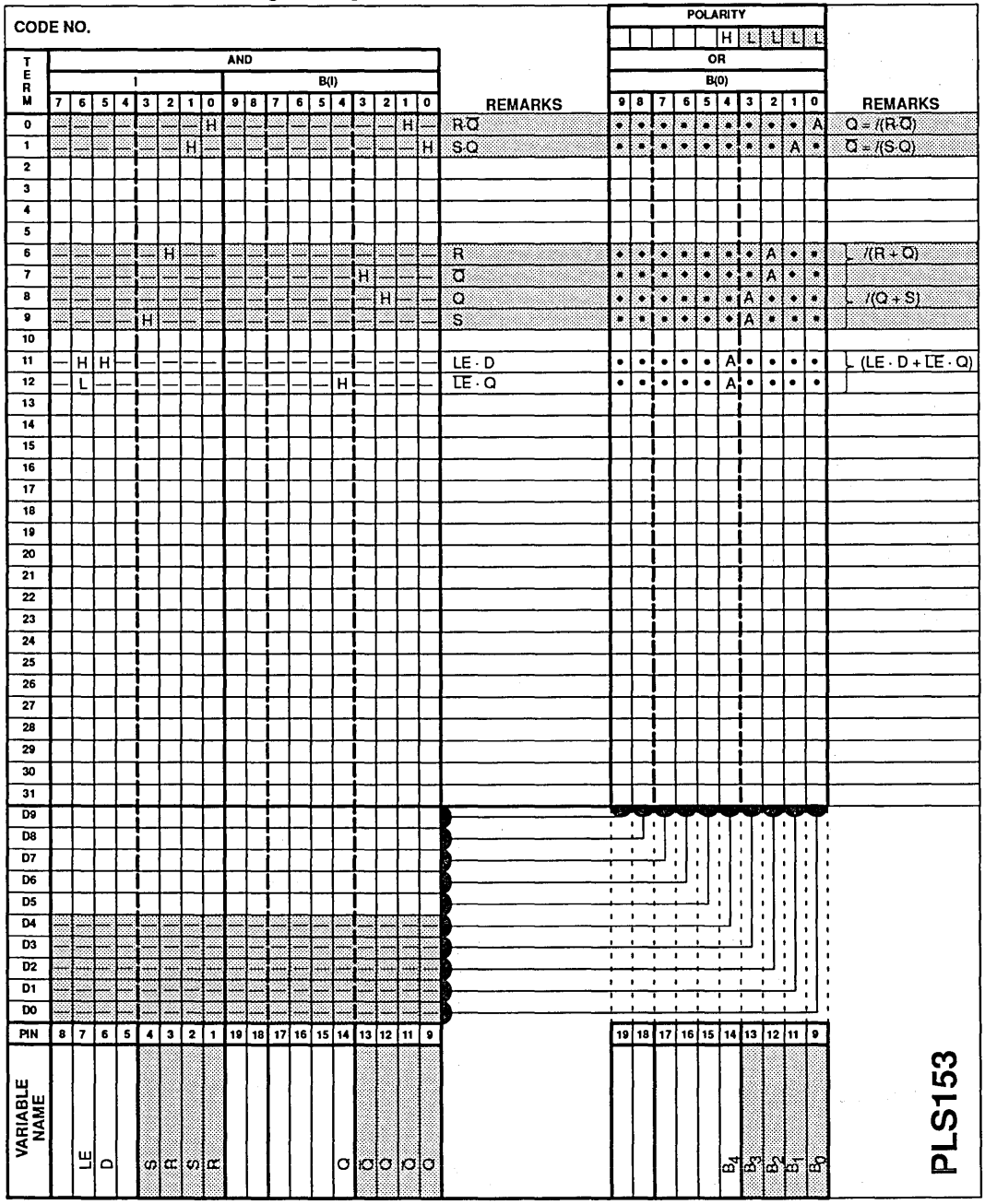


Figure 9. RS Flip-Flop

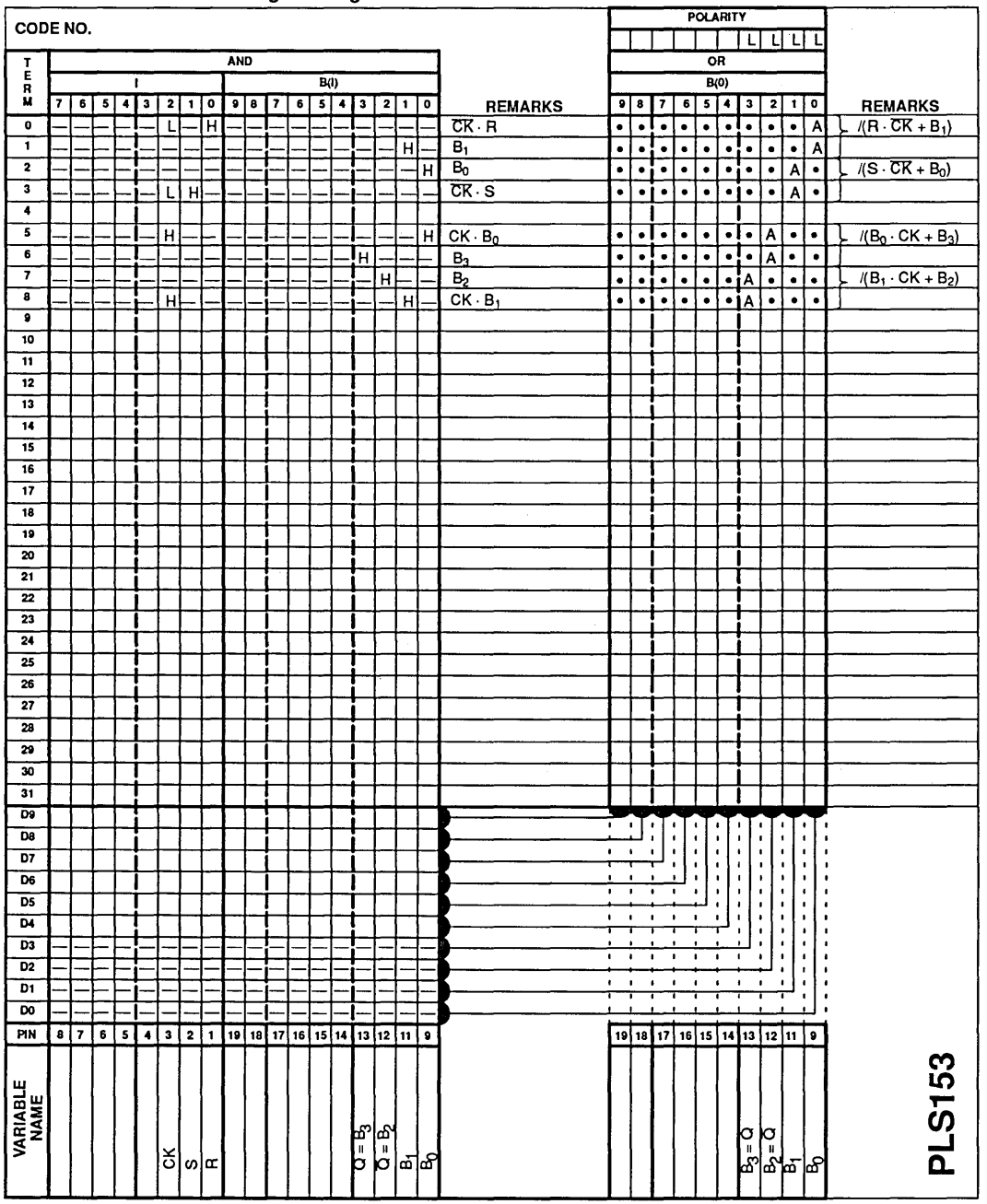
Latches and flip-flops with PLS153 AN14

Table 3. PLS153/153A Programming Table



Latches and flip-flops with PLS153 AN14

Table 4. PLS153/153A Programming Table



Latches and flip-flops with PLS153 AN14

D FLIP-FLOP

An edge-triggered master-slave D flip-flop may be constructed with two D-latches in the manner shown in Figure 10.

An PLS153 may be programmed as shown in Figure 11 to implement the D flip-flop which is equivalent to the circuit shown in Table 5 in the PLS153 logic representation.

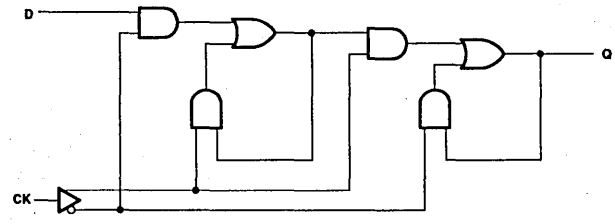


Figure 10. D Flip-Flop

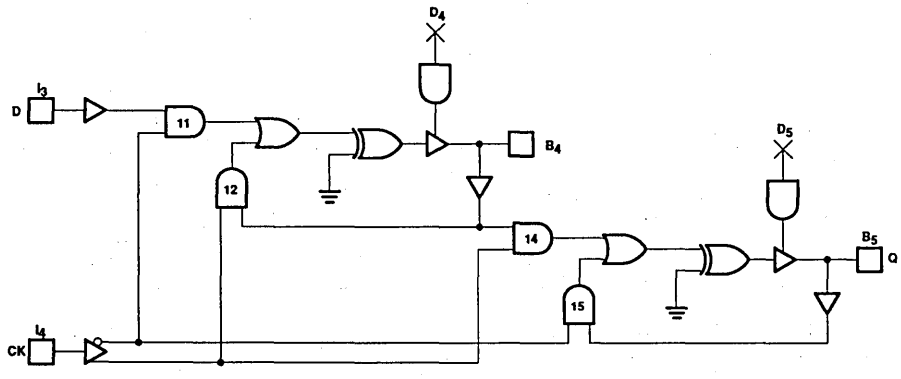
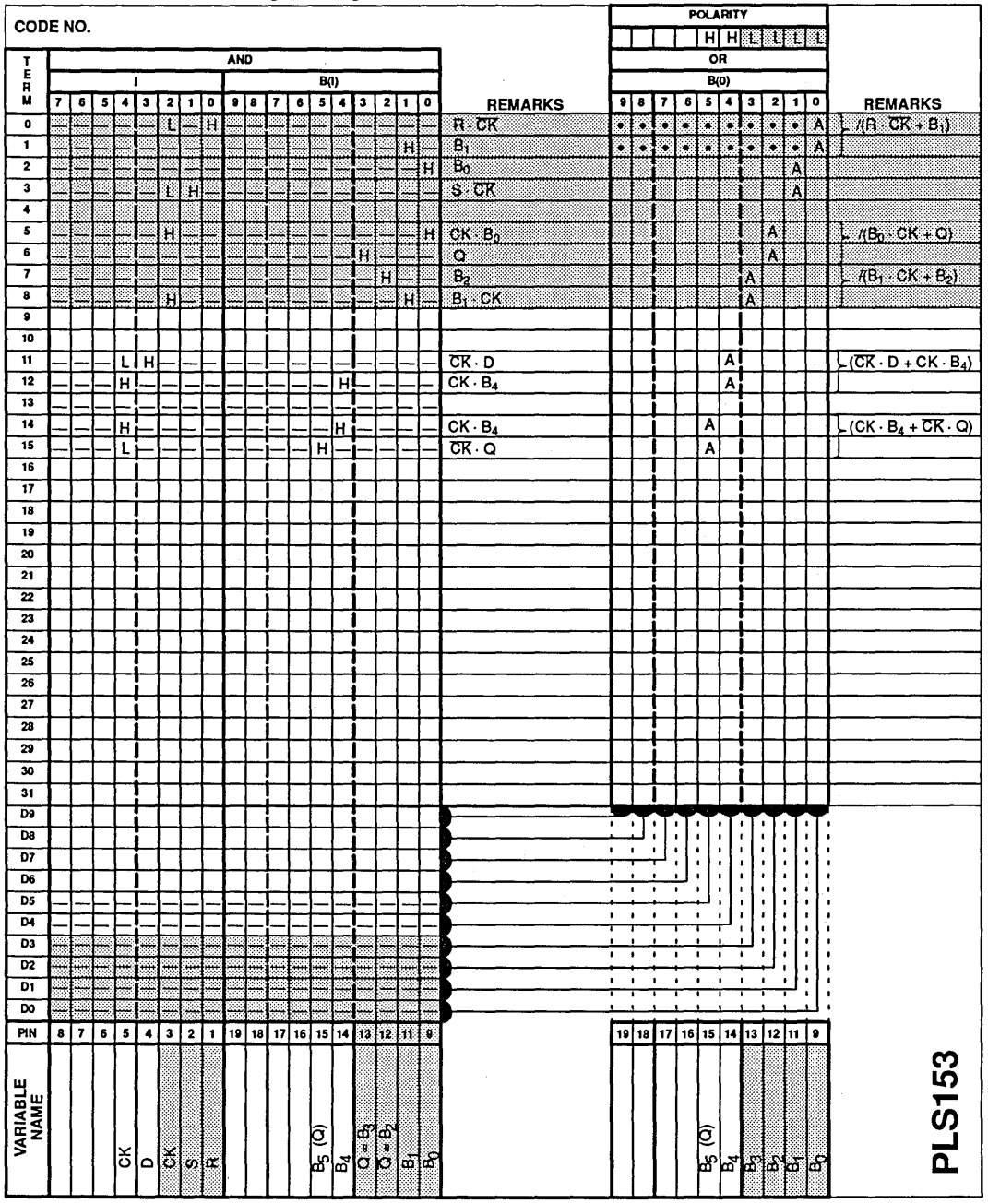


Figure 11. D Flip-Flop

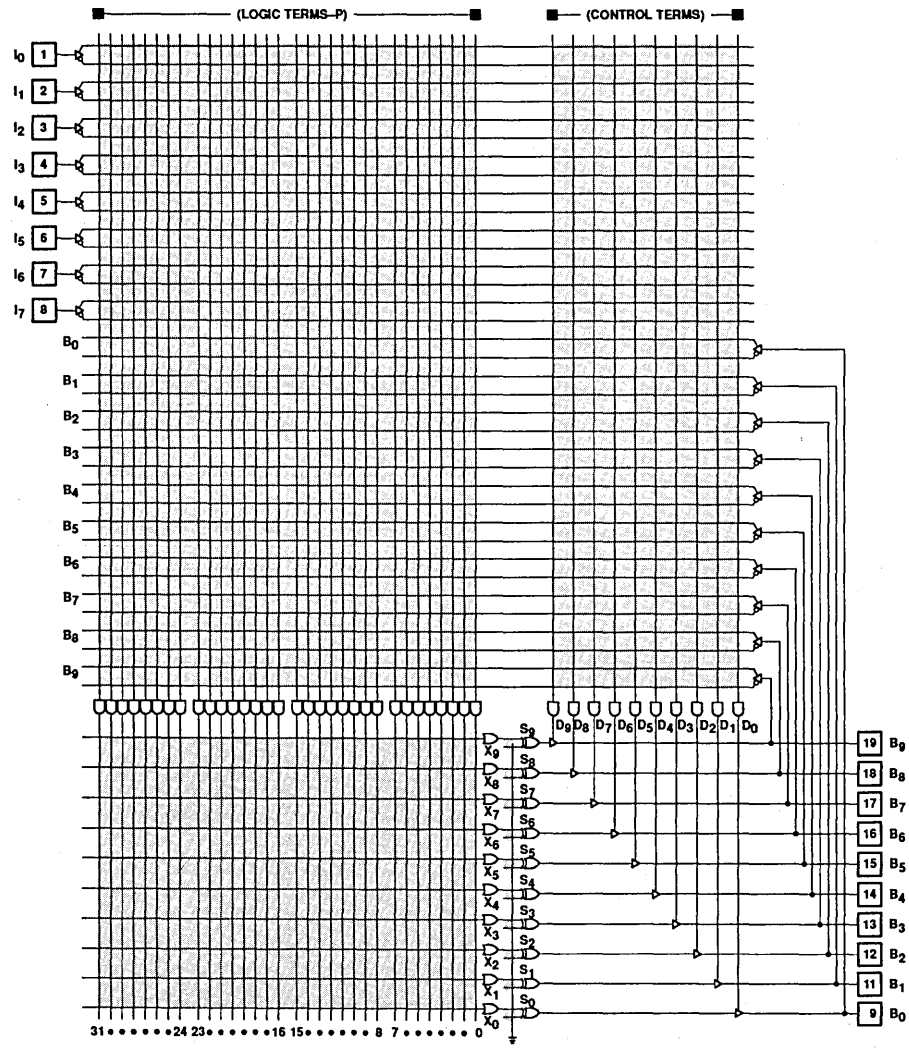
Latches and flip-flops with PLS153 AN14

Table 5. PLS153/153A Programming Table



Latches and flip-flops with PLS153 AN14

APPENDIX A

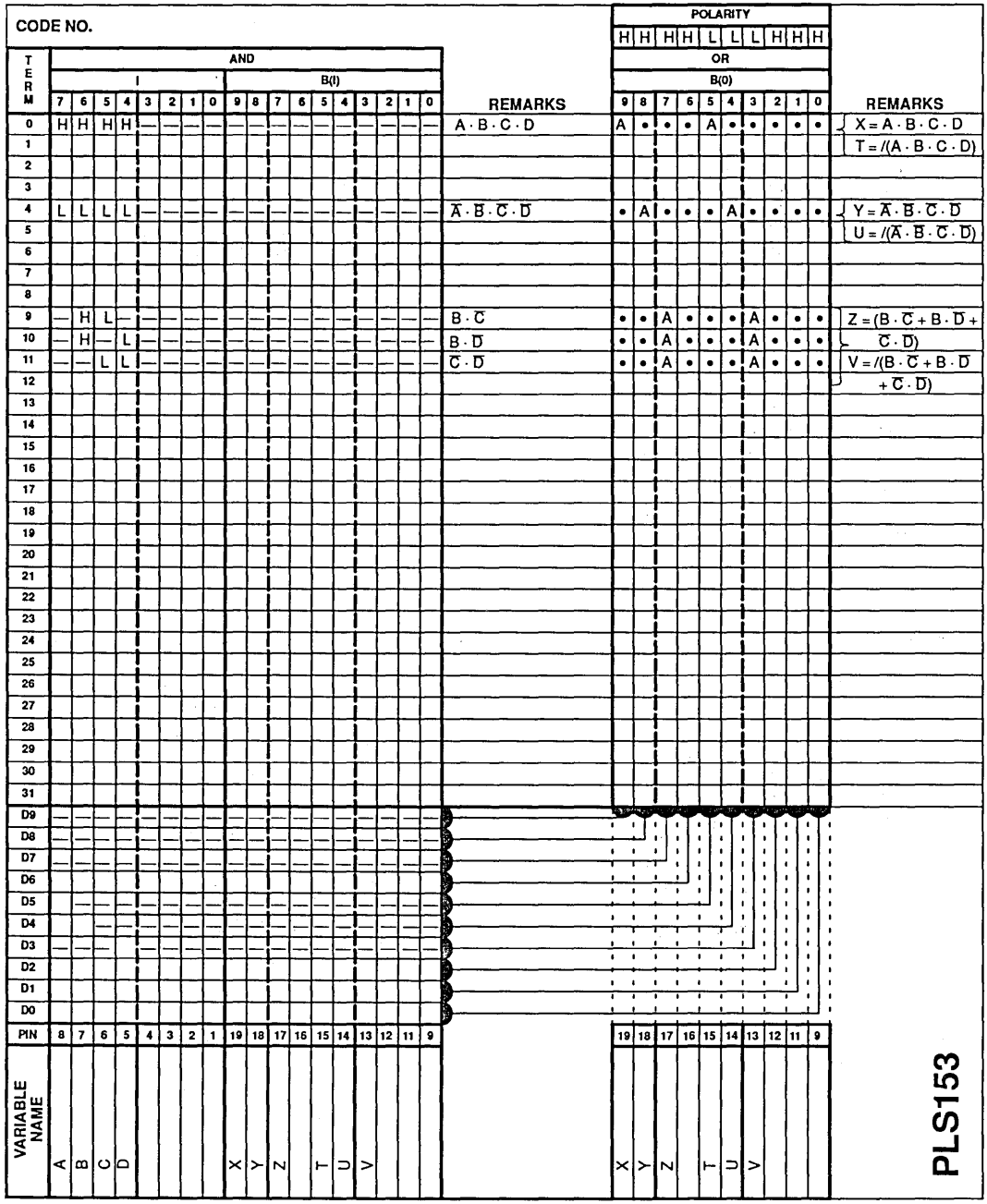


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ◎ Programmable connection.

Figure A-1. PLA Logic Diagram

Latches and flip-flops with PLS153 AN14

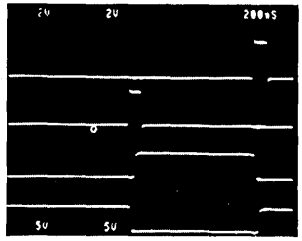
APPENDIX B. PLS153/153A Programming Table



Latches and flip-flops with PLS153 AN14

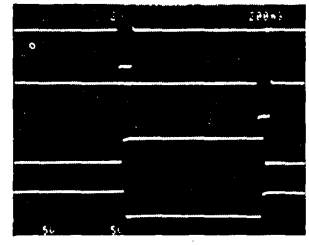
APPENDIX C

R-S LATCH (Cross-Coupled NOR)
See Figures 3 and 4 and Table 2



R Pin 3
S Pin 4
Q Pin 12
$\overline{Q}$ Pin 13

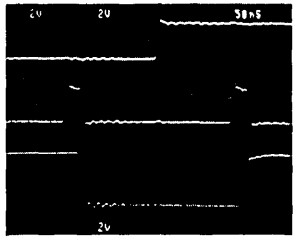
R-S LATCH (Cross-Coupled NAND)
See Figures 1 and 2 and Table 1



$\overline{S}$ Pin 1
R Pin 2
Q Pin 9
$\overline{Q}$ Pin 11

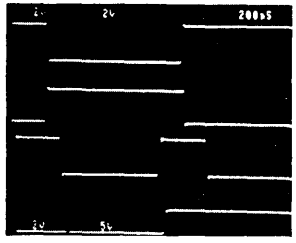
Typical set-up time ≅ 0ns
Typical hold time ≅ 0 – 5ns
Typical propagation delay ≅ 20ns

D-LATCH
See Figures 6 and 7 and Table 3



D Pin 6
LE Pin 7
Q Pin 14

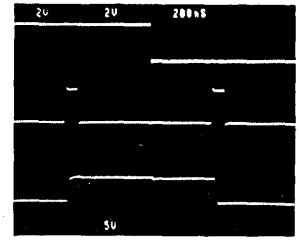
EDGE-TRIGGERED R-S Flip-Flop
See Figures 8 and 9 and Table 4



R Pin 1
S Pin 2
CLOCK Pin 3
Q Pin 13

NOTE: Timing requirements/performances are the same as the R-S latches.

EDGE-TRIGGERED D Flip-Flop
See Figures 10 and 11 and Table 5



D Pin 4
CLOCK Pin 5
Q Pin 15

NOTE: Timing requirements/performances are the same as the D latch.

Figure C-1. Timing Photos

Philips Components-Signetics

Application Note	
Date of Issue	June 1988
Rev. Date	October 1990
Programmable Logic Devices	

AN15
PLS159A primer

INTRODUCTION

The PLS159A is a programmable logic sequencer which consists of four dedicated inputs, four bidirectional I/O's, eight flip-flops, thirty two 16-input AND gates, twenty 32-input OR gates, and a complement array. Each flip-flop has a bidirectional I/O and may be individually programmed as J-K or D flip-flop, or switch between the two types dynamically. The flip-flops will accept data from the internal logic array or from the bidirectional I/O, or they may be set or reset asynchronously from the AND array. The output polarity of the four bidirectional I/O's are programmable and the direction is controlled by the AND array. Figure 1 is the logic diagram of PLS159A.

PROGRAMMING THE PLS159A

The programming table is shown in Table 1 where there is a place for everything that is shown in Figure 1. The program table is basically divided into two main sections. The left hand side of the table, section A, represents the input side of the AND gates, while the right hand side, section B, represents the OR gates sections which includes the flip-flops and the combinatorial outputs B(0) to B(3). The flip-flops modes are defined in section C and the output polarities of the combinatorial outputs are defined in section E. The programming symbols are detailed in Figure 2.

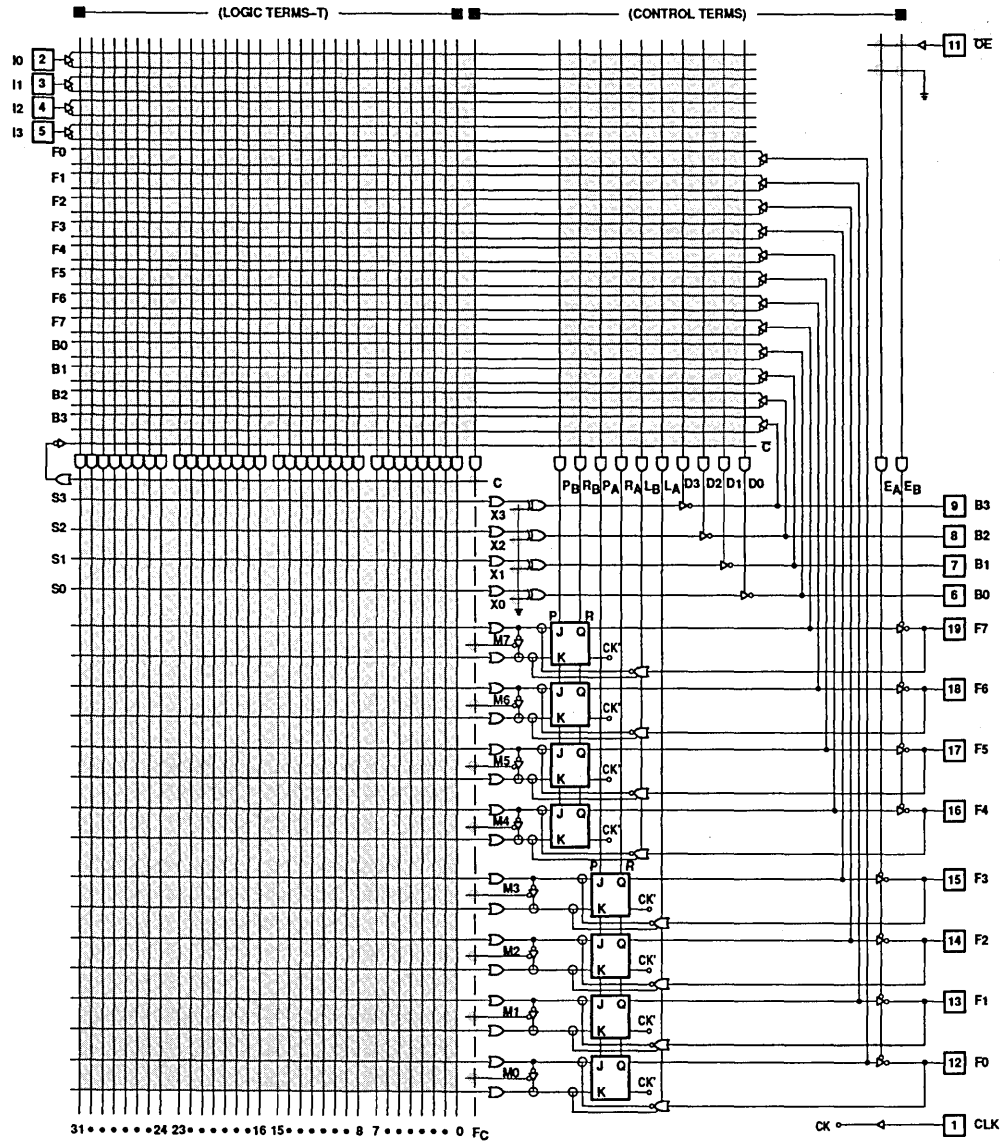
As shown in Table 1, the programming table is very similar to a truth table. Each column in section A represents an input to the 32 AND gates, and each row represents an AND gate connecting to 17 inputs. Columns I_0 to I_3 represent the 4 dedicated inputs, I_0 to I_3. Columns $B(I)_0$ to $B(I)_3$ represent the inputs of the 4 bidirectional I/O, B_0 to B_3. Columns $Q(P)_0$ to $Q(P)_7$ represent the feedback, F_0 to F_7, from the flip-flops (the present state). Column "C" represents the complement array.

As shown in Figure 1, the outputs of the AND gates are connected to an array of OR gates which, in turn, are connected to either flip-flops or output circuits. Columns $Q(N)_0$ to $Q(N)_7$ represent the next state which the flip-flops will be in. Columns $B(O)_0$ to $B(O)_3$ represent the combinatorial outputs B_0 to B_3.

Each row represents an AND gate with 17 inputs each of which may be true and/or complement and is, therefore, a perfect decoder. Referring to the programming symbols in Figure 2, to implement the equation

$$Z = A * B * C * D,$$

all one has to do is to enter one line as shown in Table 2, term-0.

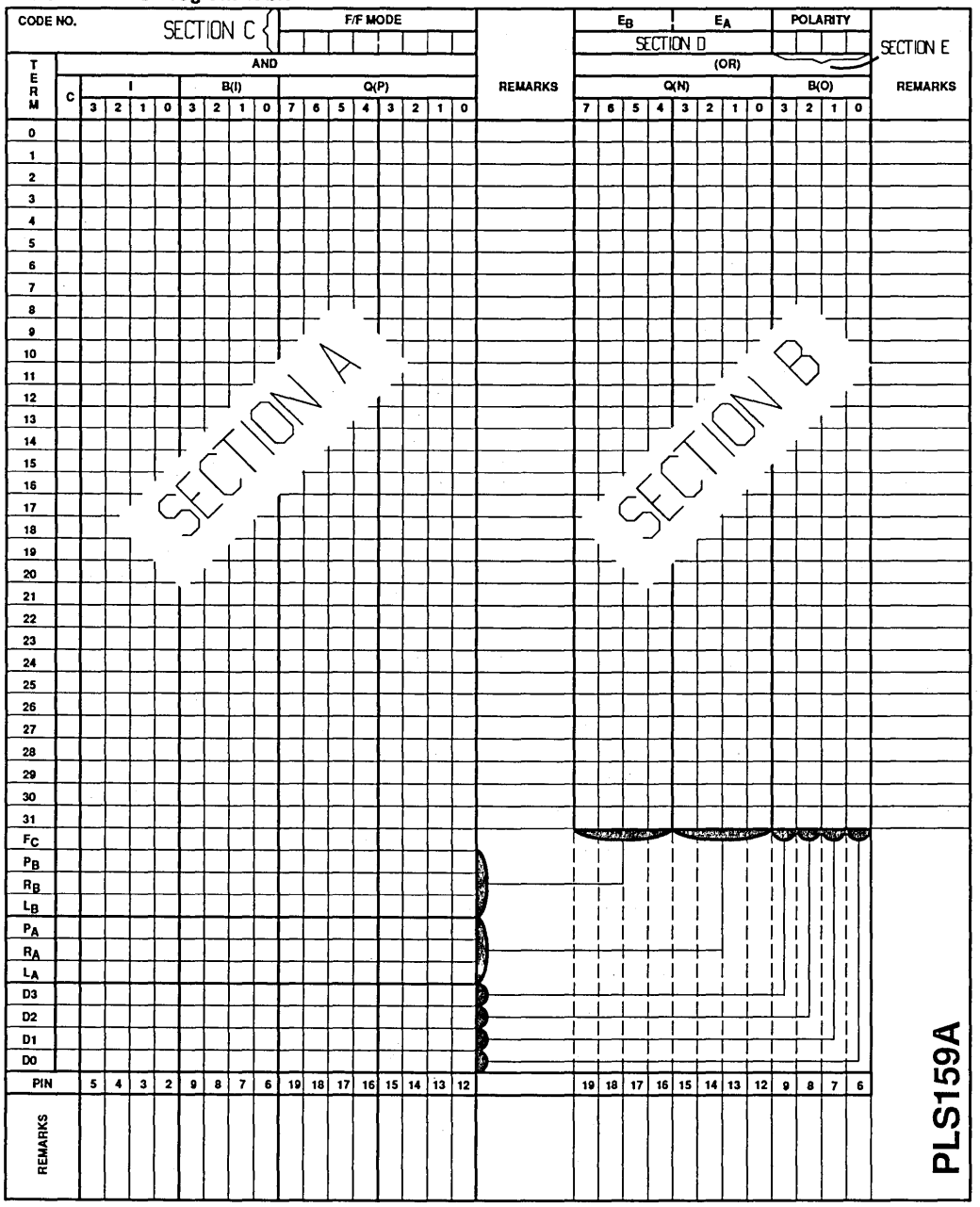


NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ▒ Programmable connection.

Figure 1. PLS Logic Diagram

PLS159A primer AN15

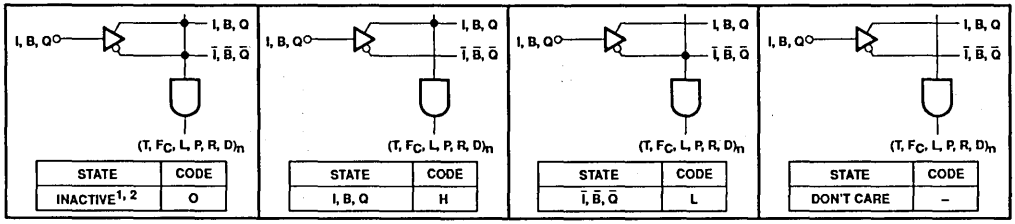
Table 1. PLS Program Table



The PLS can be programmed by means of Logic Programming equipment.

With Logic programming, the AND/OR–EX-OR input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

In these Tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:



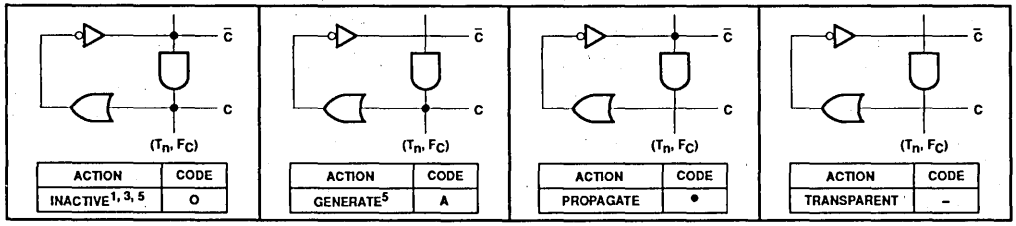
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
Ī, B̄, Q̄	L

STATE	CODE
DON'T CARE	–

"AND ARRAY – (I), (B), (Qp)



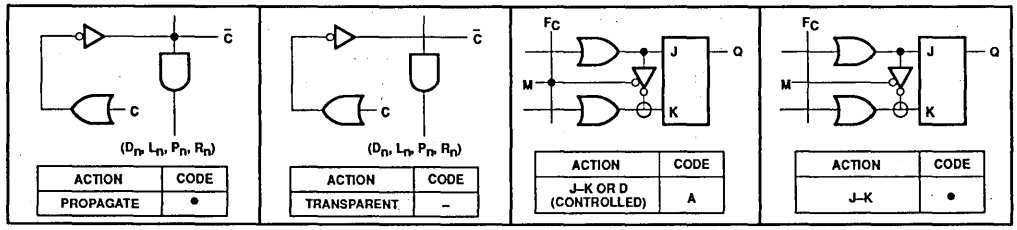
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"COMPLEMENT" ARRAY – (C)



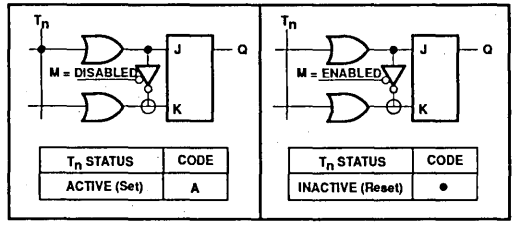
ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

ACTION	CODE
J–K OR D (CONTROLLED)	A

ACTION	CODE
J–K	•

"COMPLEMENT" ARRAY (cont.) "OR" ARRAY – (MODE)

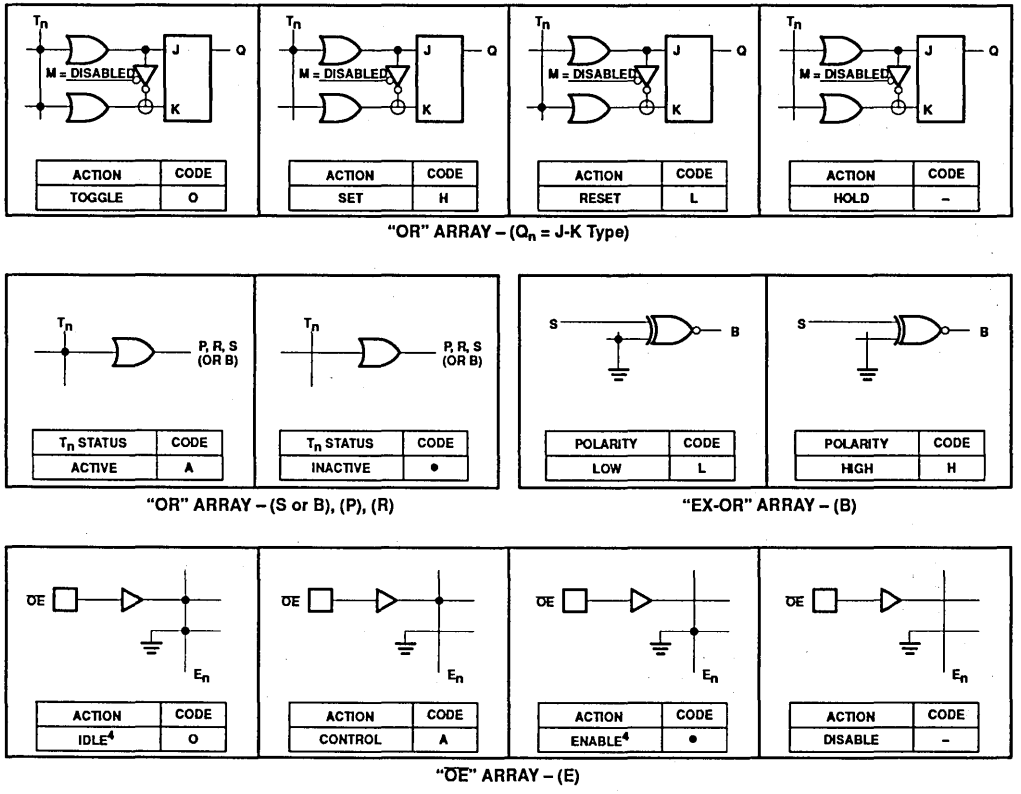


T_n STATUS	CODE
ACTIVE (Set)	A

T_n STATUS	CODE
INACTIVE (Reset)	•

"OR" ARRAY – (Q_N = D-Type)

Figure 2



NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate $(T, F_C, L, P, R, D)_n$ will be unconditionally inhibited if any one of the I, B, or Q link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates $(L, P, R, D)_n$ due to their lack of "OR" array links.

Figure 2 (cont.)

PLS159A primer AN15

Table 2. PLS Program Table

CODE NO.										F/F MODE								REMARKS	E_B				E_A				POLARITY				REMARKS
																											H	H	H	H	
TERM	AND																		(OR)												
	C	I				B(I)				Q(P)									Q(N)								B(O)				
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0	3	2	1	0	
0	—	H	H	H	H	—	—	—	—	—	—	—	—	—	—	—	—	A*B*C*D	—	—	—	—	—	—	—	—	•	•	•	A	Z = A*B*C*D
1																															
2	—	L	H	L	—	—	—	—	—	—	—	—	—	—	—	—	—	/A*B*/C	—	—	—	—	—	—	—	—	•	•	A	•	Y = /A*B*/C
3																															
4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	"I"	—	—	—	—	—	—	—	—	•	A	•	•	X = I
5																															
6	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	"O"	O	O	O	O	O	O	O	O	A	A	A	A	VIRGIN CONDITION
7	—	—	—	O	—	—	—	—	—	—	—	—	—	—	—	—	—	"O"	—	—	—	—	—	—	—	—	A	•	•	•	W = O
8																															
9	—	H	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	A*/B	—	—	—	—	—	—	—	—	A	•	•	•	W = A*/B
10	—	—	—	H	L	—	—	—	—	—	—	—	—	—	—	—	—	C*/D	—	—	—	—	—	—	—	—	A	•	•	•	W = C*/D
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
F_C																															
P_B																															
R_B																															
L_B																															
P_A																															
R_A																															
L_A																															
D3																															
D2																															
D1																															
D0																															
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12		19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS		A	B	C	D																						W	X	Y	Z	PLS159A

Notice that only I_0 to I_3 on the left hand side and $B(O)_4$ on the right hand side have entries to implement the equation. All unused columns are dashed out or dotted out.

To implement the equation

Y=/A * B * /C,

enter one line as shown in Table 2, term-2 where the entry "H" represents the non-inverting input buffer while the entry "L" represents the inverting buffer. To have the AND gate to be unconditionally "High", dash out all the inputs of that particular AND gate as shown in Table 2, term-4. The virgin condition of the device, as shipped from the factory, has all connections intact, which means that the inverting and the non-inverting buffers of the same inputs are connected together. Such connection will cause the AND gate to be unconditionally "Low" as shown in Table 2, terms 6 and 7. The unconditional High and Low states are normally useful only internally and seldom brought out to the output pins.

To implement the equation

W=A * /B+C * /D,

enter one line for A * /B and another line for C * /D as shown in Table 2, terms 9 and 10. Use one line to AND something together; use different lines to OR something together — one line per item to be OR'ed.

All the pins which are labelled B's are bidirectional I/O pins. Their input buffers are represented by the B(I) columns on the left hand side of the programming table. An "H" entry represents the non-inverting buffer and an "L" entry represents the inverting buffer. Their output buffers are represented by the B(O) columns on the right hand side of the table. An "A" entry means that the output is active (connected to the AND gates); a "." entry means that the output is inactive (not connected). The outputs may be programmed to be inverting or noninverting. The polarity of each output is determined by its exclusive OR gate (Figure 1 and Figure 2). To have a non-inverting output, enter an "H" in the section labelled "POLARITY" (Table 1, Section E). To have an inverting output, enter an "L". For example, Table 3, terms -0 and -2 implement the equation

Z=/(A * B) and Y=A * B

respectively. The above two equations may also be implement by term-4 which uses the same AND gate to drive two OR gates.

Besides being able to have programmable Active-High or Active-Low output, the programmable output polarity feature also low output, the programmable output polarity feature also allows the user to minimize his AND term utilization by converting his logic equation into other forms such as conversion by De Morgan Theorem.

For example, the equation

X=A+B+C+D

takes four AND terms to implement as shown in Table 3, terms 6 to 9. By using De Morgan Theorem, the same equation is changed to

/W=/A * /B * /C * /D

The result is as shown in term 11 — a saving of three AND terms. The output buffers are disabled in their virgin states so that they all behave as inputs. The buffers are enabled or disabled by their corresponding Control AND terms D_0 to D_3 (see Figure 1). The Control AND terms are represented in the programming table on the last four rows on the left hand side. Dashing out all the inputs will cause the output buffer to be unconditionally enabled, whereas a "0" (zero) will cause the buffer to be unconditionally disabled. The buffers may also be controlled by a logical condition, e.g. A * /B * /C, etc.

There are eight flip-flops on the chip each of which may be programmed as a J/K or a D flip-flop, or they may be programmed to switch dynamically. As shown in Figure 1, each flip-flop is a J/K to begin with. A 3-State inverter is connected in between the J and K inputs of each flip-flop, which when enabled by the AND gate F_C, will cause the flip-flop to function as a D flip-flop. The inverters are enabled by F_C through fuses M_0 to M_7. A "." in the F/F Mode entry of the programming table means that particular fuse is to be disconnected and that particular flip-flop is to be J/K. An "A" entry will leave the M fuses intact, which allow the flip-flop to be D or J/K as controlled by the output of F_C (see Figure 2, "OR" ARRAY — (MODE)). The inputs to the flip-flops are represented by the programming table as the next state, $Q(N)_{0\ to\ 7}$ since their inputs are from the OR array. The outputs of these registers are connected to their respective 3-State inverting output buffers, four of which are controlled by EA and the other four by EB. A "." in EA will enable outputs F_0 to F_3, whereas a "–" will disable them. An "A" will allow the output buffers to be controlled by /OE, pin 11. Table 4, terms 0, 1 and 3 represent the following equations

Q_0: J=A * C+/B * /E eq. 1

Q_0: K=A * /C eq. 2

Notice that the J input in equation 1 is represented by the "H" entry in terms-0 and 1, column $Q(N)_0$ while the K input in equation 2 is represented by the "L" entry in term-3, column $Q(N)_0$. An undefined input, J or K, is considered "Low".

PLS159A primer AN15

Table 3. PLS Program Table

CODE NO.									F/F MODE									E_B				E_A				POLARITY				
																										L	H	H	L	
TERM	AND																	(OR)												
	C	I				B(I)				Q(P)								REMARKS	Q(N)								B(O)				REMARKS
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0	3	2	1	0	
0	—	—	—	H	H	—	—	—	—	—	—	—	—	—	—	—	—	A*B	—	—	—	—	—	—	—	—	•	•	•	A	Z = /(A*B)
1																															
2	—	—	—	H	H	—	—	—	—	—	—	—	—	—	—	—	—	A*B	—	—	—	—	—	—	—	—	•	•	A	•	Y = A*B
3																															
4	—	—	—	H	H	—	—	—	—	—	—	—	—	—	—	—	—	A*B	—	—	—	—	—	—	—	—	•	•	A	A	Z = /(A*B)
5																															Y = A*B
6	—	H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	A	—	—	—	—	—	—	—	—	•	A	•	•	
7	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	B	—	—	—	—	—	—	—	—	•	A	•	•	
8	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	—	C	—	—	—	—	—	—	—	—	•	A	•	•	X = A+B+C+D
9	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	D	—	—	—	—	—	—	—	—	•	A	•	•	
10																															
11	—	L	L	L	L	—	—	—	—	—	—	—	—	—	—	—	—	/A*/B*/C*/D	—	—	—	—	—	—	—	—	A	•	•	•	W= /(/A*/B*/C*/D)
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
F_C																															
P_B																															
R_B																															
L_B																															
P_A																															
R_A																															
L_A																															
D3																															
D2																															
D1																															
D0																															
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12		19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS			B	A																							W	X	Y	Z	

PLS159A

Table 4. PLS Program Table

TERM	C	I 3	I 2	I 1	I 0	B(I) 3	B(I) 2	B(I) 1	B(I) 0	Q(P) 7	Q(P) 6	Q(P) 5	Q(P) 4	Q(P) 3	Q(P) 2	Q(P) 1	Q(P) 0	REMARKS	Q(N) 7	Q(N) 6	Q(N) 5	Q(N) 4	Q(N) 3	Q(N) 2	Q(N) 1	Q(N) 0	B(O) 3	B(O) 2	B(O) 1	B(O) 0	REMARKS
CODE NO. / F/F MODE										•	•	•				A	•		E_B •				E_A •				POLARITY				
0	—	H	—	H	—	—	—	—	—	—	—	—	—	—	—	—	—	A*C	—	—	—	—	—	—	—	H					J = A*C+/B*/E
1	—	—	L	—	L	—	—	—	—	—	—	—	—	—	—	—	—	/B*/E	—	—	—	—	—	—	—	H					
2																															
3	—	H	—	L	—	—	—	—	—	—	—	—	—	—	—	—	—	A*/C	—	—	—	—	—	—	—	L					K = A*/C
4																															
5	—	L	L	L	—	—	—	—	—	—	—	—	—	—	—	—	—	/A*/B*/C	•	•	•	•	•	•	A	•					D = /A*/B*/C+E
6	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	E	•	•	•	•	•	•	A	•					
7																															
8	—	L	L	L	—	—	—	—	—	—	—	—	—	—	—	—	—	/A*/B*/C	—	—	—	—	—	—	H	—					D = /A*/B*/C+E
9	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	E	—	—	—	—	—	—	H	—					
10																															
11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		—	—	O	—	—	—	—	—					T = I
12	—	—	—	—	—	—	—	—	—	—	—	H	—	—	—	—	—		—	O	—	—	—	—	—	—					T = Q_5
13	—	—	—	—	—	—	—	—	—	—	H	H	—	—	—	—	—		O	—	—	—	—	—	—	—					T = Q_5*Q_6
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
F_C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
P_B																															
R_B																															
L_B																															
P_A																															
R_A																															
L_A																															
D3																															
D2																															
D1																															
D0																															
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12		19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS		A	B	C	D																										

PLS159A

PLS159A primer AN15

A D flip-flop may be implemented by first entering an "A" in F/F MODE. Then enter "0" in the row F_C, which will unconditionally enable the 3-State inverter between the J and K inputs. The following logic equation may be implemented as shown in Table 4, term 5

Q1: D=/A * /B * /C+E.

Notice that the entries in term 5, columns $Q(N)_{0\ to\ 7}$ are "A" and "." instead of "H" and "L" as in the case of J/K flip-flops. The entry "A" will cause the fuse connecting to the "K" input to be disconnected and the "J" fuse to be intact. Whereas the entry "." will cause both fuses to be disconnected. This feature enables the user to quickly recognize the mode in which the flip-flops are operating without having to go through the control terms. Some commercially available device programmers in the market may not have the software capability to implement this feature, in which case an "H" and a "–" may be used in place of "A" and "." respectively as shown in Table 4, terms 8 and 9.

Of course, the term F_C may have inputs instead of zeros and dashes, in which case the flip-flop modes are controlled dynamically.

When both the J and K inputs are "1's", the flip-flop will toggle. A simple 3-bit counter may be implemented using only AND terms as shown in Table 4 terms 11, 12 and 13. The logic equations for the three flip-flops are as the following:

Q_5:T=1; (Q_5 toggles unconditionally)

Q_6:T=Q_5; (Q_6 toggles when Q_5=1)

Q_7:T=Q_5 * Q_6; (Q_7 toggles when Q_5 * Q_6=1)

The above equations represent an octal up-counter. However, since the outputs of the flip-flops are inverted, the counting sequence of the outputs is that of a down-counter.

The flip-flops may be asynchronously set and reset by the Control AND terms PA/PB and RA/RB respectively. As shown in Figure 1, PA and RA controls flip-flops F_0 to F_3, while PB and RB control F_4 to F_7.

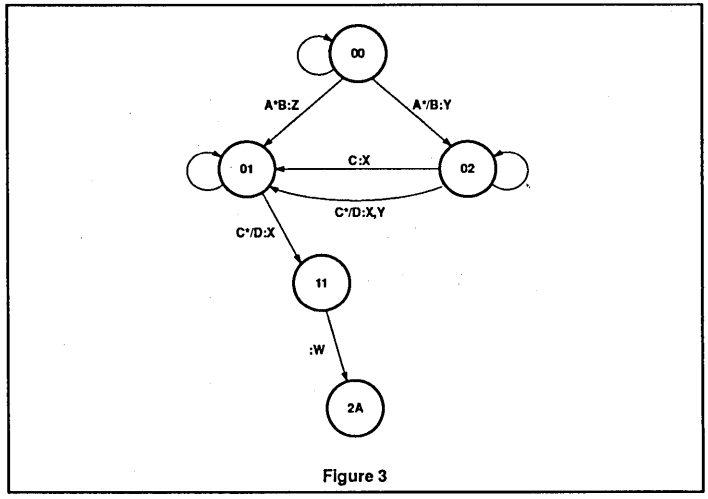


Figure 3

In order to save the number of input pins, the eight flip-flops may be synchronously loaded directly from their own output pins. To use this feature, EA and/or EB must be programmed "A" or "–" so that the output buffers may be disabled before loading. As shown in Figure 1, every flip-flop has an OR/NOR gate the input of which is directly connected to the output pin and the outputs of the OR/NOR are connected to the K and J inputs respectively. This OR/NOR gate inverts the input and feeds it to the flip-flop in a "wire-OR" fashion. Therefore, when loading data directly into the flip-flops from the output pins, caution must be exercised to insure that the inputs from the OR array does not interfere with the data being loaded. For example, if the data being loaded is a "1" on the output pin, the J input will be a "0" and the K input will be a "1". If, at the same time, a "1" is present at the J-input from the OR array, the flip-flop will see "1's" in both J and K inputs. It will toggle as a result. The OR/NOR gates are enabled by the Control AND terms LA and LB. LA controls flip-flops F_0 to F_3 and LB controls F_4 to F_7.

All Control AND terms function and are programmed in the same manner as the other AND terms. The only difference is that the Control AND terms are not connected to the OR array.

The outputs of the flip-flops may be fed back into the AND array as the present state, Q(P). The output of the AND array into the OR array and the inputs to the flip-flops is the next state, Q(N). As an example, Figure 3 is a state machine implemented in a PLS159A as shown in Table 5, terms 0 to 6.

Table 5. PLS Program Table

CODE NO.										F/F MODE									E_B				E_A				POLARITY				
										•	•	•	•	•	•	•	•		•				•				H	H	H	H	
TERM	AND																	REMARKS	(OR)												REMARKS
	C	I				B(I)				Q(P)									Q(N)								B(O)				
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0	3	2	1	0	
0	—	H	H	—	—	—	—	—	—	L	L	L	L	L	L	L	L		L	L	L	L	L	L	L	H	•	•	•	A	
1	—	—	—	H	L	—	—	—	—	L	L	L	L	L	L	L	H		L	L	L	H	L	L	L	H	•	A	•	•	
2	—	—	—	—	—	—	—	—	—	L	L	L	H	L	L	L	H		L	L	H	L	H	L	H	L	A	•	•	•	
3																															
4	—	H	L	—	—	—	—	—	—	L	L	L	L	L	L	L	L		L	L	L	L	L	L	H	L	•	•	A	•	
5	—	—	—	H	—	—	—	—	—	L	L	L	L	L	L	H	L		L	L	L	L	L	L	L	H	•	A	•	•	
6	—	—	—	H	L	—	—	—	—	L	L	L	L	L	L	H	L		L	L	L	L	L	L	L	H	•	A	A	•	
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
F_C																															
P_B																															
R_B																															
L_B																															
P_A																															
R_A																															
L_A																															
D3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
D1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
D0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12		19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS		A	B	C	D																						W	X	Y	Z	

PLS159A

Application Note	
Date of Issue	June 1988
Rev. Date	October 1990
Programmable Logic Devices	

AN18

Schmitt trigger using PLS153 and PLS159A

INTRODUCTION

One of the many features of the PLS153 to 159A series is the availability of individually controlled 3-State I/O pins. Taking advantage of this feature, a Schmitt trigger may be constructed using one input pin, two bidirectional I/O pins and additional components of three resistors. The two threshold voltages, as well as the hysterisis, are determined by the values of the three resistors and the parameters of the PLS153/159A device, which are 1) input threshold voltage, V_{TH}, 2) High output voltage, V_{OH}, and 3) Low output voltage, V_{OL}. The circuit may be simplified if Schmitt function is needed only on Low going High or High going Low, and if the hysterisis and threshold voltages are not important.

DESCRIPTION

A simplified block diagram of a non-inverting Schmitt trigger is shown in Figure 1 where R_1, R_2, and R_1, R_3, form two pairs of voltage dividers one of which get into action at input voltage direction of High going Low and the other Low going High. Assuming that input voltage starts at zero volt, the output voltage is therefore at V_{OL} which causes Q_2 to pull R_3 towards ground. As the input voltage increases, only a fraction of the voltage is impressed upon the input buffer due to the dividing network R_1 and R_3. As soon as the input voltage reaches a point where $V_1=V_{TH}$ (V_{TH}=1.38V typical), the output switches to V_{OH} which, in turn, turns off Q_2 and turns on Q_1. V_1 will jump to a value greater than V_{TH} and Q_1 then pulls the input pin, through R_2, towards V_{OH}, which in turn locks the output to a High state even if the input voltage fluctuates, as long as it does not fluctuate outside of the designed hysterisis. When the input voltage goes from a High to a Low, the Schmitt function repeats itself except that Q_1 and Q_2 reverse their roles.

The triggering voltages, V_H (Low going High) and V_L (High going Low) are:

$$V_H=V_{TH}\,[(R_1+R_3)/R_3] - V_{OL}\,(R_1/R_3);$$

$$V_L=V_{TH}\,[(R_1+R_2)/R_2] - V_{OH}\,(R_1/R_2);$$

where, at room temperature, V_{CC} = 5.0V, I_{OH}/I_{OL}<1mA. V_{TH} is the threshold voltage of the device, typically 1.38V; V_{OL} is the output Low voltage of the device, typically 0.36V at $|I_{OL}|$ <1mA; V_{OH} is the output High voltage of the device, typically 3.8V at $|I_{OH}|$ <1mA.

The implementation of Figure 1 using PLS153/153A is as shown in Table 1, and Figure 2a. A scope photo of the operation of the circuit is shown in the Appendix. The implementation using PLS159A is shown in Table 2 and Figure 2b. In Tables 1 & 2, V_1 is the input pin, V_0 is the output pin, V_2 is the output which pulls down V_1 and V_3 is the output pin that pulls up V_1. The Schmitt output is available at pin B_0 for external use, and is available internally at the input buffers of I_0 and $B(I)_0$. However, there is a propagation delay between the two signals from the I_0 buffer and the $B(I)_0$ buffer.

An inverting Schmitt triggered buffer may be constructed using the same principle. A simple block diagram of such inverter is shown in Figure 3a. The circuit is implemented using H/L programming table as shown in Table 3 for PLS153 and Table 4 for PLS159A. Table 3 is also represented in logic symbols in Figure 3b. If the voltage levels (V_L and V_H) and the hysterisis are not critical, one I/O pin may be used to pull the input pin High and Low. Therefore one I/O pin and a resistor may be saved. The drawback is that the range of V_H and V_L is quite limited. The circuit is as shown in Figure 4.

If Schmitt function is needed only in one direction, one of the resistor/output circuit may be eliminated. The circuit is as shown in Figure 5.

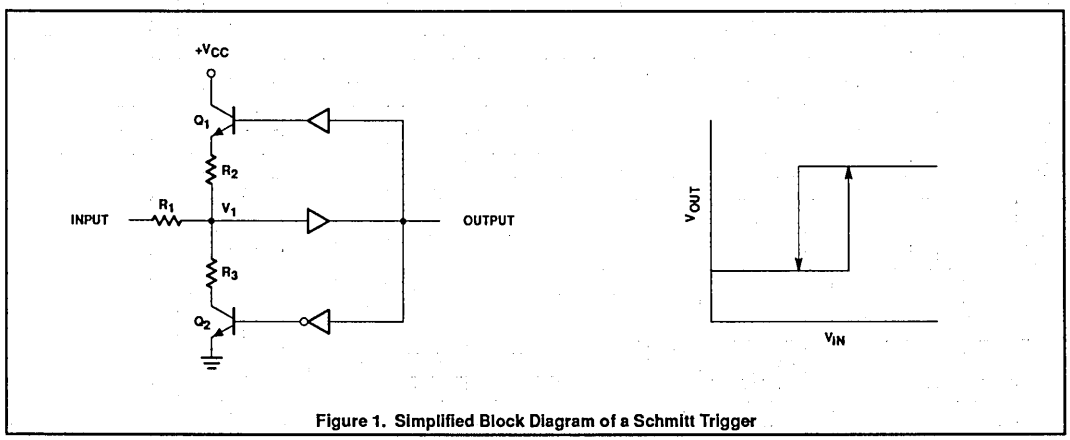
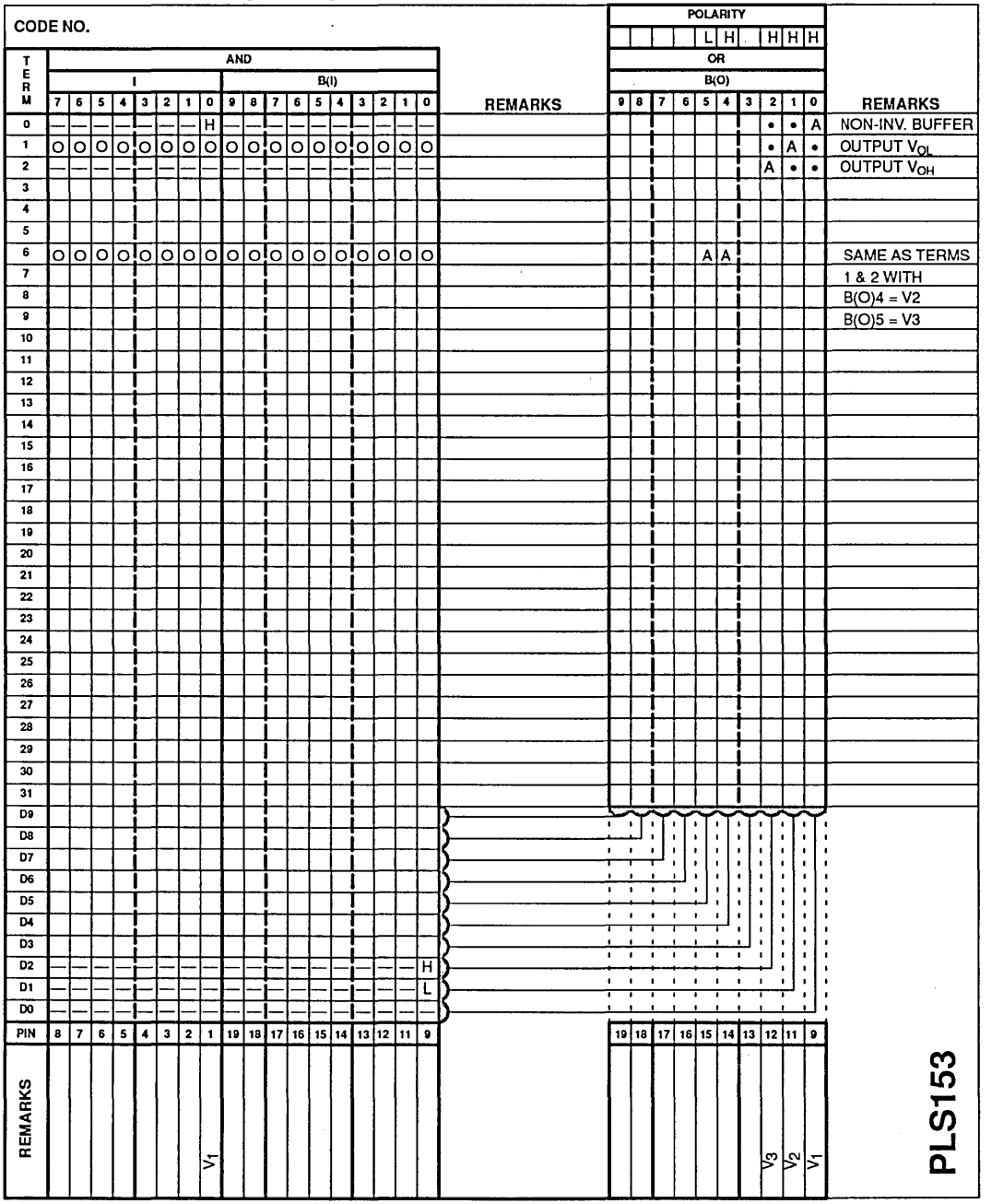


Figure 1. Simplified Block Diagram of a Schmitt Trigger

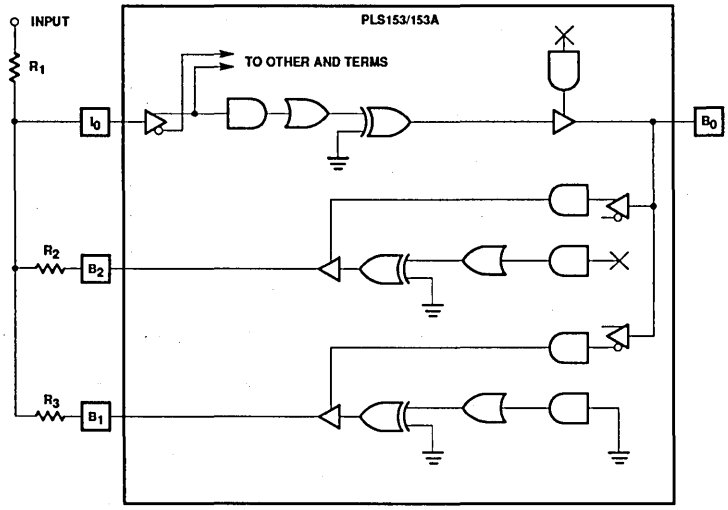
Schmitt trigger using PLS153 and PLS159A AN18

Table 1. PLS153/153A Programming Table

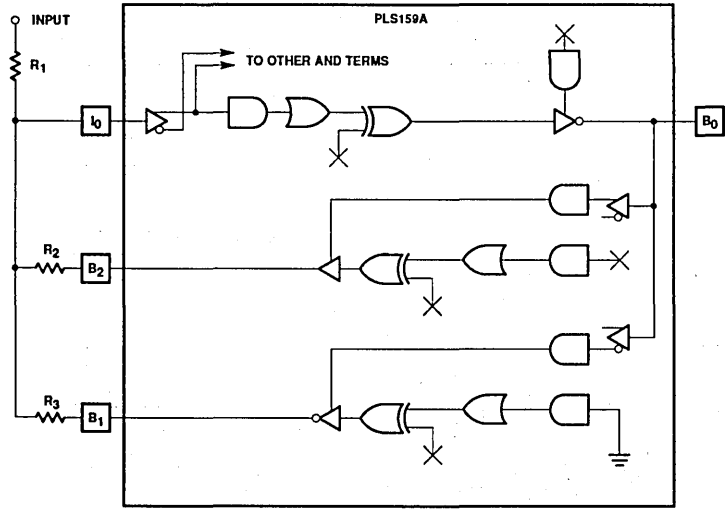


NOTE: Schmitt trigger output may be obtained from both I_0 and $B(I)_0$ to drive the AND-ARRAY.

Schmitt trigger using PLS153 and PLS159A AN18



a. Using PLS153/153A



b. Using PLS159A

Figure 2. Schmitt Trigger

Schmitt trigger using PLS153 and PLS159A | AN18

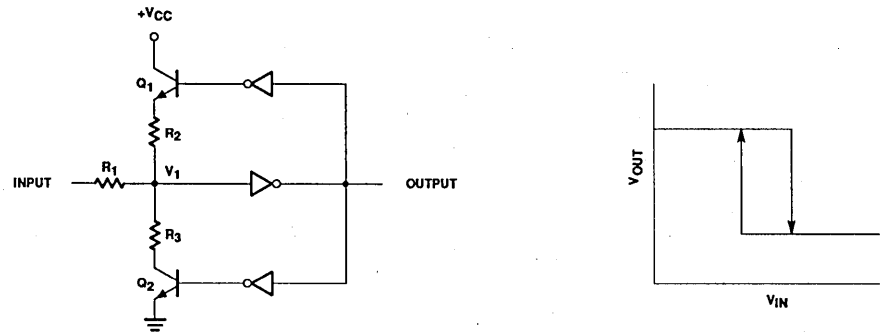
Table 2. PLS159A PLS Programming Table

CODE NO.	F/F MODE	E_B	E_A	POLARITY
				H H H

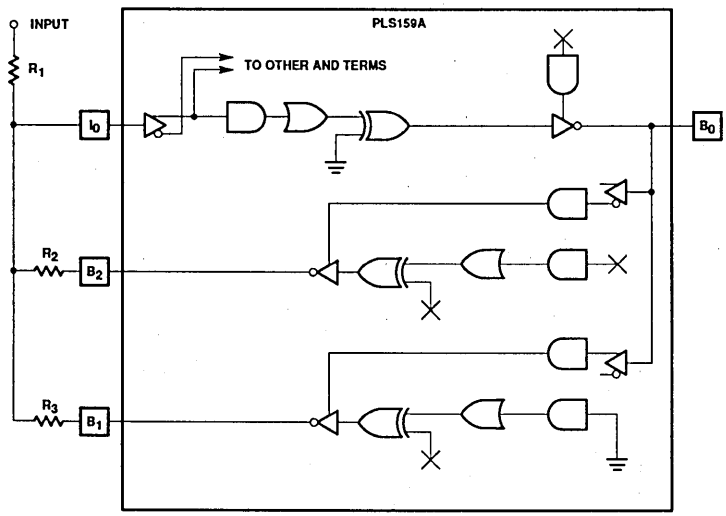
TERM	C	I 3	I 2	I 1	I 0	B(I) 3	B(I) 2	B(I) 1	B(I) 0	Q(P) 7	Q(P) 6	Q(P) 5	Q(P) 4	Q(P) 3	Q(P) 2	Q(P) 1	Q(P) 0	REMARKS	Q(N) 7	Q(N) 6	Q(N) 5	Q(N) 4	Q(N) 3	Q(N) 2	Q(N) 1	Q(N) 0	B(O) 3	B(O) 2	B(O) 1	B(O) 0	REMARKS
0	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—		—	—	—	—	—	—	—	—	•	•	•	A	NON-INV. BUFR
1	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O		—	—	—	—	—	—	—	—	•	•	A	•	OUTPUT V_{OL}
2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		—	—	—	—	—	—	—	—	•	A	•	•	OUTPUT V_{OH}
3																															
4																															
5																															
6																															
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
F_C																															
P_B																															
R_B																															
L_B																															
P_A																															
R_A																															
L_A																															
D3																															
D2	—	—	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—														
D1	—	—	—	—	—	—	—	—	L	—	—	—	—	—	—	—	—														
D0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12		19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS					V_1		V_3	V_2	V_0																						

PLS159A

Schmitt trigger using PLS153 and PLS159A AN18



a. Simplified Block Diagram



b. Using PLS159A

Figure 3. Inverting Schmitt Trigger

Schmitt trigger using PLS153 and PLS159A AN18

Table 3. PLS153/153A Programming Table

CODE NO.

TERM	AND I 7	6	5	4	3	2	1	0	AND B(I) 9	8	7	6	5	4	3	2	1	0	REMARKS	POLARITY L L L	OR B(O) 9	8	7	6	5	4	3	2	1	0	REMARKS
0	—	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—											•	•	A	INV. BUFFER
1	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o											•	A	•	OUTPUT V_{OH}
2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—											A	•	•	OUTPUT V_{OL}
3																																
4																																
5																																
6																																
7																																
8																																
9																																
10																																
11																																
12																																
13																																
14																																
15																																
16																																
17																																
18																																
19																																
20																																
21																																
22																																
23																																
24																																
25																																
26																																
27																																
28																																
29																																
30																																
31																																
D9																																
D8																																
D7																																
D6																																
D5																																
D4																																
D3																																
D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H														
D1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L														
D0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
PIN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9			19	18	17	16	15	14	13	12	11	9		
REMARKS								V_1																				V_3	V_2	V_0		

PLS153

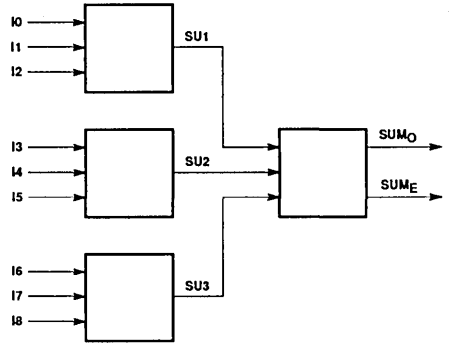
Schmitt trigger using PLS153 and PLS159A AN18

Table 4. PLS159A PLS Programming Table

CODE NO.																	F/F MODE	REMARKS	E_B				E_A				POLARITY			REMARKS
																											H	H	L	
TERM	AND																		(OR)											
	C	I				B(I)				Q(P)									Q(N)								B(O)				
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0	3	2	1	0	
0	—	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—		—	—	—	—	—	—	—	—	•	•	•	A	INV. BUFFER
1	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O		—	—	—	—	—	—	—	—	•	•	A	•	OUTPUT V_{OL}
2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		—	—	—	—	—	—	—	—	•	A	•	•	OUTPUT V_{OH}
3																															
4																															
5																															
6																															
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
F_C																															
P_B																															
R_B																															
L_B																															
P_A																															
R_A																															
L_A																															
D3																															
D2	—	—	—	—	—	—	—	—	L	—	—	—	—	—	—	—	—														
D1	—	—	—	—	—	—	—	—	H	—	—	—	—	—	—	—	—														
D0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12		19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS					V_1		V_3	V_2	V_0																						

PLS159A

9-Bit parity generator/checker with PLS153/153A AN21



I_0	I_1	I_2	SU1
0	0	0	0
0	0	1	1
0	1	1	0
0	1	0	1
1	0	1	0
1	0	0	1
1	1	0	0
1	1	1	1

I_3	I_4	I_5	SU2
0	0	0	0
0	1	1	0
0	0	1	1
0	1	0	1
1	0	1	0
1	1	0	0
1	0	0	1
1	1	1	1

I_6	I_7	I_8	SU3
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0
0	0	1	1
0	1	0	1
1	0	0	1
1	1	1	1

SU1	SU2	SU3	SUM_O	SUM_E
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Figure 1. Block Diagram of 9-Bit Parity Generator/Checker

```
 ******************** PARGEN ********************
######################## P I N   L I S T ########################

   LABEL     ** FNC **PIN --------- PIN** FNC **     LABEL
I0           ** I   **  1-|        |-20 ** +5V **VCC
I1           ** I   **  2-|        |-19 ** /O  **SUME
I2           ** I   **  3-|        |-18 ** O   **SUMO
I3           ** I   **  4-|   P    |-17 ** B   **N/C
I4           ** I   **  5-|   L    |-16 ** B   **N/C
I5           ** I   **  6-|   S    |-15 ** B   **N/C
I6           ** I   **  7-|   1    |-14 ** B   **N/C
I7           ** I   **  8-|   5    |-13 ** O   **SU3
I8           ** I   **  9-|   3    |-12 ** O   **SU2
GND          ** 0V  ** 10-|        |-11 ** O   **SU1
                           ---------
```

Figure 2. Pin List

9-Bit parity generator/checker with PLS153/153A AN21

```
 ******************** PARGEN ********************
@DEVICE TYPE
PLS153
@DRAWING
****************************** PARITY GENERATOR/CHECKER
@REVISION
****************************** REV. -
@DATE
****************************** xx/xx/xxxx
@SYMBOL
****************************** FILE ID: PARGEN
@COMPANY
****************************** SIGNETICS
@NAME
@DESCRIPTION
*********************************************************************
*   This circuit is a 9-bit parity generator/checker commonly used  *
*   for error detection in high speed data transmission/retrieval.  *
*   The odd parity output (SUMO) is high when the sum of the data   *
*   bits is odd. Otherwise it is low.                               *
*   The even parity output (SUME) is high when the sum of the data  *
*   bits is even. It is low otherwise.                              *
*********************************************************************
@COMMON PRODUCT TERM
@I/O DIRECTION
"
*********************************************************************
*   SU1, SU2 and SU3 are outputs which are defined in the PIN LIST  *
*   and therefore they don't need to be defined here again.         *
*********************************************************************
"
@OUTPUT POLARITY
"
*********************************************************************
*   The output polarities of different outputs are defined in the   *
*   PIN LIST. They don't have to be defined again here.             *
*********************************************************************
"
@LOGIC EQUATION
"
*********************************************************************
*   SU1, SU2, and SU3 are intermediate terms                        *
*********************************************************************
                    TRUTH TABLE
         INPUTS                  OUTPUTS
    ----------------      ----------------------
    SU3   SU2   SU1       SUMO     SUME = /SUMO
     I8    I7    I6        SU3
     I5    I4    I3        SU2
     I2    I1    I0        SU1

      0     0     0         0         1
      0     0     1         1         0
      0     1     0         1         0
      0     1     1         0         1
      1     0     0         1         0
      1     0     1         0         1
      1     1     0         0         1
      1     1     1         1         0
"

SU1 = /I2 * /I1 *  I0 + /I2 *  I1 * /I0 +
       I2 * /I1 * /I0 +  I2 *  I1 *  I0 ;
SU2 = /I5 * /I4 *  I3 + /I5 *  I4 * /I3 +
       I5 * /I4 * /I3 *  I5 *  I4 *  I3 ;

SU3 = /I8 * /I7 *  I6 + /I8 *  I7 * /I6 +
       I8 * /I7 * /I6 +  I8 *  I7 *  I6 ;

SUMO = /SU1 * /SU2 *  SU3 + /SU1 *  SU2 * /SU3 +
        SU1 * /SU2 * /SU3 +  SU1 *  SU2 *  SU3 ;

SUME = /(/SU1 * /SU2 *  SU3 + /SU1 *  SU2 * /SU3 +
          SU1 * /SU2 * /SU3 +  SU1 *  SU2 *  SU3) ;
```

Figure 3. AMAZE Implementation of the Parity Generator/Checker Circuit

9-Bit parity generator/checker with PLS153/153A AN21

Table 1. Programming Table

```
 ******************** PARGEN ********************
Cust/Project -
Date          - ****************************** xx/xx/xxxx
Rev/I. D.     _ ****************************** REV. -

PLS153                                    |     POLARITY      |

---                                        --------------------
 T |                                      |L:H:H:H:H:H:H:H:H:H|
 E |-------------------------------------------------------------
 R |       I        |        B(i)        |        B(o)        |
 M |-------------------------------------------------------------
---|7_6_5_4_3_2_1_0|9_8_7_6_5_4_3_2_1_0|9_8_7_6_5_4_3_2_1_0|
  0|- - - -,- L L H|- -,- - - -,- - - -|. .,A A A A,. . A A|
  1|- - - -,- L H L|- -,- - - -,- - - -|. .,A A A A,. . A A|
  2|- - - -,- H L L|- -,- - - -,- - - -|. .,A A A A,. . A A|
  3|- - - -,- H H H|- -,- - - -,- - - -|. .,A A A A,. . A A|
  4|- - L L,H - - -|- -,- - - -,- - - -|. .,A A A A,. A . A|
  5|- - L H,L - - -|- -,- - - -,- - - -|. .,A A A A,. A . A|
  6|- - H L,L - - -|- -,- - - -,- - - -|. .,A A A A,. A . A|
  7|- - H H,H - - -|- -,- - - -,- - - -|. .,A A A A,. A . A|
  8|L H - -,- - - -|- -,- - - -,- - - L|. .,A A A A,A . . A|
  9|H L - -,- - - -|- -,- - - -,- - - L|. .,A A A A,A . . A|
 10|L L - -,- - - -|- -,- - - -,- - - H|. .,A A A A,A . . A|
 11|H H - -,- - - -|- -,- - - -,- - - H|. .,A A A A,A . . A|
 12|- - - -,- - - -|- -,- - - -,H L L -|A A,A A A A,. . . A|
 13|- - - -,- - - -|- -,- - - -,L H L -|A A,A A A A,. . . A|
 14|- - - -,- - - -|- -,- - - -,L L H -|A A,A A A A,. . . A|
 15|- - - -,- - - -|- -,- - - -,H H H -|A A,A A A A,. . . A|
 16|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 17|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 18|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 19|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 20|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 21|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 22|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 23|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 24|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 25|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 26|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 27|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 28|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 29|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 30|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 31|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|A A,A A A A,A A A A|
 D9|- - - -,- - - -|- -,- - - -,- - - -|                   .
 D8|- - - -,- - - -|- -,- - - -,- - - -|                   .
 D7|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|                   .
 D6|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|                   .
 D5|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|                   .
 D4|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|                   .
 D3|- - - -,- - - -|- -,- - - -,- - - -|                   .
 D2|- - - -,- - - -|- -,- - - -,- - - -|                   .
 D1|- - - -,- - - -|- -,- - - -,- - - -|                   .
 D0|0 0 0 0,0 0 0 0|0 0,0 0 0 0,0 0 0 0|                   .

   I I I I I I I I S S N N N N S S S I S S N N N N S S S I
   7 6 5 4 3 2 1 0 U U / / / / U U U 8 U U / / / / U U U 8
                   M M C C C C 3 2 1   M M C C C C 3 2 1
                   E O                 E O
```

9-Bit parity generator/checker with PLS153/153A AN21

```
"
*************************************************************************
*   This is a test pattern for the 9-bit parity generator/checker       *
*   circuit. The simulator will use this file as an input to            *
*   simulate the logical function.                                      *
*************************************************************************

"         SS             EXPECTED
"         UU    SSS      OUTPUTS
"IIIIIIII MMBBBBUUUI      BBBBB
"76543210 E076543218      98321
 LLLLLLLL /////////L     "HLLLL
 HLHHLHLL /////////H     "LHLLH
 LHHLLHHL /////////H     "LHLHL
 HHLHLLHL /////////L     "HLLHH
 LLHLHHLH /////////H     "LHHLL
 HLLHHLLH /////////L     "HLHLH
 LHLLHLHH /////////L     "HLHHL
 HHHHHHHH /////////H     "LHHHH
QUIT
```

a. Input Pattern PARGEN.TST

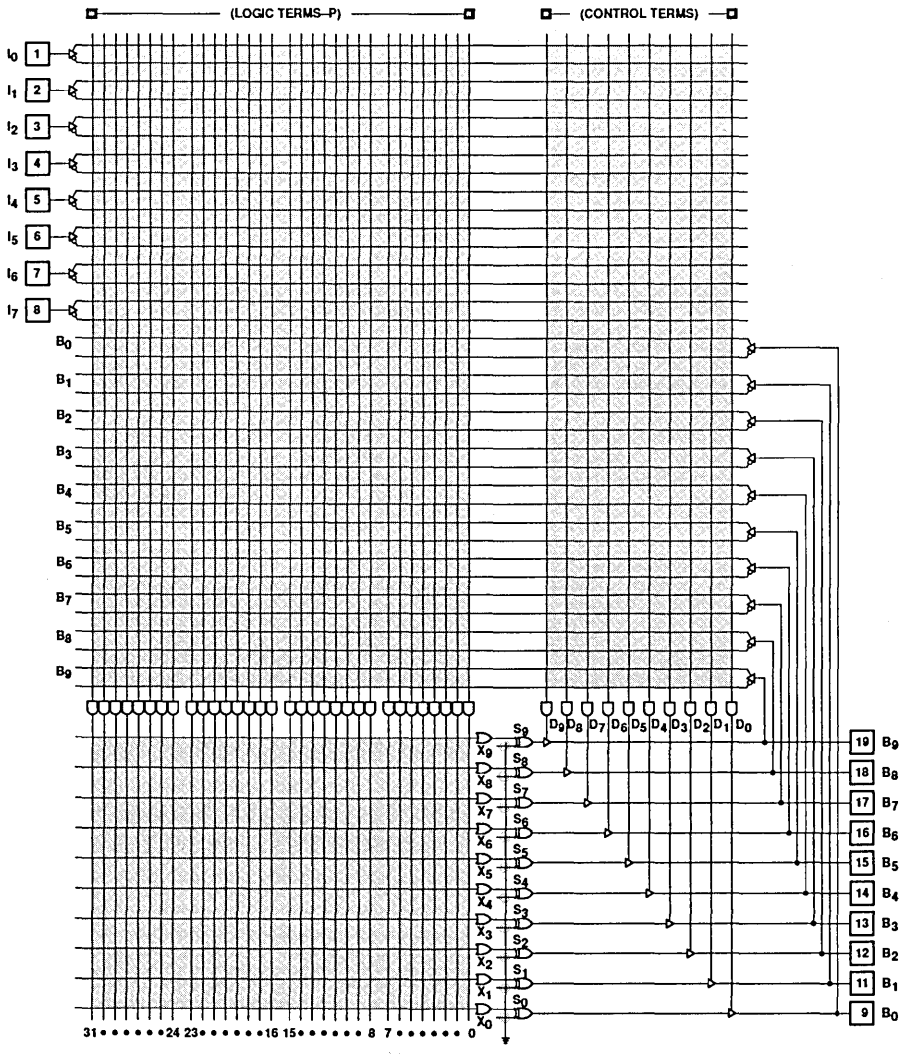
```
PLS153   A:pargen.STD
" This file is the result of logic simulation of the parity generator/checker
" circuit. The inputs are read from input file PARGEN.TST
"
"  INPUTS  <=B(I/O)=>   TRACE TERMS
" 76543210 9876543210
"
  00000000 HL....LLL0 ;
  10110100 LH....LLH1 ;
  01100110 LH....LHL1 ;
  11010010 HL....LHH0 ;
  00101101 LH....HLL1 ;
  10011001 HL....HLH0 ;
  01001011 HL....HHL0 ;
  11111111 LH....HHH1 ;
"
: -------- ----------   I/O CONTROL LINES
"          OOIIIIOOOI   DESIGNATED I/O USAGE
"          OOIIIIOOOI   ACTUAL I/O USAGE
"
" PIN LIST...
" 08 07 06 05 04 03 02 01 19 18 17 16 15 14 13 12 11 09 ;
```

b. Output File from SIMULATOR

Figure 4. Test Vectors

9-Bit parity generator/checker with PLS153/153A AN21

APPENDIX A



NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ◌ Programmable connection.

Figure A-1. PLA Logic Diagram

Application Note	
Date of Issue	June 1988
Rev. Date	
Programmable Logic Devices	

AN23
PLS168/168A primer

INTRODUCTION

THE PLS168/168A is a bipolar Programmable Logic Sequencer as shown in Figure 1, which consists of 12 inputs, a 48 product term PLA and 14 R/S flip-flops. Out of the 14 flip-flops, six are buried State Registers (P_4-P_9), four Output Registers (F_0-F_3), and four Dual-purpose Registers (P_0-P_3), which may be used as Output or State Registers. All flip-flops are positive edge-triggered. They are preset to "1" at power-up, or may be asynchronously set to "1" by an optional PR/OE pin, which may be programmed either as a preset pin or as an Output Enable pin. Additional features includes the Complement Array and diagnostics features.

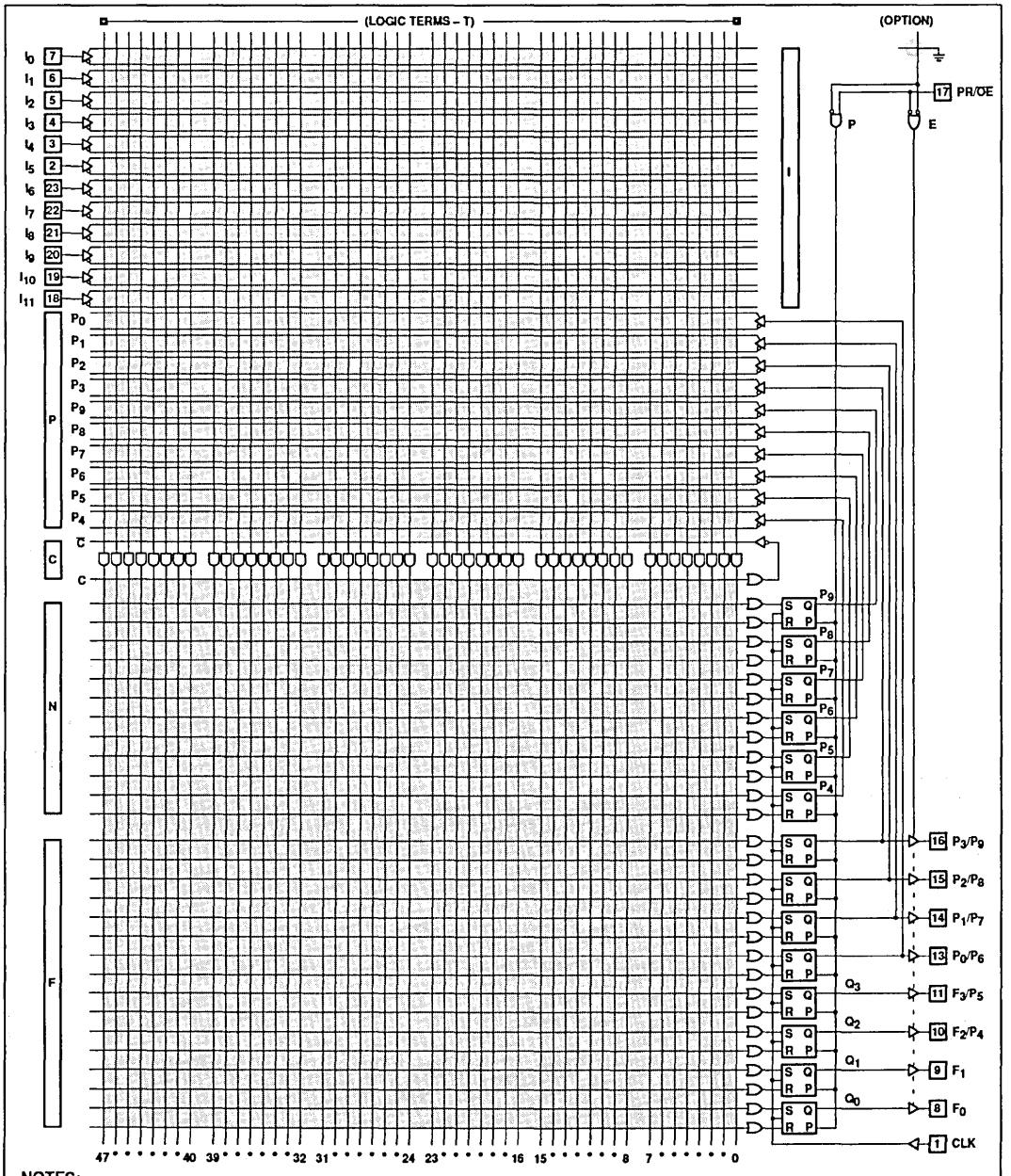
ARCHITECTURE

As shown in Figure 2, the device is organized as a decoding AND-OR network which drives a set of registers some of which, in turn, feedbacks to the AND/OR decoder while the rest serve as outputs. Outputs P_0 to P_3 may be programmed to feedback to the AND/OR decoder as State Registers and, at the same time, used as outputs. The user now can design a 10-bit state machine without external wiring. The AND/OR array is the classical PLA structure in which the outputs of all the AND gates can be programmed to drive all the OR gates. The schematic diagram of the AND-OR array is shown in Figure 3. This structure provides the user a very structured design methodology which can be automated by CAD tools, such as Signetics AMAZE software package. The output of the PLA is in the form of sum-of-products which, together with the RS flip-flops, is the ideal structure for implementation of state machines. (Refer to Appendix A for a brief description of synchronous finite state machines.)

Design Tools

A direct approach to implement a design using the PLS168/168A is the H/L table supplied in the data sheet as shown in Table 1. The table is organized according to input and output of the PLA decoding network. The lefthand side of the table represent the inputs to the AND-array, which includes input from input pins and present state information from the feedback buffers which feedback the contents of the State Register. The righthand side of the table represents the output of the OR-array, which drives the State and Output Registers as the next state and output. Each column in the lefthand side of the table represents an input buffer, which may be inverting, non-inverting, disconnected or unprogrammed. Each column in the righthand side of the table represents a pair of outputs to the flip-flops, which may be set, reset, disconnected, or unprogrammed. The programming symbols are H, L, —, and 0. (See Figure 4 for details.) For inputs buffers, "H" means that the non-inverting buffer is connected, "L" means that the inverting buffer is connected, "—" means that both inverting and non-inverting buffers are disconnected, and "0" means that both inverting and non-inverting buffers are connected which causes that particular AND-term to be unconditionally Low. On the output side of the table, "H" means that the particular AND-term is connected to the OR-term on the "S" input of the particular flip-flop, "L" means that the AND-term is connected to the "R" side, "—" means that the AND-term is not connected to the flip-flop at all, and "0" means that the AND-term is connected to both the "S" and "R" sides. More details of the symbols and their meanings are shown in Appendix B. Each row in the table represents an AND-term. There are 48 AND-terms in the device. Therefore, there are 48 rows in the table. An example of implementing a transition from one state to another is shown in Figure 4a. The state diagram can be implemented by the PLS168 as shown in Figure 4b. The state diagram is translated into H/L format as shown in Figure 4c. The first column on the lefthand side of the table is for the Complement Array which will be discussed in detail in the next section.

PLS168/168A primer AN23



NOTES:
1. All programmed "AND" gate locations are pulled to logic "1".
2. All programmed "OR" gate locations are pulled to logic "0".
3. ◎ Programmable connection.

Figure 1. Logic Diagram of PLS168/168A

PLS168/168A primer

AN23

Table 1. PLS168/168A Programming Table

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # CF (XXXX)

CUSTOMER SYMBOLIZED PART # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE _____

REV _____

DATE _____

AND

C_n	
INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

I_m, P_s	
INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

OR

N_s, Fr	
INACTIVE	0
SET	H
RESET	L
NO CHANGE	—

OPTION

P/E	
PRESET	H
$\overline{OE}$	L

TERM	C_n	AND: INPUT (I_m) 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) (P_x) 9 8 7 6 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (N_s) (P_x) 9 8 7 6 5 4 3 2 1 0	OPTION (P/E) OR: OUTPUT (Fr) 3 2 1 0 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		18 19 20 21 22 23 2 3 4 5 6 7				16 15 14 13 11 10 9 8
VARIABLE NAME						

PLS168/168A primer AN23

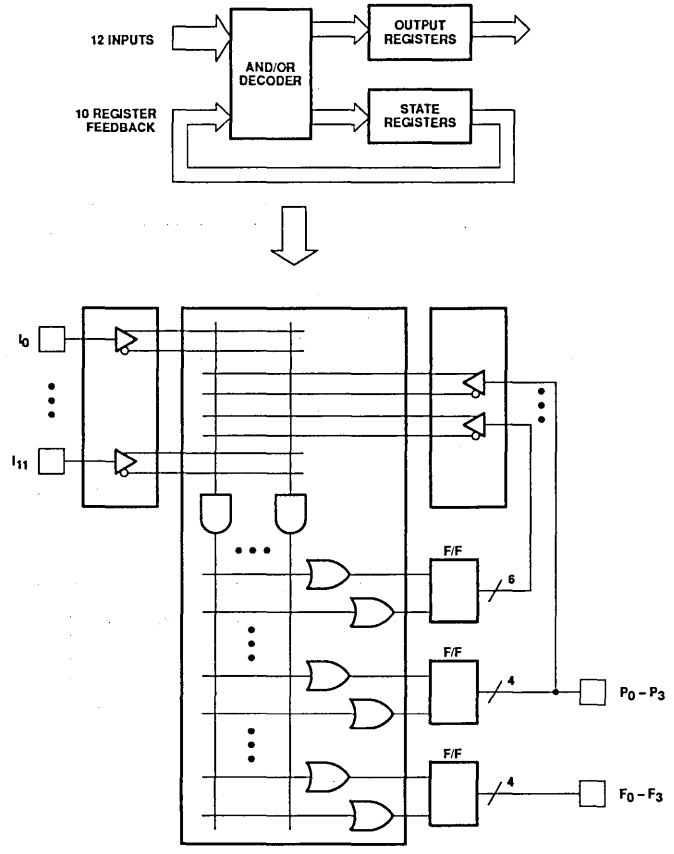


Figure 2. The Architecture of PLS168/168A

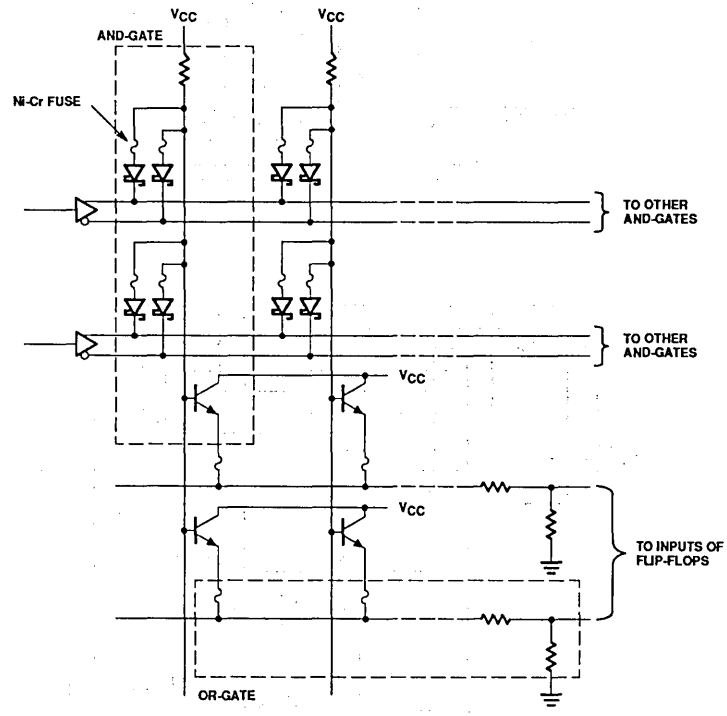
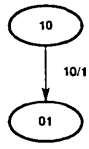
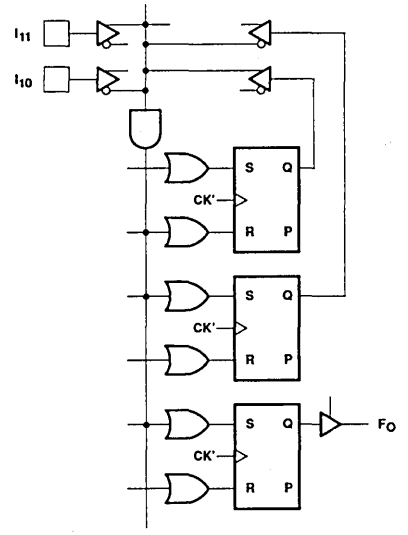


Figure 3. Schematic Diagram of AND-OR Array



a. State Diagram



b. Implementation of State Diagram (a) with PLS168

		AND																					OPTION (P/E) OR										
TERM	C_n	INPUT											PRESENT STATE										NEXT STATE					OUTPUT					
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0
00		–	H	L	–	–	–	–	–	–	–	–	–	–	–	–	–	H	L	–	–	–	–	–	–	–	–	L	H	–	–	–	–	H	–	–	–
01																																					
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																	16	15	14	13	11	10	9	8
NAME		IN1	IN0																															OUT0			

c. PLS168 Programming Table

Figure 4. Implementing State Machine with PLS168

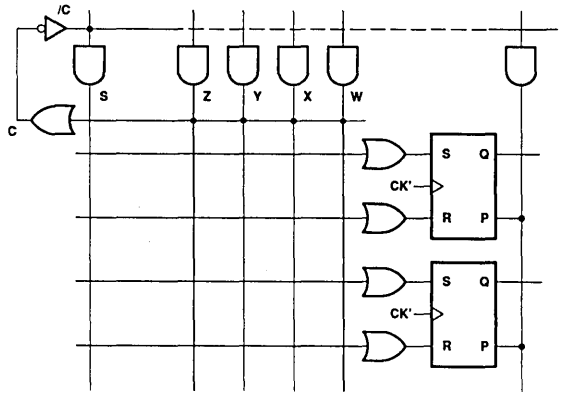
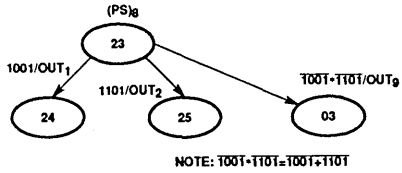


Figure 5. Logic Diagram of Complement Array

Complement Array

An additional feature is the Complement Array, which is often used to provide escape vectors in case the state machines get into undefined states during power-up or a timing violation due to asynchronous inputs. A logic diagram of the Complement Array is shown in Figure 5. The output of the Complement Array is normally Low when one or more AND-terms are High. If all of the AND-terms are Low, then the output of the Complement Array will be High. In this example, if each AND-term is a decoder for a particular state and input combination, and if the circuit gets into an undefined state, none of the AND-terms will be High. Therefore, the output /C will be High, which will then enable the AND-term S which in turn may be used to reset all registers to Low or High as pre-defined. The state machine thus escapes from being in an undefined state by using the Complement Array and one AND-term.

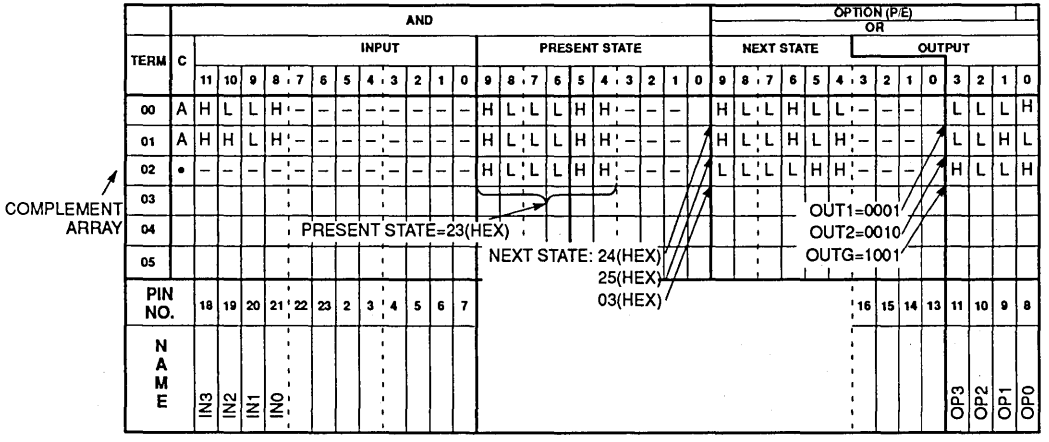
Without the Complement Array an alternate way of escaping from being in an undefined state is by defining all possible states which are not being defined. This method may require quite a few AND-terms depending on the design. Another application for the Complement Array is illustrated by the following example. As shown in Figure 6, when the machine is in state 23, if input vector equals 1001, it will go to next state 24. If the input is 1101, then go to state 25. But if the input is neither 1001 nor 1101, then go to state 03. It takes only two terms to implement the first two transition vectors. To implement the third vector "go to state 03 if input is neither 1001 nor 1101", the Complement Array accepts the outputs of the first two AND-terms as inputs. If the input vector is neither 1001 nor 1101, then both terms will be Low, which causes the output of the Complement Array (/C) to be High. A third AND-term is used to AND state 24 and /C together to set the registers to state 03. The State Diagram is translated into AMAZE syntax as shown in Figure 6b, where all vectors are in square brackets and the Complement Array is represented by the ELSE statement. The State diagram Figure 6a can also be expressed in the format of a program table as shown in Figure 6c. The complement array may be used to exit from different present states to different next states. It can be used many times in one state machine design as shown in Figures 7a, b, and c where the state diagram is implemented using the AMAZE state equation syntex and the H/L format.



a. State Diagram

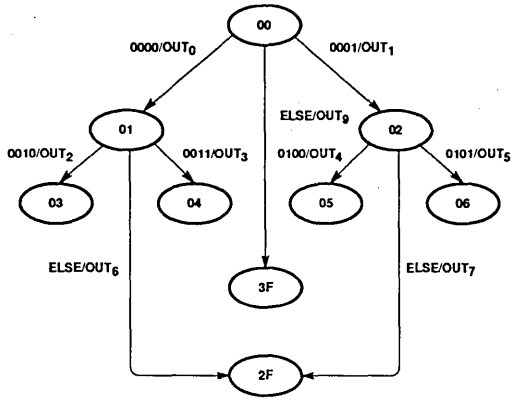
```
WHILE [23]
    IF [1001] THEN [24] WITH
[OUT1]
    IF [1101] THEN [25] WITH
[OUT2]
    ELSE: [03] WITH [OUT9]
```

b. AMAZE State Equation Syntax



c. H/L Format

Figure 6. Application of Complement Array



a. State Diagram

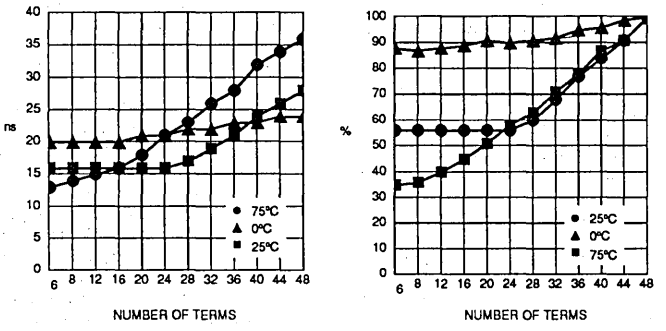
```
WHILE [00]
    IF [0000] THEN [01] WITH [OUT0]
    IF [0001] THEN [02] WITH [OUT1]
    ELSE: [3F] WITH [OUT9]
WHILE [01]
    IF [0010] THEN [03] WITH [OUT2]
    IF [0011] THEN [04] WITH [OUT3]
    ELSE: [2F] WITH [OUT6]
WHILE [02]
    IF [0100] THEN [05] WITH [OUT4]
    IF [0101] THEN [06] WITH [OUT5]
    ELSE: [2F] WITH [OUT7]
```

b. AMAZE State Equation

		AND																				OPTION (PR/OE) OR											
TERM	C	INPUT												PRESENT STATE								NEXT STATE						OUTPUT					
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0
00	A	L	L	L	L	–	–	–	–	–	–	–	–	L	L	L	L	L	L	–	–	–	–	L	L	L	L	L	H	–	–	–	–	L	L	L	L
01	A	L	L	L	H	–	–	–	–	–	–	–	–	L	L	L	L	L	L	–	–	–	–	L	L	L	L	H	L	–	–	–	–	L	L	L	H
02	•	–	–	–	–	–	–	–	–	–	–	–	–	L	L	L	L	L	L	–	–	–	–	H	H	H	H	H	H	–	–	–	–	H	L	L	H
03																																					
04	A	L	L	H	L	–	–	–	–	–	–	–	–	L	L	L	L	L	H	–	–	–	–	L	L	L	L	H	H	–	–	–	–	L	L	H	L
05	A	L	L	H	H	–	–	–	–	–	–	–	–	L	L	L	L	L	H	–	–	–	–	L	L	L	H	L	L	–	–	–	–	L	L	H	H
06	•	–	–	–	–	–	–	–	–	–	–	–	–	L	L	L	L	L	H	–	–	–	–	H	L	H	H	H	H	–	–	–	–	L	H	H	L
07																																					
08	A	L	H	L	L	–	–	–	–	–	–	–	–	L	L	L	L	H	L	–	–	–	–	L	L	L	H	L	H	–	–	–	–	L	H	L	L
09	A	L	H	L	H	–	–	–	–	–	–	–	–	L	L	L	L	H	L	–	–	–	–	L	L	L	H	H	L	–	–	–	–	L	H	L	H
10	•	–	–	–	–	–	–	–	–	–	–	–	–	L	L	L	L	H	L	–	–	–	–	H	L	H	H	H	H	–	–	–	–	L	H	H	H
11																																					
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																16	15	14	13	11	10	9	8
NAME		IN3	IN2	IN1	IN0																												OP3	OP2	OP1	OP0

c. PLS168 Programming Table

Figure 7. Applications of Complement Array



a. PLS168 Setup Time vs. P-Term Loading

b. PLS168 Setup Time vs. P-Term Loading

Figure 8

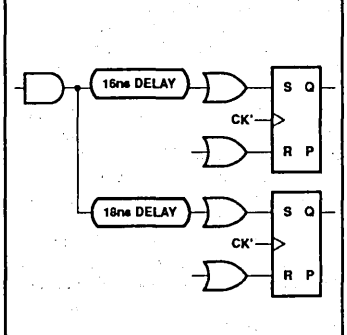


Figure 9. Difference in Propagation Delay Due to Different P-Term Loading

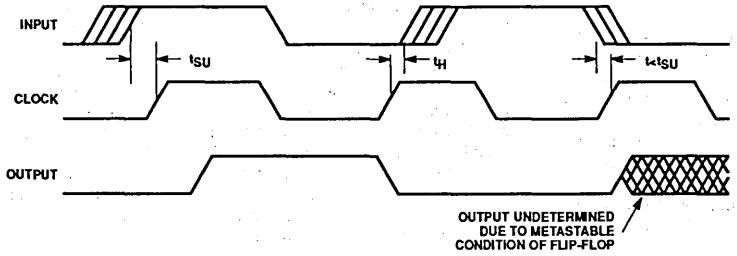


Figure 10. Change of t_{CKO} Due to Metastable Condition of Flip-Flop

PLS168/168A primer AN23

Optional Preset/Output Enable

The PR/OE pin provides the user with the option of either using that pin to control the 3-State output buffers of the Output Registers, or have that pin to asynchronously preset all registers to High. The purpose of the preset function is to provide the system a way to set the PLS168 to a known state, all Highs. The output enable function are sometimes used where the state machine is connected to a bus which is shared by other output circuits. It is also used during power-up sequence to keep the PLS168 from sending power glitches to other circuits which it drives. By programming the PR/OE pin to control the 3-State output buffers, the preset function is permanently disabled. By programming the PR/OE pin to control the asynchronous preset of the registers, the output buffers are permanently enabled. While using the preset function to asynchronously preset the register, if a rising edge of the clock occurs while the preset input is High, the registers will remain preset. Normal flip-flop operation will resume only after the preset input is Low and the rising edge of the next clock. Setting the registers to a predefined pattern other than all Highs may be accomplished by using a dedicated p-term, which is activated by an input pin which will also inhibit all other p-terms which are being used. The inhibiting of other p-terms eliminates the problem of undetermined state of an RS flip-flop caused by having Highs on both R and S inputs.

Diagnostic Features

In debugging a state machine, sometimes it is necessary to know what is the content of the state register. The buried State Register may be read by applying +10V on I_0, which will cause the contents of register bits P_4 to P_5, P_6 to P_9 to be displayed on output pins F_2 to F_3 and P_0 and P_3 respectively. While the device can handle the +10V on pin I_0, prolonged and continuous use will cause the chip to heat up since more power is being dissipated at +10V. To facilitate more expedient functional tests, synchronous preset vectors as described above may be used to set the State Register to different states without having to go through the entire sequence.

Timing Requirements

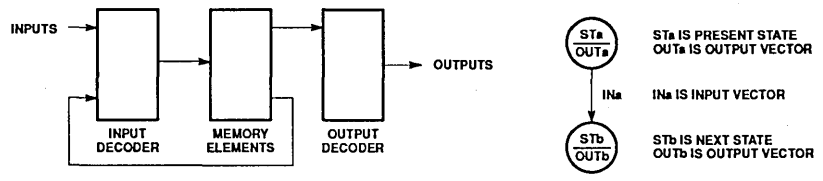
Since the PLS168 is intended to be a synchronous finite state machine, the inputs are expected to be synchronous to the clock and set-up and hold time requirements are expected to be met. In general, the set-up time requirement is measured at its worst case as having the entire AND-array connected to the OR-term being measured and there is only one active AND-term to drive the entire line. The set-up time decreases from there as less p-terms are used. This is due to the capacitance of the unused AND-terms being removed from the line. Figure 8a shows the typical set-up time requirement of a PLS168A device. Figure 8b shows the normalized set-up time as a percentage of the worst case, which is with 48 terms connected. In a typical state machine design, some flip-flops will change states more frequently than others. Those that change more frequently will have more p-term loading on its OR gates than those that change states less frequently. The different loadings on the OR-terms cause different delay on the inputs of the flip-flops as shown in Figure 9. If an input fails to meet the set-up time specification, it is possible that the resultant of the input change gets to one set of flip-flops before the rising edge of the clock while it gets to other flip-flops during or after the clock's rising edge. The result is that some flip-flops have changed states and some have not, or some get into metastable condition as shown in Figure 10. The state machine is now either out of sequence or is in an undefined state. This problem often occurs with asynchronous inputs which is generated totally independent of the clock on the system. A common remedy for the problem of asynchronous inputs is to use latches or flip-flops to catch the input and then synchronously feed it to the state machine. This minimizes the problem with the different propagation delays due to different p-term loading. But there is still a finite probability that the external latches or flip-flops will get into metastable condition, which may be propagated into the state machine. Nevertheless, the window for the flip-flops in state machine to get into undefined states or metastable condition is narrowed by a great extent.

APPENDIX A

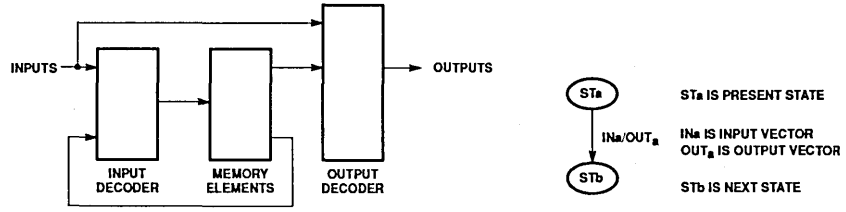
INTRODUCTION TO STATE MACHINE

A state machine is a synchronous sequential circuit which interprets inputs and generates outputs in accordance with a predetermined logic sequence. It is analogous to running a computer program with a computer. The state machine, with its sequence coded in hardware, can run much faster than a computer running the sequence in software. Therefore, it is often used in controller applications where speed is important.

Generally, state machines may be classified as Mealy or Moore machines as shown in Figures 1a and 1b. The fundamental difference of the two types are: the output of a Moore machine is a dependent of only the state of the memory elements whereas the output of a Mealy machine is a dependent of both the state of the memory elements and the inputs to the state machine. The figures also show graphic representations of the logic sequence in the form of state diagram in which the bubbles represent state vectors, and the arrows represent transitions from present states to next states.



Appendix A-1. Moore Machine Model



Appendix A-2. Mealy Machine Model

APPENDIX B

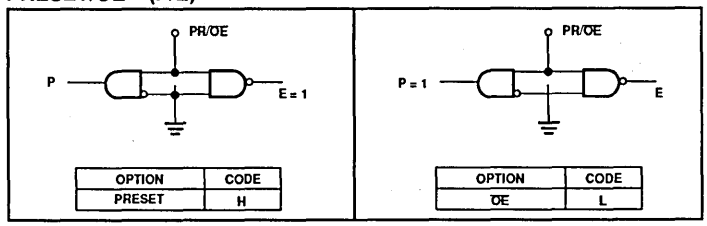
LOGIC PROGRAMMING

The PLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term T_n, is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

PRESET/$\overline{OE}$ – (P/E)

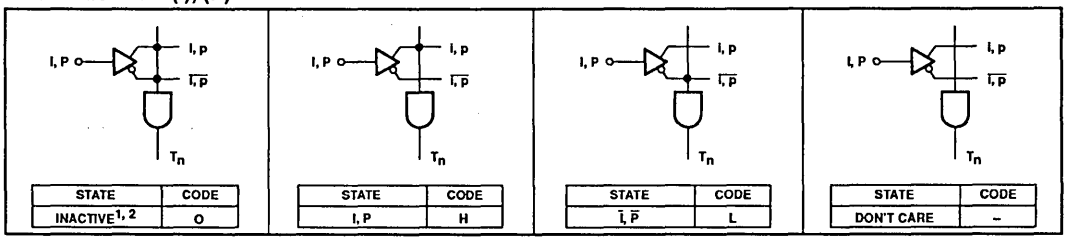


OPTION	CODE
PRESET	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING:
The PLS168/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY – (I), (P)



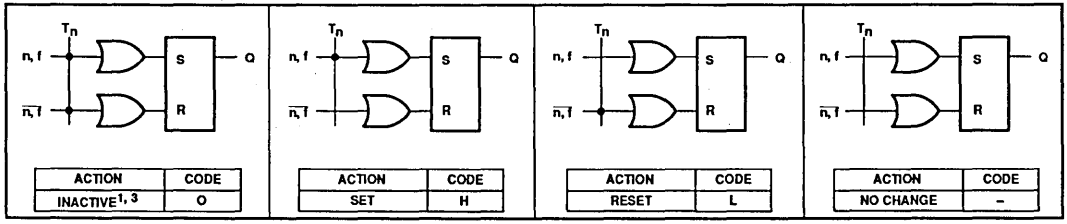
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I, P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



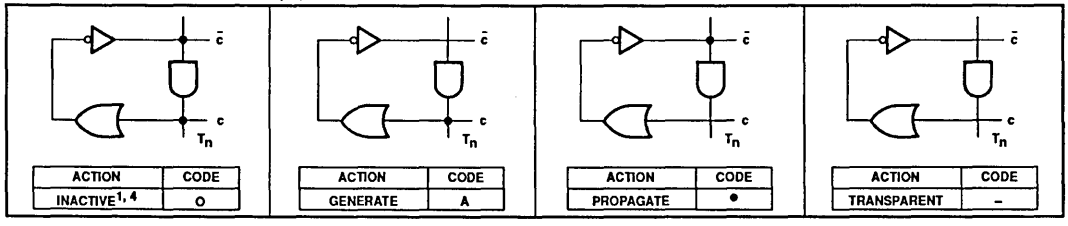
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I or P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Application Note	
Date of Issue	June 1988
Rev. Date	
Programmable Logic Devices	

AN24

PLS173 as a 10-bit comparator, 74LS460

DESCRIPTION

The PLS173 is a 24-pin PLA device which has 10 bidirectional outputs and 12 dedicated inputs. The output of the device is the sum of products of the inputs. The polarity of each output may be individually programmed as Active-High or Active-Low. A logic diagram of the device is shown in Appendix A. A 10-bit comparator similar to the 74LS460 compares two 10-bit data inputs to establish if EQUIVALENCE or NOT EQUIVALENCE exists. The output has True and Complement comparison status outputs. The logic diagram of the comparator is shown in Figure 1.

The truth table is as shown in Table 1 where vectors a and b are 10-bit inputs to A9 to A0 and B9 to B0. If the input to A9-A0 is bit-to-bit equivalent to the input to B9-B0, the two input vectors are considered EQUIVALENT, and output EQ goes High and NE goes Low. If the two input vectors are not bit-to-bit equivalent, then EQ goes Low and NE goes High. The circuit is implemented with AMAZE as shown in Figures 2a, 2b and 2c. The result of logic simulation of the circuit is shown in Figure 2d.

Notice that on the OR side of the program table in Figure 5, all the fuses in the OR-term are intact, which means that all the AND-terms are still connected to all the OR-terms. This feature provides for future modification. But if all the unused AND-terms are deleted, the device will run faster. There are also many unused AND-terms which will provide for future modification. But if they are deleted (both on the AND and OR side), it will amount to about 450μA per term power saving. Figure 3 is the program table with all unused terms deleted.

Table 1. Function Table

$A_9 - A_0$	$B_9 - B_0$	EQ	NE
a	a	H	L
b	b	H	L
a	b	L	H
b	a	L	H

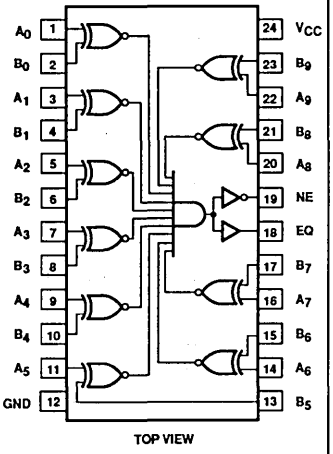


Figure 1. Logical Equivalent Circuit of 10-Bit Comparator

```
******************** AN24_173 ********************
#################### P I N  L I S T ####################

  LABEL    ** FNC **PIN --------- PIN** FNC **    LABEL
A0         ** I   **  1-|        |-24 ** +5V **VCC
B0         ** I   **  2-|        |-23 ** I   **B9
A1         ** I   **  3-|        |-22 ** I   **A9
B1         ** I   **  4-|   P    |-21 ** I   **B8
A2         ** I   **  5-|   L    |-20 ** I   **A8
B2         ** I   **  6-|   S    |-19 ** O   **NE
A3         ** I   **  7-|   1    |-18 ** /O  **EQ
B3         ** I   **  8-|   7    |-17 ** I   **B7
A4         ** I   **  9-|   3    |-16 ** I   **A7
B4         ** I   ** 10-|        |-15 ** I   **B6
A5         ** I   ** 11-|        |-14 ** I   **A6
GND        ** 0V  ** 12-|        |-13 ** I   **B5
                        ---------
```

a. Pin List of 10-Bit Comparator
Figure 2.

```
****************** AN24_173 ******************
@DEVICE TYPE
PLS173
@DRAWING
.................. 10-BIT COMPARATOR USING PLS173
@REVISION
.................. REV-0
@DATE
.................. OCT-14-85
@SYMBOL
.................. AN24_173
@COMPANY
.................. SIGNETICS
@NAME
.................. DAVID WONG
@DESCRIPTION
This circuit compares two 10-bit inputs. If they are bit-to-bit equivalent,
outputs EQ goes HIGH and NE goes LOW. If the inputs are not bit-to-bit equiv-
alent to each other, outputs EQ goes LOW and NE goes HIGH.
@COMMON PRODUCT TERM

T0  =  A0 * /B0 ;
T1  = /A0 *  B0 ;
T2  =  A1 * /B1 ;
T3  = /A1 *  B1 ;
T4  =  A2 * /B2 ;
T5  = /A2 *  B2 ;
T6  =  A3 * /B3 ;
T7  = /A3 *  B3 ;
T8  =  A4 * /B4 ;
T9  = /A4 *  B4 ;
T10 =  A5 * /B5 ;
T11 = /A5 *  B5 ;
T12 = /A6 *  B6 ;
T13 =  A6 * /B6 ;
T14 =  A7 * /B7 ;
T15 = /A7 *  B7 ;
T16 =  A8 * /B8 ;
T17 = /A8 *  B8 ;
T18 =  A9 * /B9 ;
T19 = /A9 *  B9 ;

@I/O DIRECTION
@OUTPUT POLARITY
@LOGIC EQUATION

EQ = /( T0 + T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 + T9 +
        T10 + T11 + T12 + T13 + T14 + T15 + T16 + T17 + T18 + T19 ) ;

NE =    T0 + T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 + T9 +
        T10 + T11 + T12 + T13 + T14 + T15 + T16 + T17 + T18 + T19 ;
```

b. Boolean Equations of 10-Bit Comparator
Figure 2 (Continued)

PLS173 as a 10-bit comparator, 74LS460 AN24

```
 ****************** AN24_173 ********************
Cust/Project - ................. DAVID WONG
Date         - ................. OCT-14-85
Rev/I. D.    - ................. REV-0

PLS173                                          !     POLARITY      !
---                                             --------------------
 T !                                            !H:H:H:H:H:L:H:H:H:H!
 E !-------------------------------------------------------------------
 R !              I       !       B(i)         !        B(o)        !
 M !1 1 ----------------------------------------------------------------
___!1_0_9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!
  0!- - - -,- - - -,- - L H!- -,- - - -,- - - -!A A,A A A A,A A A A!
  1!- - - -,- - - -,- - H L!- -,- - - -,- - - -!A A,A A A A,A A A A!
  2!- - - -,- - - -,L H - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  3!- - - -,- - - -,H L - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  4!- - - -,- - L H,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  5!- - - -,- - H L,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  6!- - - -,L H - -,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  7!- - - -,H L - -,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  8!- - L H,- - - -,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
  9!- - H L,- - - -,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
 10!L H - -,- - - -,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
 11!H L - -,- - - -,- - - -!- -,- - - -,- - - -!A A,A A A A,A A A A!
 12!- - - -,- - - -,- - - -!- -,- - - -,- - H L!A A,A A A A,A A A A!
 13!- - - -,- - - -,- - - -!- -,- - - -,- - L H!A A,A A A A,A A A A!
 14!- - - -,- - - -,- - - -!- -,- - - -,L H - -!A A,A A A A,A A A A!
 15!- - - -,- - - -,- - - -!- -,- - - -,H L - -!A A,A A A A,A A A A!
 16!- - - -,- - - -,- - - -!- -,L H - -,- - - -!A A,A A A A,A A A A!
 17!- - - -,- - - -,- - - -!- -,H L - -,- - - -!A A,A A A A,A A A A!
 18!- - - -,- - - -,- - - -!L H,- - - -,- - - -!A A,A A A A,A A A A!
 19!- - - -,- - - -,- - - -!H L,- - - -,- - - -!A A,A A A A,A A A A!
 20!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 21!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 22!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 23!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 24!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 25!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 26!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 27!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 28!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 29!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 30!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 31!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
 D9!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D8!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D7!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D6!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D5!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                    .
 D4!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                    .
 D3!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D2!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D1!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D0!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .

   B A B A B A B A B A B A B A N E B A B A B A B A N E B A B A
   5 5 4 4 3 3 2 2 1 1 0 0 9 9 8 8 E Q 7 7 6 6 9 9 8 8 E Q 7 7 6 6
```

c. Program Table of 10-Bit Comparator After Assembly of Boolean Equation File
Figure 2 (Continued)

```
PLS173    A:AN24_173.STD
" AMAZE FILE ID: AN24_173
" LOGIC SIMULATION OF 10-BIT COMPARATOR
"
" <==INPUTS==> <=B(I/O)=>    TRACE TERMS
" 11
" 109876543210 9876543210
"
  000000000000 0000LH0000 ;
  010000000000 0000HL0000 ;
  100000000000 0000HL0000 ;
  110000000000 0000LH0000 ;
  000100000000 0000HL0000 ;
  001000000000 0000HL0000 ;
  001100000000 0000LH0000 ;
  000001000000 0000HL0000 ;
  000010000000 0000HL0000 ;
  000011000000 0000LH0000 ;
  000000010000 0000HL0000 ;
  000000100000 0000HL0000 ;
  000000110000 0000LH0000 ;
  000000000100 0000HL0000 ;
  000000001000 0000HL0000 ;
  000000001100 0000LH0000 ;
  000000000001 0000HL0000 ;
  000000000010 0000HL0000 ;
  000000000011 0000LH0000 ;
  000000000000 0100HL0000 ;
  000000000000 1000HL0000 ;
  000000000000 1100LH0000 ;
  000000000000 0001HL0000 ;
  000000000000 0010HL0000 ;
  000000000000 0011LH0000 ;
  000000000000 0000HL0100 ;
  000000000000 0000HL1000 ;
  000000000000 0000LH1100 ;
  000000000000 0000HL0001 ;
  000000000000 0000HL0010 ;
  000000000000 0000LH0011 ;
"
" ------------ ----------    I/O CONTROL LINES
"              IIIIOOIIII    DESIGNATED I/O USAGE
"              IIIIOOIIII    ACTUAL I/O USAGE
"
" PINLIST...
" 13 11 10 09 08 07 06 05 04 03 02 01
" 23 22 21 20 19 18 17 16 15 14 ;
```

d. Test Vectors Generated by AMAZE After Logic Simulation
Figure 2 (Continued)

PLS173 as a 10-bit comparator, 74LS460 AN24

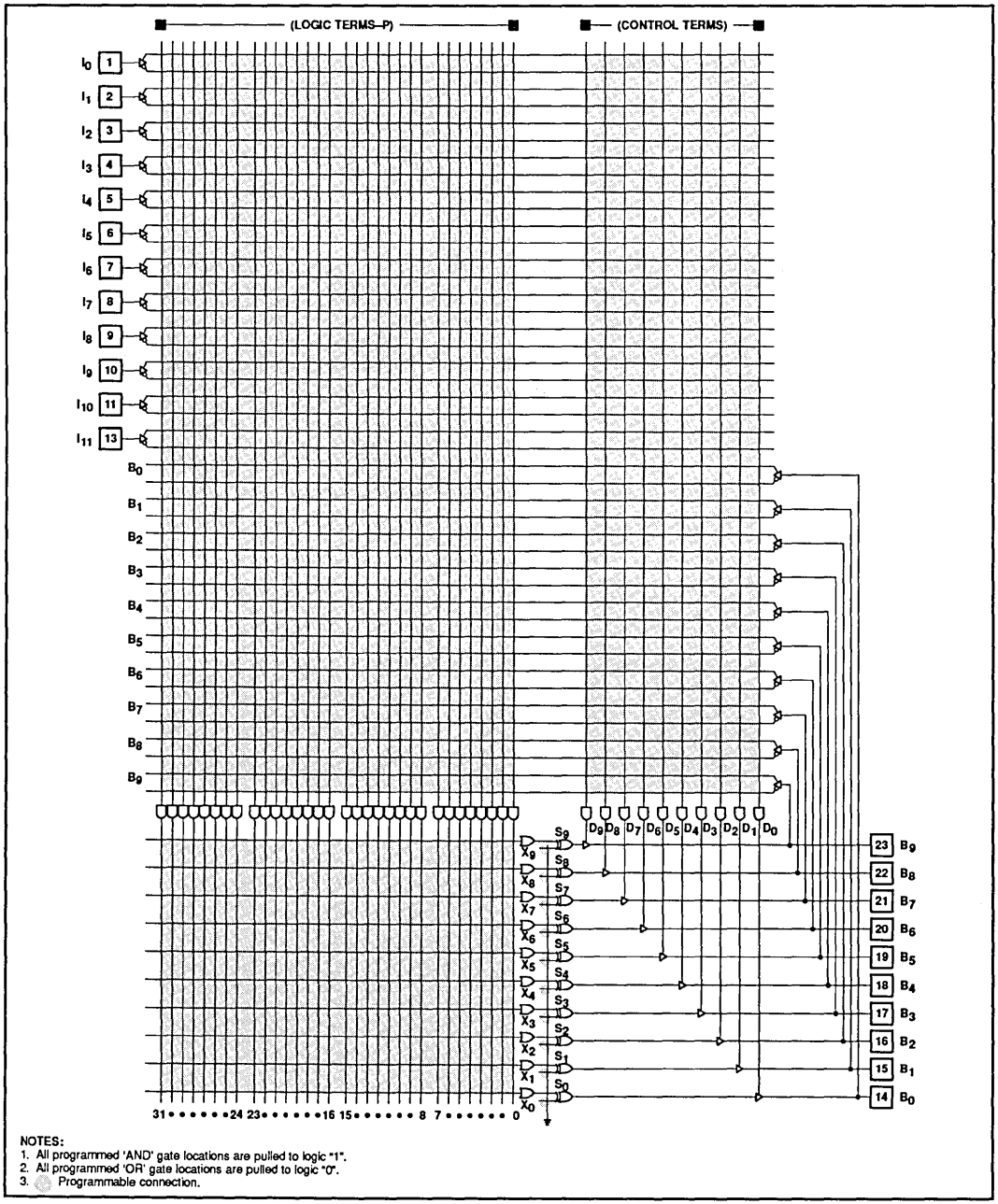
```
****************** AN24_173 ******************
Cust/Project - ................. DAVID WONG
Date         - ................. OCT-14-85
Rev/I. D.    - ................. REV-0

PLS173                                                    !     POLARITY      !
---                                                       --------------------
 T !                                                      !H:H:H:H:H:L:H:H:H:H!
 E !-------------------------------------------------------------------------
 R !                I       !        B(i)       !        B(o)        !
 M !1 1 ---------------------------------------------------------------------
   !1_0_9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!
  0!- - - -,- - - -,- - L H!- -,- - - -,- - - -!. .,. . A A,. . . .!
  1!- - - -,- - - -,- - H L!- -,- - - -,- - - -!. .,. . A A,. . . .!
  2!- - - -,- - - -,L H - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  3!- - - -,- - - -,H L - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  4!- - - -,- - L H,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  5!- - - -,- - H L,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  6!- - - -,L H - -,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  7!- - - -,H L - -,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  8!- - L H,- - - -,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
  9!- - H L,- - - -,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
 10!L H - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
 11!H L - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . A A,. . . .!
 12!- - - -,- - - -,- - - -!- -,- - - -,- - H L!. .,. . A A,. . . .!
 13!- - - -,- - - -,- - - -!- -,- - - -,- - L H!. .,. . A A,. . . .!
 14!- - - -,- - - -,- - - -!- -,- - - -,L H - -!. .,. . A A,. . . .!
 15!- - - -,- - - -,- - - -!- -,- - - -,H L - -!. .,. . A A,. . . .!
 16!- - - -,- - - -,- - - -!- -,L H - -,- - - -!. .,. . A A,. . . .!
 17!- - - -,- - - -,- - - -!- -,H L - -,- - - -!. .,. . A A,. . . .!
 18!- - - -,- - - -,- - - -!L H,- - - -,- - - -!. .,. . A A,. . . .!
 19!- - - -,- - - -,- - - -!H L,- - - -,- - - -!. .,. . A A,. . . .!
 20!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 21!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 22!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 23!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 24!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 25!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 26!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 27!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 28!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 29!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 30!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 31!- - - -,- - - -,- - - -!- -,- - - -,- - - -!. .,. . . .,. . . .!
 D9!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D8!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D7!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D6!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D5!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                    .
 D4!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                    .
 D3!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D2!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D1!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
 D0!0 0 0 0,0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .

   B A B A B A B A B A B A B A N E B A B A B A B A N E B A B A
   5 5 4 4 3 3 2 2 1 1 0 0 9 9 8 8 E Q 7 7 6 6 9 9 8 8 E Q 7 7 6 6
```

Figure 3. Program Table of 10-Bit Comparator with All Unused Terms Deleted

PLS173 as a 10-bit comparator, 74LS460 AN24

APPENDIX A

PLA LOGIC DIAGRAM FOR PLS173



NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. Programmable connection.

Philips Components-Signetics

Application Note	
Date of Issue	June 1988
Rev. Date	
Programmable Logic Devices	

AN28

High-speed 12-bit tracking A/D converter using PLS179

INTRODUCTION

The general technique underlying the operation of this A/D converter is illustrated by the functional block diagram in Figure 1. The system consists of a D/A converter, a comparator circuit, and digital logic circuitry. The digital logic circuitry outputs a digital value which is converted to analog by the D/A converter.

The comparator senses when the output is greater or less than the input and causes the digital circuit to decrement or increment its digital output respectively. The initial conversion is completed in 13 clock cycles. If tracking mode is used, the A/D converter then tracks the input voltage as it changes by incrementing or decrementing 1-LSB per clock. The tracking function makes it possible to make an A/D conversion in one clock cycle if the input changes less than the value of 1-LSB per clock period. The conversion may be halted and the digital output, as well as the converted analog output from DAC, will hold their output constant indefinitely. This feature works well as sample-and-hold since its output voltage will not decay over time whereas the output of an analog sample/hold will decay due to charge leakages.

In order to avoid the violation of setup time by the output of the comparator, its output is latched. There is a built-in 2-phase clock in U2 which may be used to drive the logic circuitry and the latch of the comparator (see Signetics NE5105 data sheet for details on output latches of voltage comparators).

The analog input voltage may be sampled and held by an analog sample/hold circuit to keep the input to the ADC from changing. The DONE output may be used to control the sample-and-hold if needed.

This paper discusses only the digital circuit which contains the SAR and the Up/Down Counter. The analog circuits are not within the scope of this paper.

SAR

Two PLS179s are connected together to form a 12-bit shift register and up/down counter. The schematic diagram of the A/D converter is shown in Figure 2. U2 contains bits 0 to 4 and U1 contains Bits 5 to 11. Interconnects are made as shown in the diagram. The digital output to the DAC is in natural binary format (e.g. 0000 0000 0000 equal zero, and 1111 1111 1111 is full scale or 4095). After the/ST input becomes 0, at the rising edge of the next clock, the SAR is initialized to half-scale (1000 0000 0000) and the DONE flip-flop is reset to output 0 which causes the open-collector output /DONE_OC to become high impedance. The digital output is converted by the DAC and is compared to the analog input voltage by the comparator. If the digital output is greater than the analog input, the SAR shifts the 1 to next MSB on the right. The content of the SAR becomes (0100 0000 0000). If the digital output is still greater than the input, the SAR shifts right one bit again. The content of the SAR then becomes (0010 0000 0000). The shifting of 1 to the next MSB in equivalent to reducing by half the value of the bit under consideration. If the output is still too large, the SAR reduces it by half again by shifting to the right one more time. The SAR keeps shifting to the right until the digital output is less than the input. When the output is less than the input, the SAR adds one bit to the next MSB while keeping all the higher order bits unchanged. For example, if the current output is 0001 0000 0000 and the output is less than the input, the SAR adds one bit to the right at the next clock. The output becomes 0001 1000 0000. The output is again compared to the input. If the addition of that one bit is too much, it will be shifted to the right until the output becomes less than the input. When that happens, that SAR will again add one bit to the right. The algorithm of the SAR may be summarized as the following: If the output is greater than the input, shift to the right; otherwise add one bit to the right. This process continues until all 12 bits have been operated on. The last bit (Bit 0) is always changed from 0 to 1, which is used as the condition to set DONE to 1 which, in turn, sets open-collector output, /DONE_OC, to 0.

UP/DOWN COUNTER

After DONE becomes 1, if /ST and /HOLD are 1 and /TRACK is 0, the SAR turns into a 12-bit up/down counter. If the analog input voltage increases, the counter will increment by 1 at every clock until it matches the input. If the input decreases, the counter will decrement by 1. When /HOLD becomes 0, the counter is inhibited and the output is held indefinitely. The counters consist of 12 toggle flip-flops and 2 p-terms per flip-flop for directional control. The counter will operate only after the approximation cycle is completed and DONE is 1.

Since the /ST and /HOLD inputs may be asynchronous with the clock, in order to minimize the possibilities of having a metastable condition from happening, these inputs close-up are latched by flip-flops /START of U1 and /HLD of U2 respectively. Once they are latched, subsequent operation begins at the rising-edge of the next clock. The output of the comparator may be latched to prevent setup time violation. (Signetics NE5105 is a high-speed comparator with an output latch. External latch may be used with other comparators.)

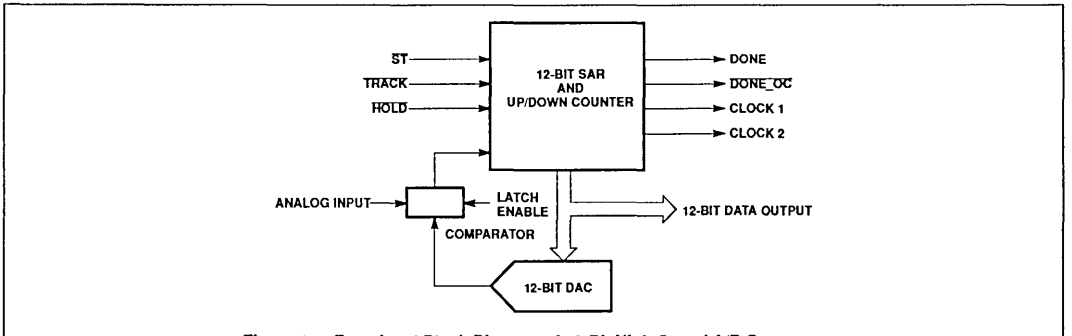


Figure 1. Functional Block Diagram of 12-Bit High-Speed A/D Converter

High-speed 12-bit tracking A/D converter using PLS179 AN28

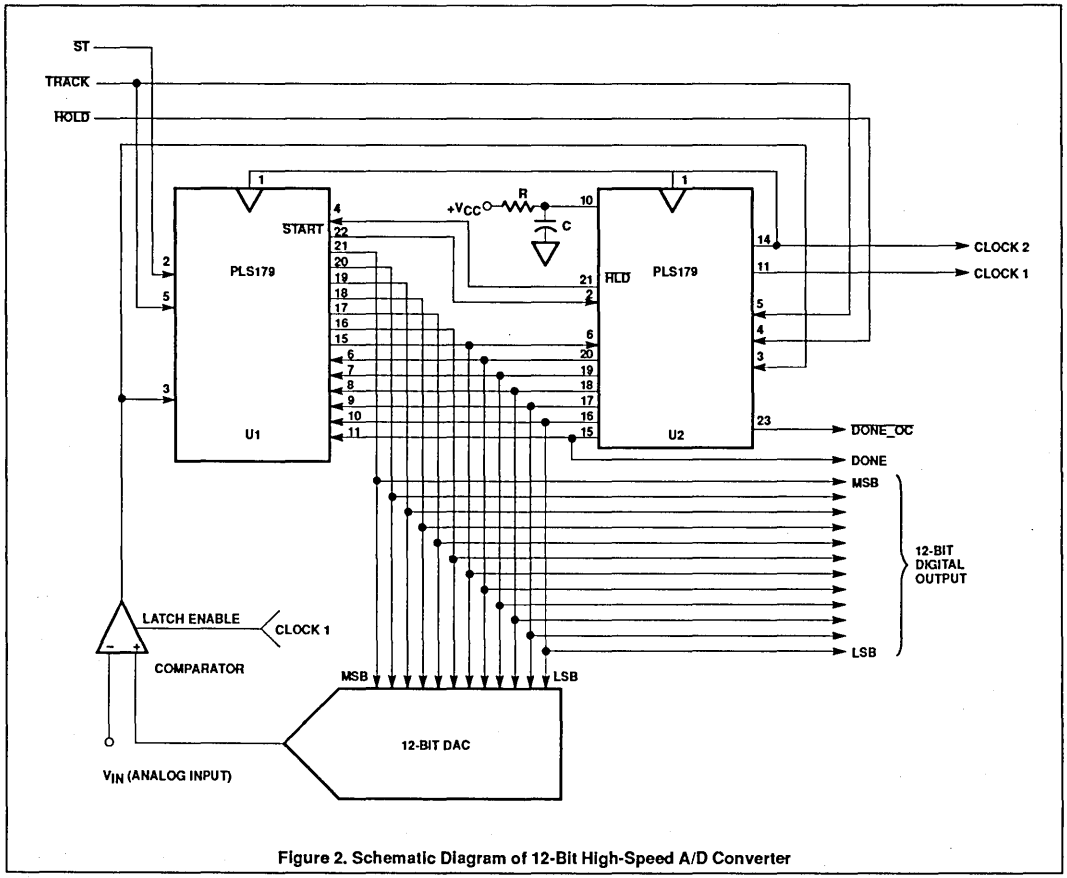


Figure 2. Schematic Diagram of 12-Bit High-Speed A/D Converter

High-speed 12-bit tracking A/D converter using PLS179 AN28

CLOCKS

U2 generates an optional 2-phase clock which may be used to control the latch of the comparator. The two clocks are basically 180° out of phase and CLOCK2 has an additional 25ns propagation delay behind CLOCK1. CLOCK2 is used to drive the clock-inputs of the PLS179 devices.

The clock frequency is controlled by R and C. Those who want to use the built-in clock should experiment with RC time constants for the best value. It is recommended that the capacitance should be less than 1000pF for best results.

DONE AND /DONE_OC

The output DONE is reset to 0 when /ST is 0. It remains 0 until the approximation cycle is completed. After the least significant bit becomes 1, the DONE bit becomes 1 at the next clock. It remains 1 until it is reset again by input /ST.

The /DONE_OC output is configured to emulate an open-collector output. The output is programmed to have a logic 0. When DONE is 0, the 3-State output buffer is set to Hi-Z condition. As soon as DONE equals 1, the 3-State buffer is enabled and /DONE_OC becomes 0.

In the initial phase of A/D conversion, 13 clock cycles are required. It is essential that the input voltage to the comparator remains unchanged while the SAR is converting. It may be necessary to have a sample/hold at the front end. The DONE output may be used to control the analog sample/hold circuit.

INPUT LATCHES

Flip-flop /START and 2 p-terms in U1 are configured as a non-inverting D flip-flop. The input, /ST, and the output /START have the same polarities. Flip-flop /HLD and 2 p-terms in U2 also form a non-inverting D flip-flop. The output /HLD and the input /HOLD have the same polarities.

AMAZE INPLEMENTATION

The implementation of the logic circuit using AMAZE is as shown in the appendices. The SAR circuit is first designed as a state machine (file name: ADCS.SEE). It is then partitioned into two PLS179s after proper pin assignments are made. Then the up/down counter, input latches, 2-phase clocks and the open-collector output, are implemented by using Boolean equations in their respective .BEE files (file names: ADCB1.BEE and ADCB2.BEE) in AMAZE. The files are then assembled to produce the fuse-maps of PLS179 (ADCB1.STD and ADCB2.STD).

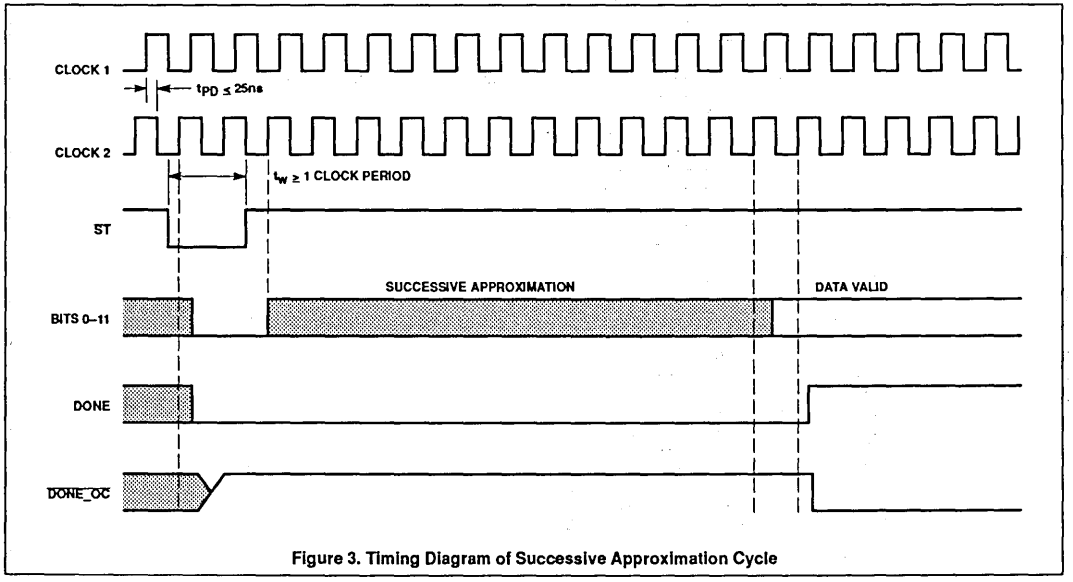


Figure 3. Timing Diagram of Successive Approximation Cycle

High-speed 12-bit tracking A/D converter using PLS179 AN28

APPENDIX A: STATE EQUATIONS OF SAR

```
File Name : ADCS
Date : 10/21/1986
Time : 11:2:14

@DEVICE SELECTION
ADCB1/PLS179
ADCB2/PLS179

@STATE VECTORS
[ /START, BIT11, BIT10, BIT9, BIT8, BIT7, BIT6, BIT5, BIT4, BIT3, BIT2,
BIT1, BIT0, DONE ]

INIT       = 0 ---- ---- ---- - b ;   "START CONVERSION PROCESS"
HALFSCALE, = - 1000 0000 0000 0 b ;   "SET SAR TO HALF SCALE"
ST2048     = 1 1000 0000 0000 0 b ;   "PRESENT STATE = 2048 (HALF SCALE)"
ST1024     = 1 -100 0000 0000 0 b ;
ST512      = 1 --10 0000 0000 0 b ;
ST256      = 1 ---1 0000 0000 0 b ;
ST128      = 1 ---- 1000 0000 0 b ;
ST64       = 1 ---- -100 0000 0 b ;
ST32       = 1 ---- --10 0000 0 b ;
ST16       = 1 ---- ---1 0000 0 b ;
ST8        = 1 ---- ---- 1000 0 b ;
ST4        = 1 ---- ---- -100 0 b ;
ST2        = 1 ---- ---- --10 0 b ;
ST1        = 1 ---- ---- ---1 0 b ;

AD1024     = - -1-- ---- ---- - b ;   "ADD 1 BIT TO THE RIGHT"
AD512      = - --1- ---- ---- - b ;
AD256      = - ---1 ---- ---- - b ;
AD128      = - ---- 1--- ---- - b ;
AD64       = - ---- -1-- ---- - b ;
AD32       = - ---- --1- ---- - b ;
AD16       = - ---- ---1 ---- - b ;
AD8        = - ---- ---- 1--- - b ;
AD4        = - ---- ---- -1-- - b ;
AD2        = - ---- ---- --1- - b ;
AD1        = - ---- ---- ---1 - b ;
END        = - ---- ---- ---- 1 b ;

SH1024     = - 01-- ---- ---- - b ;   "SHIFT ONE BIT TO THE RIGHT"
SH512      = - -01- ---- ---- - b ;
SH256      = - --01 ---- ---- - b ;
SH128      = - ---0 1--- ---- - b ;
SH64       = - ---- 01-- ---- - b ;
SH32       = - ---- -01- ---- - b ;
SH16       = - ---- --01 ---- - b ;
SH8        = - ---- ---0 1--- - b ;
SH4        = - ---- ---- 01-- - b ;
SH2        = - ---- ---- -01- - b ;
SH1        = - ---- ---- --01 - b ;
SH0        = - ---- ---- ---0 1 b ;

@INPUT VECTORS
[ COMPARE ]
GREATER = 1 b ;  "IF DIGITAL OUTPUT IS GREATER THAN ANALOG INPUT, ...."
LESS    = 0 b ;  "IF DIGITAL OUTPUT IS LESS THAN ANALOG INPUT, ...."

@OUTPUT VECTORS
```

APPENDIX A: STATE EQUATIONS OF SAR (Continued)

```
@TRANSITIONS
WHILE [ INIT ]
        IF [] THEN [ HALFSCALE ]  "INITIALIZE REGISTER TO HALF SCALE"
WHILE [ ST2048 ]
        IF [ GREATER ] THEN [ SH1024 ] "IF GREATER THAN, SHIFT 1 BIT"
        IF [ LESS ] THEN [ AD1024 ]    "IF LESS THAN, ADD 1 BIT"
WHILE [ ST1024 ]
        IF [ GREATER ] THEN [ SH512 ]
        IF [ LESS ] THEN [ AD512 ]
WHILE [ ST512 ]
        IF [ GREATER ] THEN [ SH256 ]
        IF [ LESS ] THEN [ AD256 ]
WHILE [ ST256 ]
        IF [ GREATER ] THEN [ SH128 ]
        IF [ LESS ] THEN [ AD128 ]
WHILE [ ST128 ]
        IF [ GREATER ] THEN [ SH64 ]
        IF [ LESS ] THEN [ AD64 ]
WHILE [ ST64 ]
        IF [ GREATER ] THEN [ SH32 ]
        IF [ LESS ] THEN [ AD32 ]
WHILE [ ST32 ]
        IF [ GREATER] THEN [ SH16 ]
        IF [ LESS ] THEN [ AD16 ]
WHILE [ ST16 ]
        IF [ GREATER ] THEN [ SH8 ]
        IF [ LESS ] THEN [ AD8 ]
WHILE [ ST8 ]
        IF [ GREATER ] THEN [ SH4 ]
        IF [ LESS ] THEN [ AD4 ]
WHILE [ ST4 ]
        IF [ GREATER ] THEN [ SH2 ]
        IF [ LESS ] THEN [ AD2 ]
WHILE [ ST2 ]
        IF [ GREATER ] THEN [ SH1 ]
        IF [ LESS ] THEN [ AD1 ]
WHILE [ ST1 ]
        IF [ GREATER ] THEN [ SH0 ]
        IF [] THEN [END]
```

High-speed 12-bit tracking A/D converter using PLS179 AN28

APPENDIX B: PIN LISTS

```
File Name : ADCB2
Date : 10/21/1986
Time : 10:58:26

#################### P I N  L I S T ####################

   LABEL  ** FNC **PIN --------- PIN** FNC **    LABEL
CLOCK     ** CK  **  1-:        :-24 ** +5V **VCC
/START    ** I   **  2-:        :-23 ** /B  **/DONE_OC
COMPARE   ** I   **  3-:        :-22 ** B   **N/C
/HOLD     ** I   **  4-:   P    :-21 ** O   **/HLD
/TRACK    ** I   **  5-:   L    :-20 ** O   **BIT4
BIT5      ** I   **  6-:   S    :-19 ** O   **BIT3
N/C       ** I   **  7-:   1    :-18 ** O   **BIT2
N/C       ** I   **  8-:   7    :-17 ** O   **BIT1
N/C       ** I   **  9-:   9    :-16 ** O   **BIT0
RC        ** /B  ** 10-:        :-15 ** O   **DONE
CLOCK1    ** O   ** 11-:        :-14 ** /O  **CLOCK2
GND       ** 0V  ** 12-:        :-13 ** /OE **N/C
                     ---------
```

```
File Name : ADCB1
Date : 10/21/1986
Time : 10:53:7

#################### P I N  L I S T ####################

   LABEL  ** FNC **PIN --------- PIN** FNC **    LABEL
CLOCK     ** CK  **  1-:        :-24 ** +5V **VCC
/ST       ** I   **  2-:        :-23 ** /B  **N/C
COMPARE   ** I   **  3-:        :-22 ** O   **/START
/HLD      ** I   **  4-:   P    :-21 ** O   **BIT11
/TRACK    ** I   **  5-:   L    :-20 ** O   **BIT10
BIT4      ** I   **  6-:   S    :-19 ** O   **BIT9
BIT3      ** I   **  7-:   1    :-18 ** O   **BIT8
BIT2      ** I   **  8-:   7    :-17 ** O   **BIT7
BIT1      ** I   **  9-:   9    :-16 ** O   **BIT6
BIT0      ** I   ** 10-:        :-15 ** O   **BIT5
DONE      ** I   ** 11-:        :-14 ** /B  **N/C
GND       ** 0V  ** 12-:        :-13 ** /OE **N/C
                     ---------
```

APPENDIX C: BOOLEAN EQUATIONS OF UP/DOWN COUNTER AND INPUT LATCH

```
File Name : ADCB1
Date : 10/21/1986
Time : 10:54:48

@DEVICE TYPE
PLS179
@DRAWING
@REVISION
@DATE
@SYMBOL
    FILE NAME : ADCB1

@COMPANY
@NAME
@DESCRIPTION
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@I/O DIRECTION
@OUTPUT POLARITY
@FLIP FLOP CONTROL
     FC = 1 ;           "SET ALL FLIP FLOP TO BE J/K"

@OUTPUT ENABLE
@REGISTER LOAD
@ASYNCHRONOUS PRESET/RESET
@FLIP FLOP MODE
@LOGIC EQUATION
 "NON-INVERTING INPUT LATCH: /START = /ST "
  START  : J =  ST ;
           K = /ST ;
 "UP/DOWN COUNTER ROUTINE"
  /BIT5  : T = /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 *
               /BIT2 * /BIT3 * /BIT4 +
               /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *
                BIT2 *  BIT3 * BIT4 ;
  /BIT6  : T = /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 *
               /BIT2 * /BIT3 * /BIT4 * /BIT5 +
               /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *
                BIT2 *  BIT3 * BIT4 *  BIT5 ;
  /BIT7  : T = /START * TRACK * DONE * /HLD *  COMPARE *
               /BIT0 * /BIT1 * /BIT2 * /BIT3 * /BIT4 *  /BIT5 * /BIT6 +
               /START * TRACK * DONE * /HLD * /COMPARE  *
                BIT0 *  BIT1 *  BIT2 *  BIT3 *  BIT4 *   BIT5 *  BIT6 ;
  /BIT8  : T = /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 *
               /BIT2 * /BIT3 * /BIT4 * /BIT5 * /BIT6 * /BIT7 +
               /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *

                BIT2 *  BIT3 *  BIT4 *  BIT5 *  BIT6 *  BIT7 ;
  /BIT9  : T = /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 *
               /BIT2 * /BIT3 * /BIT4 * /BIT5 * /BIT6 * /BIT7 * /BIT8 +
               /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *

                BIT2 *  BIT3 *  BIT4 *  BIT5 *  BIT6 *  BIT7 *  BIT8 ;
  /BIT10 : T = /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 *
               /BIT2 * /BIT3 * /BIT4 * /BIT5 * /BIT6 * /BIT7 * /BIT8 *
               /BIT9 +
               /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *

                BIT2 *  BIT3 *  BIT4 *  BIT5 *  BIT6 *  BIT7 *  BIT8 *
                BIT9 ;
  /BIT11 : T = /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 *
               /BIT2 * /BIT3 * /BIT4 * /BIT5 * /BIT6 * /BIT7 * /BIT8 *
               /BIT9 *
               /BIT10 +
               /START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
                BIT2 *  BIT3 *  BIT4 *  BIT5 *  BIT6 *  BIT7 *  BIT8 *
                BIT9 *  BIT10 ;
```

APPENDIX C: BOOLEAN EQUATIONS OF UP/DOWN COUNTER AND INPUT LATCH (Continued)

```
              File Name : ADCB2
              Date : 10/21/1986
              Time : 10:58:56

              @DEVICE TYPE
              PLS179
              @DRAWING
              @REVISION
              @DATE
              @SYMBOL
                FILE NAME : ADCB2

              @COMPANY
              @NAME
              @DESCRIPTION
              @COMMON PRODUCT TERM
              @COMPLEMENT ARRAY
              @I/O DIRECTION
                   D0 = RC ;          "RC OSCILLATOR"
                   D3 = DONE ;        "ENABLE /DONE_OC TO OUTPUT A LOGIC LOW."
              @OUTPUT POLARITY
              @FLIP FLOP CONTROL
                   FC = 1 ;
              @OUTPUT ENABLE
              @REGISTER LOAD
              @ASYNCHRONOUS PRESET/RESET
              @FLIP FLOP MODE
                   "M0, M1, M2, M3, M4, M5 = 1 ; SET F0 - F5 TO J/K FLIP FLOPS."

              @LOGIC EQUATION
               "NON-INVERTING INPUT LATCH :  /HLD = /HOLD"
                HLD    : J =  HOLD ;
                         K = /HOLD ;

               "UP/DOWN COUNTER ROUTINE"
               /BIT0   : T = /START * TRACK * DONE * /HLD ;
               /BIT1   : T = /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 +
                             /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 ;
               /BIT2   : T = /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 +
                             /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 ;
               /BIT3   : T = /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *
BIT2 +
                             /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 * /
BIT2 ;
               /BIT4   : T = /START * TRACK * DONE * /HLD * /COMPARE *  BIT0 *  BIT1 *
BIT2 *
                              BIT3 +
                             /START * TRACK * DONE * /HLD *  COMPARE * /BIT0 * /BIT1 * /
BIT2 *
                             /BIT3 ;
               /DONE_OC    = /( 1 ) ;

               "RC OSCILLATOR"
                RC         = /(1) ;
                CLOCK1     = RC ;
                CLOCK2     = /( CLOCK1 ) ;       "BUILT-IN DELAY OF 1 tPD"
```

High-speed 12-bit tracking A/D converter using PLS179 AN28

APPENDIX D: U1 ADCB1 FUSE MAP

```
File Name : ADCB1
Date : 10/21/1986
Time : 10:56:5

Cust/Project -
Date        -
Rev/I. D.   -

PLS179                          ! F/F TYPE      ! E(b)= !E(a) = !POLARTY!
                                !               !       !       !        !
 T !                            !A:A:A:A:A:A:A:A! O     !  O    !L:L:L:L!
 E !-------------------------------------------------------------------------
 R ! !       I       ! B(i)  !     Q(p)      !       Q(n)     ! B(o)  !
 M !C!-------------------------------------------------------------------------
   !  !7 6 5 4 3 2 1 0!3 2 1 0!7 6 5 4 3 2 1 0!7 6 5 4 3 2 1 0!3 2 1 0!
  0!A!- - - -,- - - L!- - - -!- - - -,- - - -!H - - -,- - - -!A A A A!
  1!A!- - - -,- - - H!- - - -!- - - -,- - - -!L - - -,- - - -!A A A A!
  2!A!L L L L,L H H -!- - H L!L - - -,- - - -!- - - -,- - - -!A A A A!
  3!A!H H H H,L H L -!- - H H!L - - -,- - - -!- - - -,- - - -!A A A A!
  4!A!L L L L,L H H -!- - H L!L - - -,- - - H!- - - -,- - O -!A A A A!
  5!A!H H H H,L H L -!- - H H!L - - -,- - - L!- - - -,- - O -!A A A A!
  6!A!L L L L,L H H -!- - H L!L - - -,- - H H!- - - -,- O - -!A A A A!
  7!A!H H H H,L H L -!- - H H!L - - -,- - L L!- - - -,- O - -!A A A A!
  8!A!L L L L,L H H -!- - H L!L - - -,- H H H!- - - -,O - - -!A A A A!
  9!A!H H H H,L H L -!- - H H!L - - -,- L L L!- - - -,O - - -!A A A A!
 10!A!L L L L,L H H -!- - H L!L - - -,H H H H!- - - O,- - - -!A A A A!
 11!A!H H H H,L H L -!- - H H!L - - -,L L L L!- - - O,- - - -!A A A A!
 12!A!L L L L,L H H -!- - H L!L - - H,H H H H!- - O -,- - - -!A A A A!
 13!A!H H H H,L H L -!- - H H!L - - L,L L L L!- - O -,- - - -!A A A A!
 14!A!L L L L,L H H -!- - H L!L - H H,H H H H!- O - -,- - - -!A A A A!
 15!A!H H H H,L H L -!- - H H!L - L L,L L L L!- O - -,- - - -!A A A A!
 16!A!- - - -,- - - -!- - - -!H - - -,- - - -!- L H H,H H H H!A A A A!
 17!A!L L L L,- - H -!- - L L!L L H H,H H H H!- H L -,- - - -!A A A A!
 18!A!L L L L,- - L -!- - L L!L L H H,H H H H!- - L -,- - - -!A A A A!
 19!A!L L L L,- - H -!- - L L!L - L H,H H H H!- - H L,- - - -!A A A A!
 20!A!L L L L,- - L -!- - L L!L - L H,H H H H!- - - L,- - - -!A A A A!
 21!A!L L L L,- - H -!- - L L!L - - L,H H H H!- - - H,L - - -!A A A A!
 22!A!L L L L,- - L -!- - L L!L - - L,H H H H!- - - -,L - - -!A A A A!
 23!A!L L L L,- - H -!- - L L!L - - -,L H H H!- - - -,H L - -!A A A A!
 24!A!L L L L,- - L -!- - L L!L - - -,L H H H!- - - -,- L - -!A A A A!
 25!A!L L L L,- - H -!- - L L!L - - -,- L H H!- - - -,- H L -!A A A A!
 26!A!L L L L,- - L -!- - L L!L - - -,- L H H!- - - -,- - L -!A A A A!
 27!A!L L L L,- - H -!- - L L!L - - -,- - L H!- - - -,- - H L!A A A A!
 28!A!L L L L,- - L -!- - L L!L - - -,- - L H!- - - -,- - - L!A A A A!
 29!A!L L L L,- - H -!- - L L!L - - -,- - - L!- - - -,- - - H!A A A A!
 30!A!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 31!A!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 Fc!A!- - - -,- - - -!- - - -!- - - -,- - - -!                  .
 Pb!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 Rb!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 Lb!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 Pa!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 Ra!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 La!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 D3!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 D2!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 D1!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .
 D0!.!O O O O,O O O O!O O O O!O O O O,O O O O!                  .

     B B B B / / C / N N D B / B B B B B B B / B B B B B B B B N N D B
     I I I I I T H O S / / O I S I I I I I I I S I I I I I I I I / / O I
     T T T T T R L M T C C N T T T T T T T T T T T T T T T T T C C N T
     1 2 3 4 A D P       E O A 1 1 9 8 7 6 5 A 1 1 9 8 7 6 5     E O
               C   A           R 1 0           R 1 0
               K   R           T               T
                   E
```

High-speed 12-bit tracking A/D converter using PLS179 AN28

APPENDIX D: U2 ADCB2 FUSE MAP (Continued)

```
File Name : ADCB2
Date : 10/21/1986
Time : 11:0:7

Cust/Project -
Date         -
Rev/I. D.    -

PLS179                              ! F/F TYPE      ! E(b)= !E(a) = !POLARTY!
                                    !_______________!_______!_______!_______!
 T !                                !A:A:A:A:A:A:A:A!   O   !   O   !L:L:H:L!
 E !-------------------------------------------------------------------------
 R ! !        I          ! B(i)  !      Q(p)       !       Q(n)      ! B(o)  !
 M !C!-------------------------------------------------------------------------
   !_!7_6_5_4_3_2_1_0!3_2_1_0!7_6_5_4_3_2_1_0!7_6_5_4_3_2_1_0!3_2_1_0!
  0!A!- - - -,- L - -!- - - -!- - - -,- - - -!O H - -,- - - -!. . . .!
  1!A!- - - -,- H - -!- - - -!- - - -,- - - -!O L - -,- - - -!. . . .!
  2!A!- - - -,L - - H!- - - -!- L - -,- - - -!O - - -,- - O -!. . . .!
  3!A!- - - -,L - L H!- - - -!- L - -,- - L L!O - - -,- O - -!. . . .!
  4!A!- - - -,L - H H!- - - -!- L - -,- - H L!O - - -,- O - -!. . . .!
  5!A!- - - -,L - L H!- - - -!- L - -,- L L L!O - - -,O - - -!. . . .!
  6!A!- - - -,L - H H!- - - -!- L - -,- H H L!O - - -,O - - -!. . . .!
  7!A!- - - -,L - L H!- - - -!- L - -,L L L L!O - - O,- - - -!. . . .!
  8!A!- - - -,L - H H!- - - -!- L - -,H H H L!O - - O,- - - -!. . . .!
  9!A!- - - -,L - L H!- - - -!- L - L,L L L L!O - O -,- - - -!. . . .!
 10!A!- - - -,L - H H!- - - -!- L - H,H H H L!O - O -,- - - -!. . . .!
 11!A!- - - -,- - - -!- - - -!- - - -,- - - -!O - - -,- - - -!A . . .!
 12!A!- - - -,- - - -!- - - H!- - - -,- - - -!O - - -,- - - -!. . A .!
 13!A!- - - -,- - - -!- - H -!- - - -,- - - -!O - - -,- - - -!. A . .!
 14!A!- - - -,- - - L!- - - -!- - - -,- - - -!O - H H,H H H H!. . . .!
 15!A!- - - H,- - H H!- - - -!- - H H,H H H H!O - L -,- - - -!. . . .!
 16!A!- - - H,- - L H!- - - -!- - H H,H H H H!O - L -,- - - -!. . . .!
 17!A!- - - -,- - H H!- - - -!- - L H,H H H H!O - H L,- - - -!. . . .!
 18!A!- - - -,- - L H!- - - -!- - L H,H H H H!O - - L,- - - -!. . . .!
 19!A!- - - -,- - H H!- - - -!- - - L,H H H H!O - - H,L - - -!. . . .!
 20!A!- - - -,- - L H!- - - -!- - - L,H H H H!O - - -,L - - -!. . . .!
 21!A!- - - -,- - H H!- - - -!- - - -,L H H H!O - - -,H L - -!. . . .!
 22!A!- - - -,- - L H!- - - -!- - - -,L H H H!O - - -,- L - -!. . . .!
 23!A!- - - -,- - H H!- - - -!- - - -,- L H H!O - - -,- H L -!. . . .!
 24!A!- - - -,- - L H!- - - -!- - - -,- L H H!O - - -,- - L -!. . . .!
 25!A!- - - -,- - H H!- - - -!- - - -,- - L H!O - - -,- - H L!. . . .!
 26!A!- - - -,- - - H!- - - -!- - - -,- - L H!O - - -,- - - L!. . . .!
 27!O!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 28!O!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 29!O!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 30!O!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 31!O!O O O O,O O O O!O O O O!O O O O,O O O O!O O O O,O O O O!A A A A!
 Fc!A!- - - -,- - - -!- - - -!- - - -,- - - -!
 Pb!.!O O O O,O O O O!O O O O!O O O O,O O O O!
 Rb!.!O O O O,O O O O!O O O O!O O O O,O O O O!
 Lb!.!O O O O,O O O O!O O O O!O O O O,O O O O!
 Pa!.!O O O O,O O O O!O O O O!O O O O,O O O O!
 Ra!.!O O O O,O O O O!O O O O!O O O O,O O O O!
 La!.!O O O O,O O O O!O O O O!O O O O,O O O O!
 D3!-!- - - -,- - - -!- - - -!- - - -,- - - -!
 D2!-!- - - -,- - - -!- - - -!- - - -,- - - -!
 D1!-!- - - -,- - - -!- - - -!- - - -,- - - -!
 D0!-!- - - -,- - - -!- - - -!- - - -,- - - -!

    N N N B / / C / / C C R N / B B B B B D N / B B B B B D / C C R
    / / / I T H O S D L L C / H I I I I I O / H I I I I I O D L L C
    C C C T R O M T O O O   C L T T T T T N C L T T T T T N O O O
          5 A L P A N C C       D 4 3 2 1 0 E   D 4 3 2 1 0 E N C C
            C D A R E K K                                   E K K
            K   R T _ 2 1                                   _ 2 1
                E   O                                       O
                    C                                       C
```

Philips Components–Signetics

Application Note	
Date of Issue	June 1988
Rev. Date	
Programmable Logic Devices	

AN29

PLHS501 Programmable Macro Logic primer

SUMMARY

The evolution of Programmable Logic Devices (PLD's) has led to the birth of a new generation of programmable devices designated as PML (Programmable Macro Logic). The immense versatility of these devices brings them closer as plausible alternatives to semicustom design approaches in low-to-medium ranges of applications. The following paper begins with a background on PML and a brief description of the PML basic architecture (See Reference 1). Next, the first PML devices are introduced with a detailed discussion of the PLHS501. The implementation of PML in the SNAP software package is presented. A system level example intends to demonstrate the capabilities of PML as an eloquent and efficient design alternative.

THE EMERGENCE OF THE THIRD GENERATION PLD ARCHITECTURE

PML was introduced at WESCON '85 by Signetics Corporation. The unique architecture of PML breaks away into a new era of programmable logic devices. The purpose of the PML architecture is to overcome the two level AND-OR bottleneck and provide the user with a higher level of logic integration. Current PLD's rely on two levels of logic transformation to implement combinational logic in Sum-Of-Products (SOP) form. In addition, various PLD's make use of higher level macros such as flip-flops to form sequential logic functions. These macros connect the AND-OR chain to dedicated I/O pins.

Figure 1 show the basic architecture of one of the most recent PAL® devices. It is clear that this architecture is inefficient in making full use of the available on-chip resources. This is due to the fact that an unused I/O macro will be wasted and remains futile.

For example, if an I/O pin is used as an input, the output macros are all wasted. Obviously, such an architecture cannot provide the user with an increase in the levels of logic integration. The PML device takes advantage of the fundamental architecture shown in Figure 2 to overcome these deficiencies and waste of on-chip resources. As shown in Figure 2, PML incorporates the NAND-NAND gate equivalence to break the AND-OR bottleneck.

The core of the PML is the programmable NAND-NAND network which connects the input and output macros to each other. Thus the inputs, outputs, and function macros are all connected by a single array.

The first device is the PLHS501. The seemingly simple structure of this device can implement every logic function furnished by the current PAL/PLA devices. Although the PLHS501 is principally a combinational logic device, its unique architecture makes it an ideal tool for applications involving asynchronous state machines (See Reference 2).

The PLHS502 is a sequential logic device. It supplements the PLHS501 features with sixteen edge-triggered flip-flops. The device can provide diverse applications encompassing synchronous and asynchronous state machine designs.

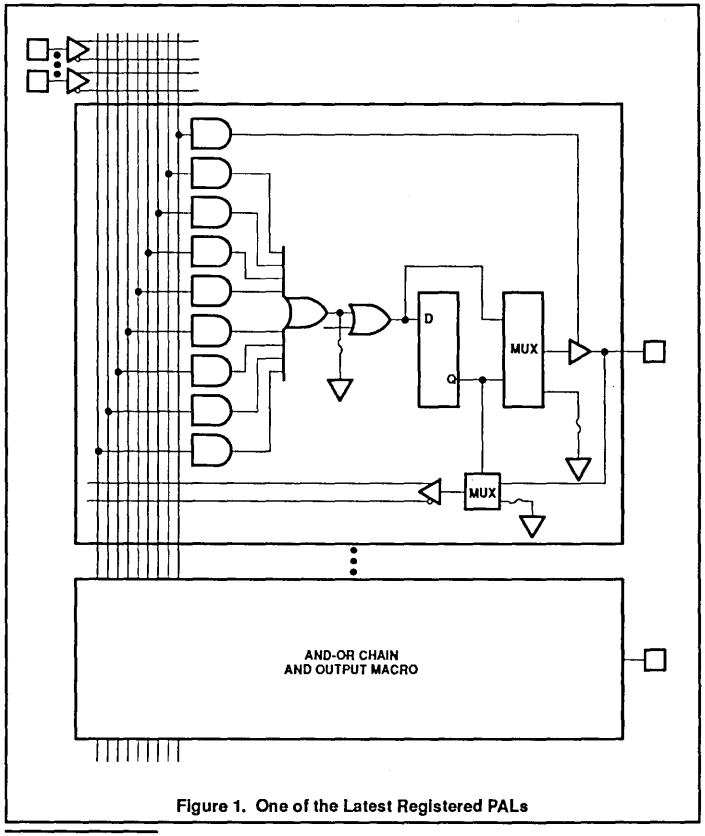


Figure 1. One of the Latest Registered PALs

PAL is a registered trademark of AMD/MMI, Inc.

PLHS501 Programmable Macro Logic primer AN29

Table 1. Functional Description of PLHS501 I/O Pins

PIN NO.	IDENTIFIER	FUNCTION
41–45, 47–52, 1–7, 9–14	I0–I23	Dedicated inputs
37–40	/B0–/B3	Fuse-programmable bidirectional I/Os with Active-Low outputs. Can be configured as open-collector outputs.
15–18	B4–B7	Logic controlled bidirectional I/Os with Active-High 3-State outputs.
28–29 30–31 32–33 35–36	X0–X1 X2–X3 X4–X5 X6–X7	Pairs of 3-State Exclusive-OR outputs that have common Output Enable.
19, 21 22–23	O0–O1 O2–O3	Pairs of dedicated Active-Low 3-State output buffers. Each pair has common Output Enable control.
24–25 26–27	O4–O5 O6–O7	Pairs of dedicated Active-High 3-State output buffers. Each pair has common Output Enable control.

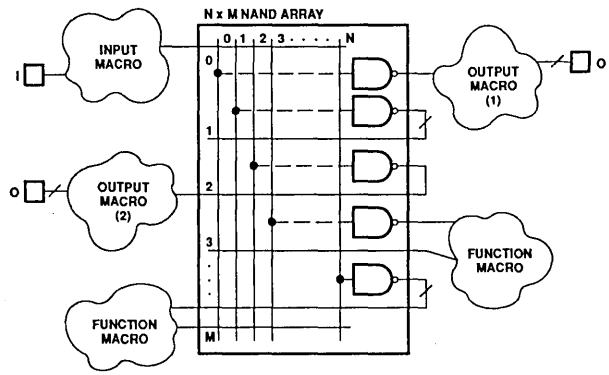


Figure 2. PML Fundamental Architecture

PLHS501 Programmable Macro Logic primer AN29

THE PLHS501 PML

The PLHS501 architecture in Figure 3 exhibits an exquisite logic tool. The device provides a combination of 72 NAND terms, 24 dedicated inputs (I0–I23), eight bidirectional I/O's (B0–B7), eight exclusive-OR outputs (X0–X7), and eight dedicated outputs (O0–O7). Figure 4 shows the PLHS501 logic diagram and Table 1 illustrates the functional breakdown of the PLHS501 I/O pins.

Since the output of each NAND term feeds back to the inputs of the NAND array, intricate logic functions can be implemented without wasting valuable I/O pins. For example, in order to implement an internal 'RS' latch in a combinational PAL/PLD, at least two inputs and two outputs are required. The same internal latch can be configured by the PLHS501 without using any I/O pins.

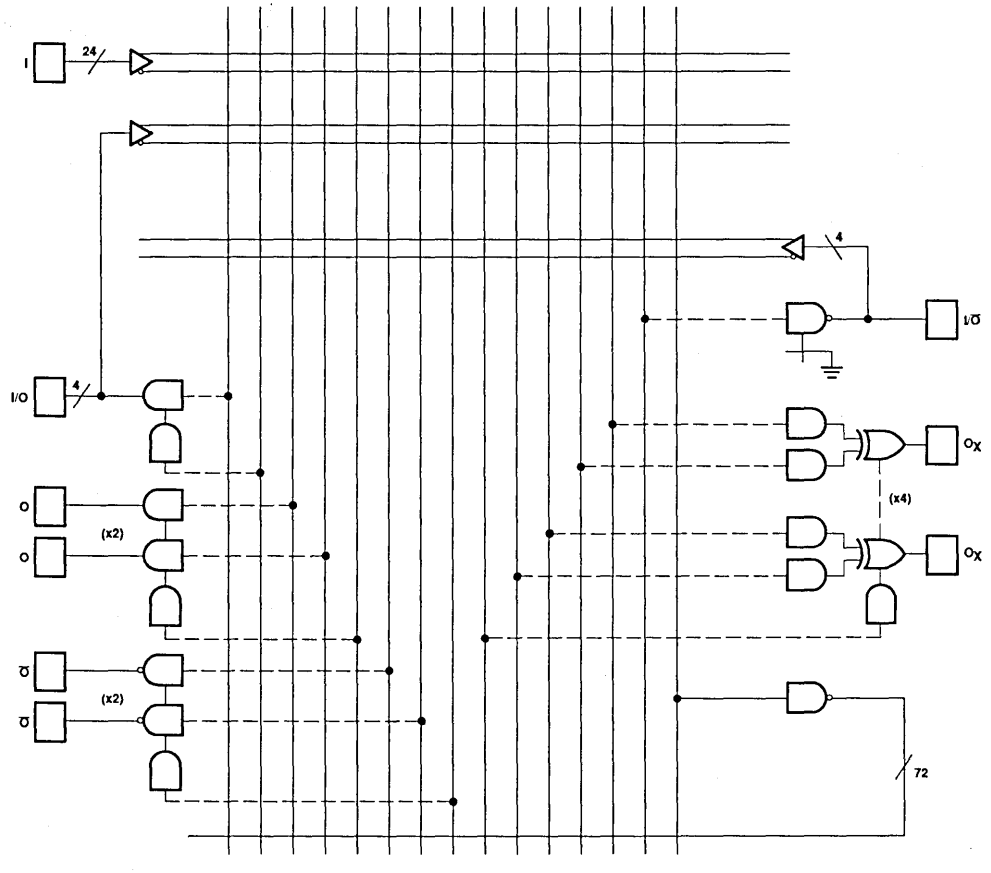


Figure 3. PLHS501 Logic Diagram

PLHS501 Programmable Macro Logic primer AN29

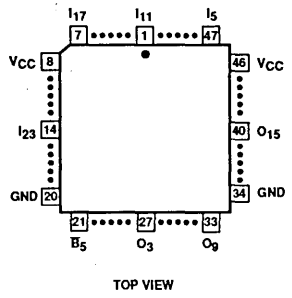
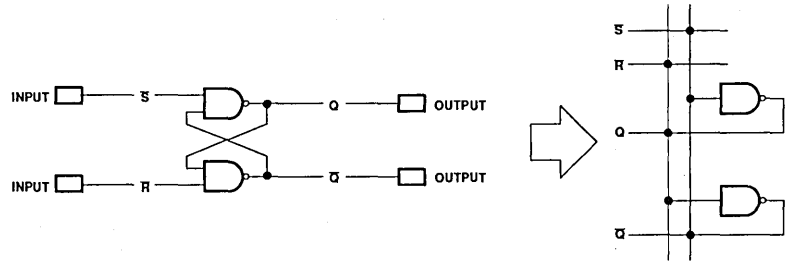


Figure 4. PLHS501 Pin Assignments

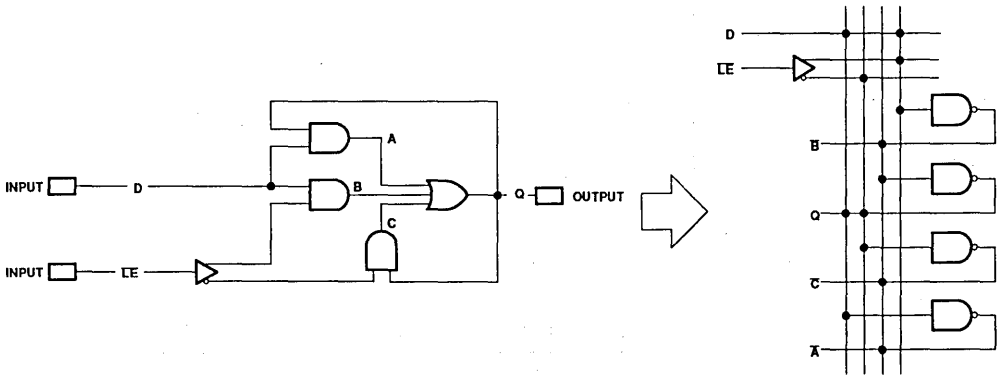
Figure 5 illustrates how 'RS' and 'D' latches are implemented in the PLHS501.

Another eminent application of the PLHS501 is in generating asynchronous state machines.

The blend of internal feedback paths together with the abundant number of gates makes this device suitable for designing asynchronous state machines which employ propagation delays of feedback paths as memory elements as shown in Figure 6. (See Reference 2.)



a. RS Latch



b. D Latch

Figure 5. R/S and D Latch Implementation with PML

PLHS501 Programmable Macro Logic primer AN29

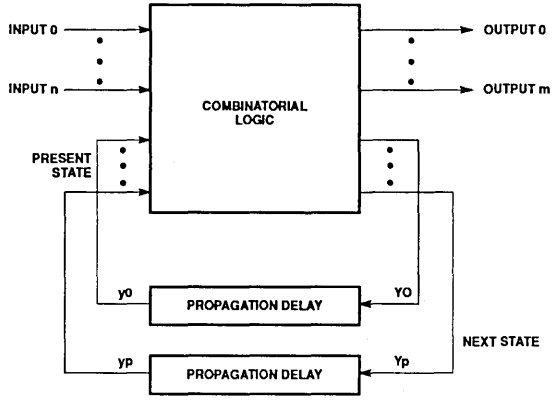


Figure 6. Model of Asynchronous State Machines Using Propagation Delays of the Feedback Path as Memory Elements

PML DEVELOPMENT SOFTWARE

Programmable logic development software has become an integral part of the PLD design process. Without software tools PLDs become perplexing devices which are inconvenient to use. Development software enables the user to take full advantage of the programmable logic's resources. The complexity of the PML devices makes software an indispensable element in the design process.

The SNAP PLD design software has been developed for Signetics programmable logic devices. PML design and development is fully supported by SNAP.

The foundation for the above modules is an AMAZE standard fuse file (STD). The STD file is the common means of communication between all the modules. The following paragraphs briefly explain the implementation of PML in the AMAZE structure.

PLHS501 DESIGN EXAMPLE

The following example intends to manifest the capabilities of the PLHS501. Figure 8 shows a system formed with TTL logic. The system requirements make it imperative only to use discrete asynchronous latches. Thus, none of the 7 latches in the system can be directly replaced by registers. The system is partitioned into two PLS173s and one PLS153. The specified PLD's are labeled with the same labels as those on the system schematic (Figure 7). SNAP automatically converts the system to the designated PLDs. The SNAP Boolean equation files are shown in Figures 8a, 8b, and 8c. Figure 9 shows the overall system implemented with PLDs. The logic condensation capabilities of PML makes it feasible to replace the whole system by a single PLHS501 (Figure 10). The PLHS501 in this design will still have ample space for any future additions.

The above example demonstrates only part of the PLHS501 capabilities. The introduction of PML devices and their immense logic power will pave the way for a new generation of efficient and elegant systems.

REFERENCES

1. Cavlan, Napoleone 1985. "Third Generation PLD Architecture Breaks AND-OR Bottleneck", WESCON 1985 Conference Proceedings.
2. Wong, David K. "Third Generation PLD Architecture and its Applications", Electro 1986 Conference Proceedings.

PLHS501 Programmable Macro Logic primer AN29

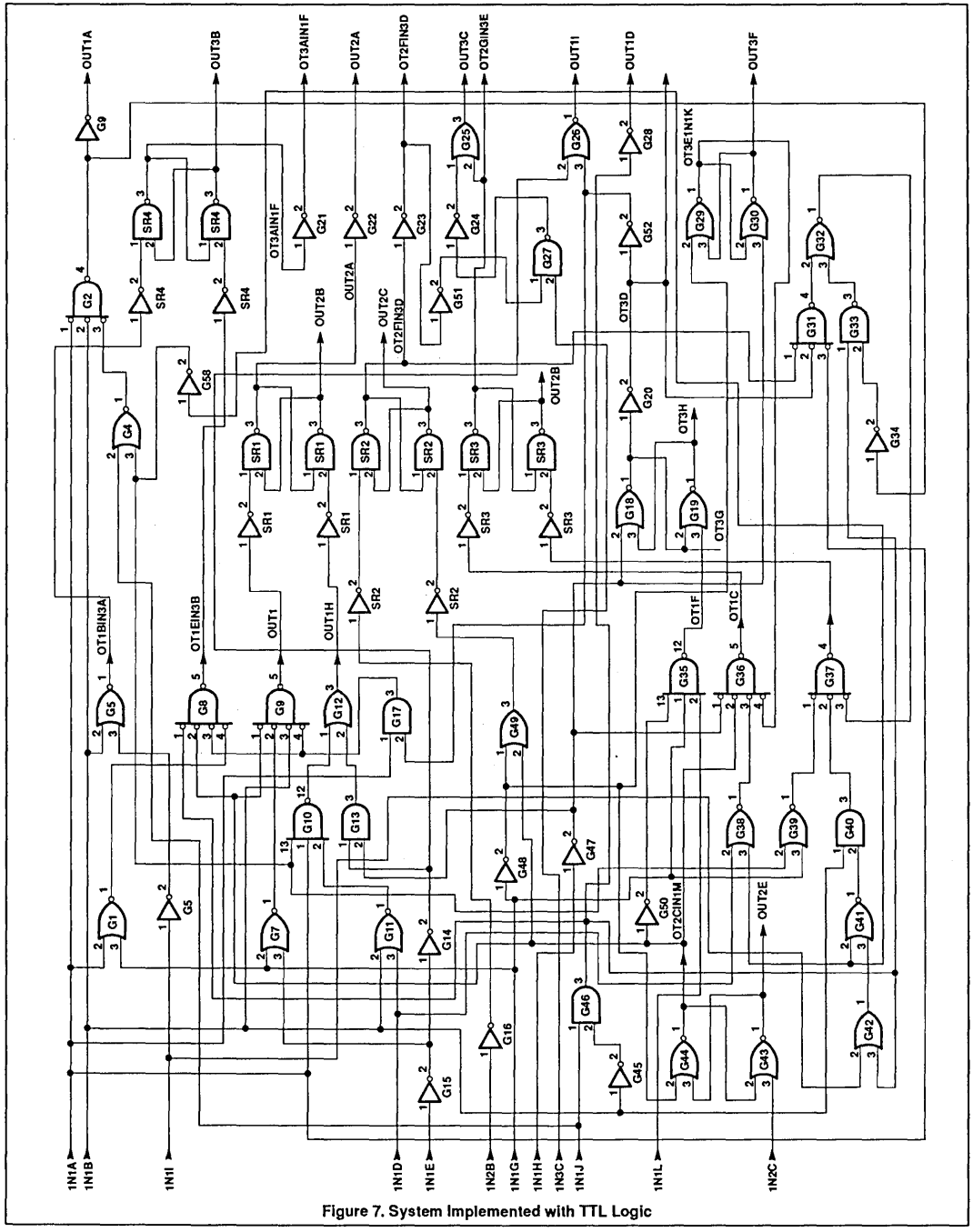


Figure 7. System Implemented with TTL Logic

```
File name: PART1
Date : 12/10/1986
Time : 18:25:54

###################### P I N   L I S T ######################

   LABEL     ** FNC **PIN --------- PIN** FNC **    LABEL
IN1A         ** I   **  1-|        |-24 ** +5V **VCC
IN1B         ** I   **  2-|        |-23 ** O   **OUT1I
OT3G         ** I   **  3-|        |-22 ** O   **OUT1H
IN1D         ** I   **  4-|   P    |-21 ** O   **OUT1G
IN1E         ** I   **  5-|   L    |-20 ** O   **OT1F
OT3AIN1F     ** I   **  6-|   S    |-19 ** O   **OT1EIN3B
IN1G         ** I   **  7-|   1    |-18 ** O   **OUT1D
IN1H         ** I   **  8-|   7    |-17 ** O   **OT1C
IN1I         ** I   **  9-|   3    |-16 ** O   **OT1BIN3A
IN1J         ** I   ** 10-|        |-15 ** O   **OUT1A
OT3EIN1K     ** I   ** 11-|        |-14 ** I   **OT2CIN1M
GND          ** 0V  ** 12-|        |-13 ** I   **IN1L
                          ---------

File Name : PART1
Date : 12/10/1986
Time : 18:26:56

@DEVICE TYPE
 PLS173
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
 PART1
@DESCRIPTION
@I/O DIRECTION
@LOGIC EQUATION
     OUT1A =/IN1A*/IN1B*(IN1J+/OT3AIN1F);
     OT1BIN3A =/IN1B*IN1I;
     OT1C =/IN1H+OT2CIN1M+/IN1D*/OT3AIN1F+OT3EIN1K;
     OUT1D =IN1B+/IN1J;
     OT1EIN3B =/IN1B*IN1J+OT2CIN1M+/OT3G*IN1A+/IN1A*/IN1G;
     OT1F =OT2CIN1M*/IN1G*/IN1L;
     OUT1G =OT2CIN1M+/IN1G*IN1E+IN1B+/OT3G*IN1A;
     OUT1H =/IN1H*(IN1E)+OT3AIN1F+/IN1A+IN1B+IN1D;
     OUT1I =(/IN1E)*OT3G;
```

Figure 8a. Part 1: PLS173

PLHS501 Programmable Macro Logic primer AN29

```
File Name : PART 1
Date : 12/10/1986
Time : 18:27:40

Cust/Project - PART 1
Date         -
Rev/I. D.    -

PLS173                                                  !     POLARITY      !
```

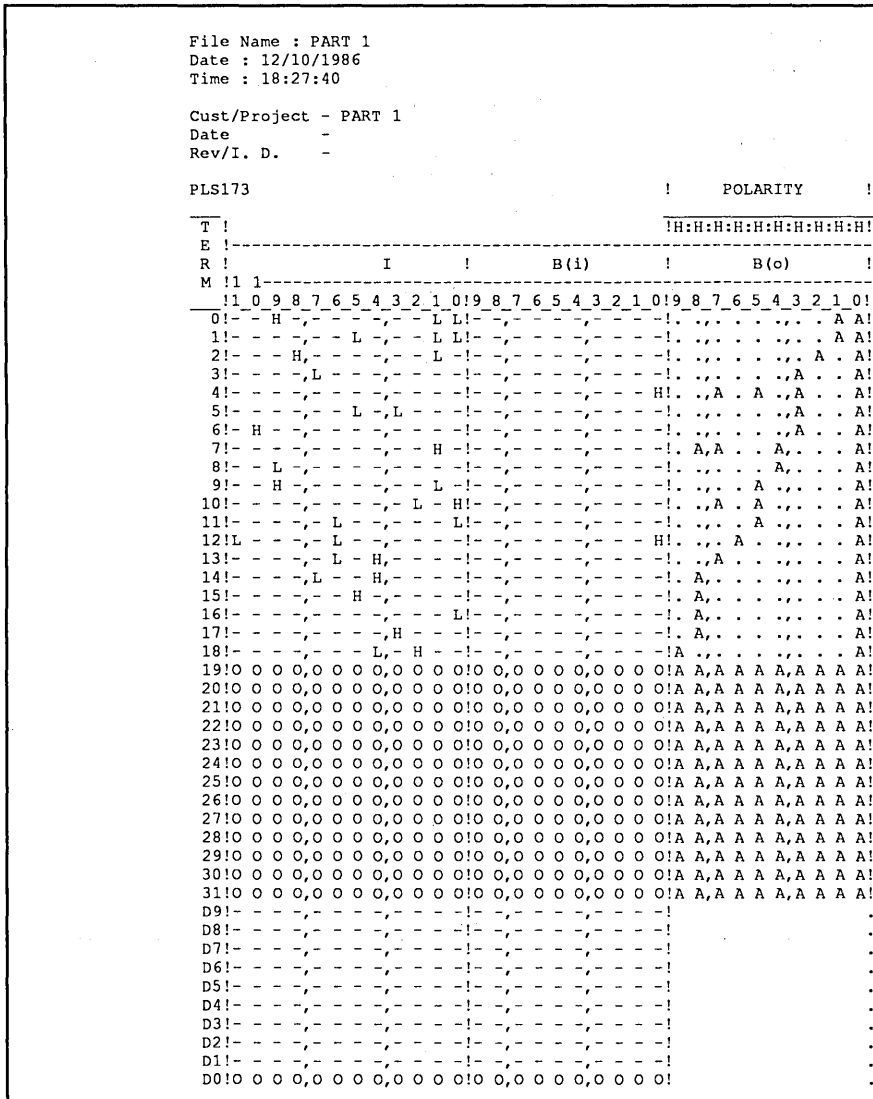


Figure 8a. Part 1: PLS173 (Continued)

```
File name: PART2
Date : 12/10/1986
Time : 18:30:42

#################### P I N   L I S T ####################

   LABEL     ** FNC **PIN --------- PIN** FNC **    LABEL
IN1A         ** I   **  1-|        |-24 ** +5V **VCC
IN2B         ** I   **  2-|        |-23 ** B   **N/C
IN2C         ** I   **  3-|        |-22 ** B   **N/C
IN1G         ** I   **  4-|   P    |-21 ** O   **OT2GIN3E
IN1I         ** I   **  5-|   L    |-20 ** O   **/OT2FIN3D
IN1J         ** I   **  6-|   S    |-19 ** O   **OT2CIN1M
IN1B         ** I   **  7-|   1    |-18 ** O   **OUT2E
OUT1G        ** I   **  8-|   7    |-17 ** O   **OUT2D
OUT1H        ** I   **  9-|   3    |-16 ** O   **OUT2C
OT1C         ** I   ** 10-|        |-15 ** O   **OUT2B
OT3AIN1F     ** I   ** 11-|        |-14 ** /O  **/OUT2A
GND          ** 0V  ** 12-|        |-13 ** I   **OT3G
                          ---------

File Name : PART2
Date : 12/10/1986
Time : 18:31:49

@DEVICE TYPE
 PLS173
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
 PART2
@DESCRIPTION
@I/O DIRECTION
@LOGIC EQUATION
     /OUT2A =/((OUT2B*/OUT1G));
     OUT2B =OUT1H+OUT2A;
     OUT2C =OT2CIN1M+/IN1G+OT2FIN3D;
     OUT2D =OT3AIN1F*/IN1G+(/OT3AIN1F*((IN1B+/IN1J)*/IN1I))*IN1B
            +(OT2FIN3D*/OT3G*/IN1A)*((/IN1A*/IN1B*(IN1J+/OT3AIN1F))
            *(/IN1B*IN1J))+/OT2GIN3E;
     OUT2E =/OT2CIN1M*/IN2C;
     OT2CIN1M =IN1G*/OUT2E;
     /OT2FIN3D =/((OUT2C*(IN2B)));
     OT2GIN3E =/OUT2D+OT1C;
```

Figure 8b. Part 2: PLS173

```
File Name : PART 2
Date : 12/10/1986
Time : 18:32:35

Cust/Project  - PART 2
Date          -
Rev/I. D.     -

PLS173                                         !      POLARITY       !
                                               _____________________
 T !                                           !H:H:H:H:H:H:H:H:H:L!
 E !-----------------------------------------------------------------
 R !             I           !       B(i)        !       B(o)        !
 M !1 1--------------------------------------------------------------
___!1_0_9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!
  0!- - - -,L - - -,- - - -!- -,- - - -,- - H -!A A,. . . .,. . . A!
  1!- - - H,- - - -,- - - -!- -,- - - -,- - - -!A A,. . . .,. . A .!
  2!- - - -,- - - -,- - - -!- -,- - - -,- - - L!A A,. . . .,. . A .!
  3!- - - -,- - - -,- - - -!- -,- - H -,- - - -!A A,. . . .,. A . .!
  4!- - - -,- - - -,L - - -!- -,- - - -,- - - -!A A,. . . .,. A . .!
  5!- - - -,- - - -,- - - -!- -,- L - -,- - - -!A A,. . . .,. A . .!
  6!- H - -,- - - -,L - - -!- -,- - - -,- - - -!A A,. . . .,A . . .!
  7!- L - -,- H - L,- - - -!- -,- - - -,- - - -!A A,. . . .,A . . .!
  8!- L - -,- H L L,- - - -!- -,- - - -,- - - -!A A,. . . .,A . . .!
  9!L L - -,- L H -,- - - L!- -,- L - -,- - - -!A A,. . . .,A . . .!
 10!L - - -,- L H -,- - - L!- -,- L - -,- - - -!A A,. . . .,A . . .!
 11!- - - -,- - - -,- - - -!- -,L - - -,- - - -!A A,. . . .,A . . .!
 12!- - - -,- - - -,- L - -!- -,- - L -,- - - -!A A,. . . A,. . . .!
 13!- - - -,- - - -,H - - -!- -,- - - L,- - - -!A A,. . A .,. . . .!
 14!- - - -,- - - -,- - - -!- -,- - - -,- L - -!A A,. A . .,. . . .!
 15!- - - -,- - - -,- - L -!- -,- - - -,- - - -!A A,. A . .,. . . .!
 16!- - - -,- - - -,- - - -!- -,- - - -,L - - -!A A,A . . .,. . . .!
 17!- - H -,- - - -,- - - -!- -,- - - -,- - - -!A A,A . . .,. . . .!
 18!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 19!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 20!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 21!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 22!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 23!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 24!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 25!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 26!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 27!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 28!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 29!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 30!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 31!O O O O,O O O O,O O O O!O O,O O O O,O O O O!A A,A A A A,A A A A!
 D9!O O O O,O O O O,O O O O!O O,O O O O,O O O O!                   .
 D8!O O O O,O O O O,O O O O!O O,O O O O,O O O O!                   .
 D7!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D6!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D5!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D4!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D3!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D2!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D1!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .
 D0!- - - -,- - - -,- - - -!- -,- - - -,- - - -!                   .

    O O O O O I I I I I I I N N O / O O O O O / N N O / O O O O O /
    T T T U U N N N N N N N / / T O T U U U U O / / T O T U U U U O
    3 3 1 T T 1 1 1 1 2 2 1 C C 2 T 2 T T T T U C C 2 T 2 T T T T U
    G A C 1 1 B J I G C B A     G 2 C 2 2 2 2 T     G 2 C 2 2 2 2 T
      I   H G                   I F I E D C B 2     I F I E D C B 2
      N                         N I N         A     N I N         A
      1                         3 N 1               3 N 1
      F                         E 3 M               E 3 M
                                  D                   D
```

Figure 8b. Part 2: PLS173 (Continued)

```
File Name : part3
Date : 12/10/1986
Time : 8:16:10

###################### P I N   L I S T ######################

   LABEL     ** FNC **PIN --------- PIN** FNC **    LABEL
OT1BIN3A     ** I   **  1-|        |-20 ** +5V **VCC
OT1EIN3B     ** I   **  2-|        |-19 ** B   **N/C
IN3C         ** I   **  3-|   P    |-18 ** B   **N/C
OT2FIN3D     ** I   **  4-|   L    |-17 ** O   **OT3H
OT2GIN3E     ** I   **  5-|   S    |-16 ** /O  **OT3G
OT1F         ** I   **  6-|   1    |-15 ** O   **OUT3F
IN1L         ** I   **  7-|   5    |-14 ** O   **OT3EIN1K
IN1H         ** I   **  8-|   3    |-13 ** O   **OUT3D
OT3AIN1F     ** /O  **  9-|        |-12 ** O   **OUT3C
GND          ** 0V  ** 10-|        |-11 ** I   **IN1G
                          ---------


File Name : part3
Date : 12/10/1986
Time : 8:16:14

@DEVICE TYPE
PLS153
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
@COMMON PRODUCT TERM  "CPT_label = (expression)"
@I/O DIRECTION  " D0 .. Dn "
@LOGIC EQUATION
OT3AIN1F  =  /(/OT1BIN3A * OUT3D);
OUT3C     =   (IN3C * /OT2FIN3D) + OT2GIN3E;
OUT3D     =  /(OT3AIN1F * /OT1EIN3B);
OT3EIN1K  =  /(/IN1G + OUT3F);
OUT3F     =  /(OT3EIN1K + /IN1H);
OT3G      =  /(IN1H + OT3H);
OT3H      =  /(OT3G + OT1F);
```

Figure 8c. Part 3: PLS153

PLHS501 Programmable Macro Logic primer AN29

```
File Name : part3
Date : 12/10/1986
Time : 8:16:17

Cust/Project -
Date        -
Rev/I. D.   -

PLS153                                          !     POLARITY      !

 T !                                            !H:H:H:L:H:H:H:H:L!
 E !-----------------------------------------------------------------
 R !       I        !        B(i)        !        B(o)        !
 M !-----------------------------------------------------------------
   !7 6 5 4 3 2 1 0!9 8 7 6 5 4 3 2 1 0!9 8 7 6 5 4 3 2 1 0!
 0!- - - -,- - - L!- -,- - - -,H - - -!A A,. . . .,. . A A!
 1!- - - -,L H - -!- -,- - - -,- - - -!A A,. . . .,. A A .!
 2!- - - H,- - - -!- -,- - - -,- - - -!A A,. . . .,. A A .!
 3!- - - -,- - - -!- -,- - - -,- - - L!A A,. . . .,A . A .!
 4!- - - -,- - H -!- -,- - - -,- - - -!A A,. . . .,A . A .!
 5!- - - -,- - - -!- -,- - L -,- - H -!A A,. . . A,. . A .!
 6!H - - -,- - - -!- -,- - - L,- - - -!A A,. . A .,. . A .!
 7!H - - -,- - - -!- -,- - - -,- - - -!A A,. A . .,. . A .!
 8!- - - -,- - - -!- -,H - - -,- - - -!A A,. A . .,. . A .!
 9!- - L -,- - - -!- -,- L - -,- - - -!A A,A . . .,. . A .!
10!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
11!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
12!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
13!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
14!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
15!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
16!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
17!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
18!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
19!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
20!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
21!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
22!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
23!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
24!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
25!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
26!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
27!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
28!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
29!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
30!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
31!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
D9!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
D8!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
D7!- - - -,- - - -!- -,- - - -,- - - -!                    .
D6!- - - -,- - - -!- -,- - - -,- - - -!                    .
D5!- - - -,- - - -!- -,- - - -,- - - -!                    .
D4!- - - -,- - - -!- -,- - - -,- - - -!                    .
D3!- - - -,- - - -!- -,- - - -,- - - -!                    .
D2!- - - -,- - - -!- -,- - - -,- - - -!                    .
D1!0 0 0 0,0 0 0 0!0 0,0 0 0 0,0 0 0 0!                    .
D0!- - - -,- - - -!- -,- - - -,- - - -!                    .

   I I O O O I O O N N O O O O O O I O N N O O O O O O I O
   N N T T T N T T / / T T U T U U N T / / T T U T U U N T
   1 1 1 2 2 3 1 1 C C 3 3 T 3 T T T 1 3 C C 3 3 T 3 T T 1 3
   H L F G F C E B     H G 3 E 3 3 G A     H G 3 E 3 3 G A
         I I   I I         F I D C   I         F I D C   I
         N N   N N           N       N           N       N
         3 3   3 3           1       1           1       1
         E D   B A           K       F           K       F
```

Figure 8c. Part 3: PLS153 (Continued)

PLHS501 Programmable Macro Logic primer AN29

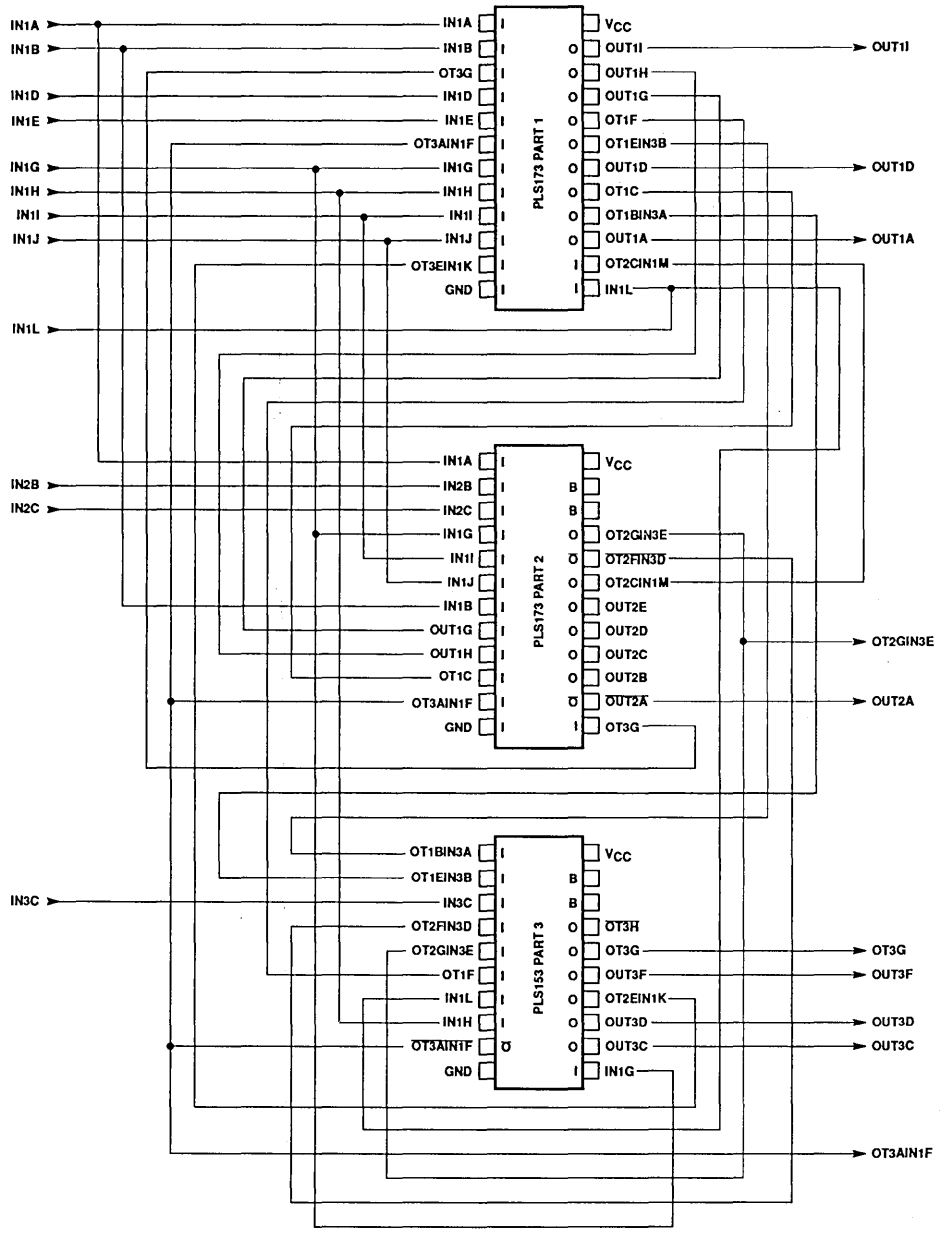


Figure 9. System in Figure 8 Implemented with PLDs

PLHS501 Programmable Macro Logic primer AN29

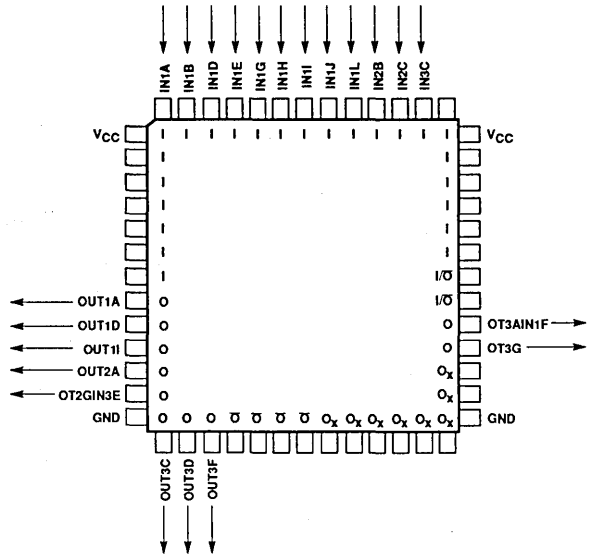


Figure 10. System In Figure 10 Implemented with PLHS501

Designing with Programmable Macro Logic

Programmable Logic Devices

INTRODUCTION TO PML DESIGN CONCEPTS

Programmable Macro Logic, an extension of the Programmable Logic Array (PLA) concept combines a programming or fuse array with an array of wide input NAND gates wherein each gate folds back upon itself and all other such NAND gates. This is called a foldback NAND structure and its basic elements have been outlined previously (Cavlan[1], Wong[2], Gheissari and Safari[3]).

The choice of an internal NAND logic cell is appropriate because the cell is functionally complete, requiring but a single cell type to generate any Boolean function. A cell within the PLHS501 may be configured to accommodate from one to 32 inputs from the outside world, and up to 72 inputs from within the chip. Because the user can select either direct or inverted input variables, and either a direct or complemented output, the NAND function can generate, with a single pass through the programming array, the basic four logic functions of AND, OR, NAND, NOR. all these basic functions, can be extremely wide, of course (see Figure 1.1). This convenient structure allows efficient exploitation of all widely used minimization techniques (Karnaugh Maps, Quine-McClusky, Boolean Algebra, etc.).

The obvious extensions to additional combinational functions for decoding, multiplexing and general Boolean functions is straightforward. Adding feedback to the system expands the range of realizable functions to include sequential as well as combinational functions. Figure 1.2 illustrates the basic arrangement of the PLHS501. Because of the large number of inputs each NAND gate has available, logic functions that require several levels of conventional 4 or 8 input gates may be able to be reduced to 1 or 2 levels. However, it is important to realize that unlike AND-OR PLD architectures, more than 2 levels of logic may be implemented in the PLHS501 without wasting output or input pins. Up to 72 levels of logic may be implemented due to each of the 72 foldback NAND gates.

So far, the concept of a "macro" is still not evident. Two ways for the generation of a macro exist—namely, hard and soft. Borrowing from the concept in computer programming wherein a section of code (called a macro) is repeated every time its use is required, we can establish subfunctions which can be repeated each time required. The user defined or soft macro can be one which will generate a function by fused interconnect. When a fixed design function is provided, it is a hard macro. This may be an optimized structure like a flip-flop or an adder, or some other function which is generated on the foundation, by the manufacturer. Soft macros are seldom optimized or precisely consistent, but hard macros are both optimized and unalterable.

When a user function for a particular use is isolated, defined and repetition of the function is required, special software constructs are provided which will allow it to be defined as a higher performance and functional density, and array of choices which contain optimized functions or hard macros will be offered in successor chips. In particular, the PLHS502 (described in Section 4) includes an array of flip-flops for high performance state machine design.

Optimizing combinational functions in PML consists largely in making choices and trade-offs. For single output logic functions, the choice is obvious from the truth table. If a particular function's truth table has fewer entries that are logical zeroes than logical ones, product of sums should be chosen and the appropriate OR-AND structure generated. Otherwise, the usual sum of products should be chosen, minimizing as usual, before dropping into the two level AND-OR structure (using the NAND-NAND realization). Combining the availability of inversion at the input and output of the chip, the NAND-NAND structure can perform either the OR-AND or the AND-OR rendition of a function with equal logic levels. The designer needs only to choose the optimal rendition to suit his needs (see Table 1.1). Truth tables with 50% ones can use either version at the designers whim unless other uses arise.

PERFORMANCE

The PLHS501 (Figure 1.2) is a high speed, oxide isolated, vertically fused PML device containing 72 internal NAND functions which are combined with 24 dedicated outputs. A large collection of applications, both combinational and sequential, may be configured using this part which looks roughly like a small, user definable gate array. For the sake of clarity, worst case passing a signal from an input, making one pass through the NAND array (output terms) and exiting an output takes around 25 nanoseconds with each incremental pass through the NAND foldback array taking about 8 nanoseconds.

Designing with Programmable Macro Logic

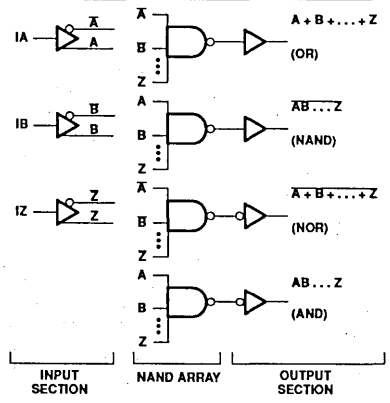


Figure 1.1 PML Basic Functions

Table 1.1 Example Demonstration

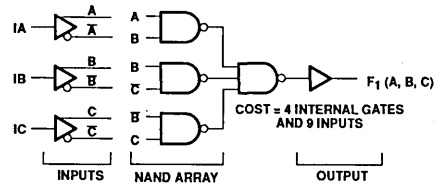
$F1(A, B, C) = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C + AB\overline{C} + ABC$

A	B	C	f1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

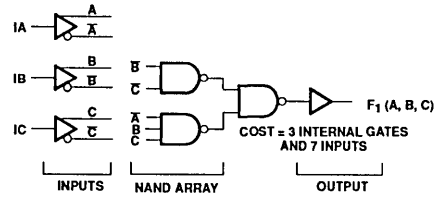
C \ AB	00	01	11	10
0	0	1	1	0
1	1	0	1	1

The optimal choice would be to generate the zero entries

If we group on the <u>one</u> entries we shall get: $AB + \overline{B}C + B\overline{C}$



If we group on the <u>zero</u> entries we get instead: $F_1 = (\overline{B} + \overline{C})(\overline{A} + B + C)$



Designing with Programmable Macro Logic

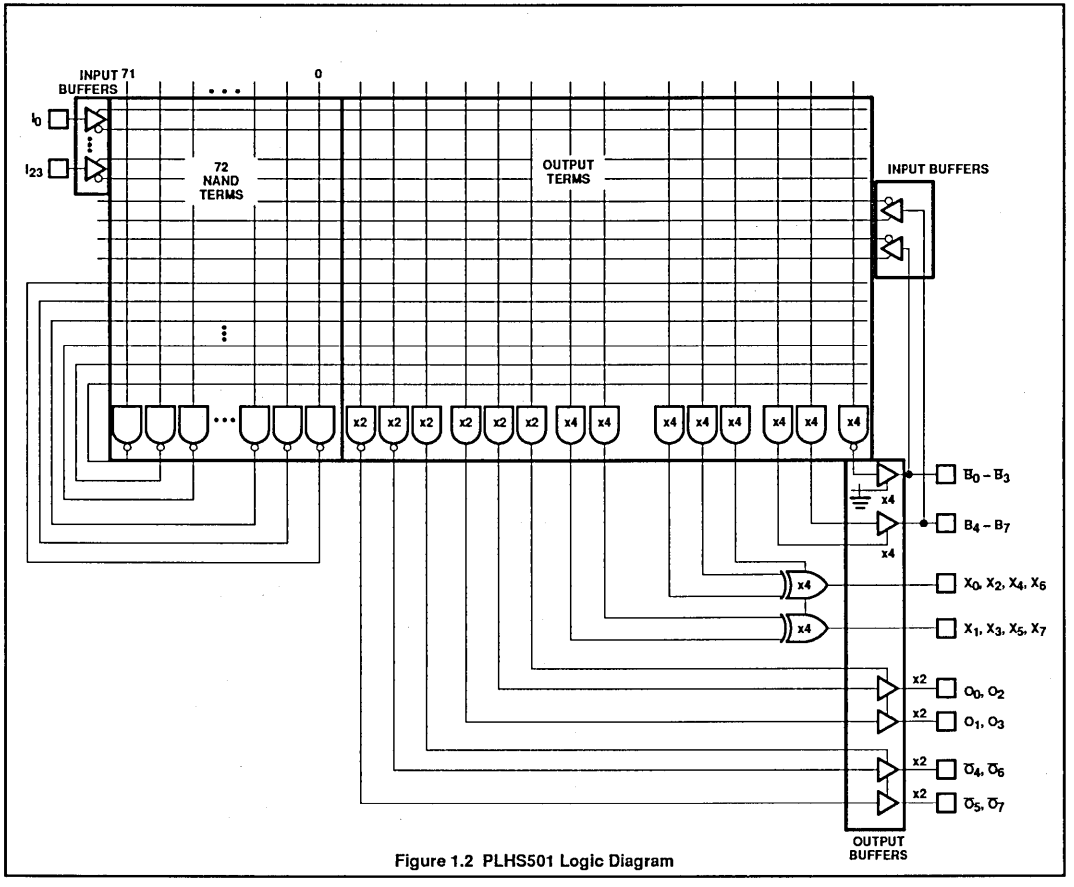


Figure 1.2 PLHS501 Logic Diagram

Designing with Programmable Macro Logic

The data sheet first lists some maximum propagation delays from an input, through a NAND output term and out through various output gates. Secondly, it lists maximum propagation delays from an input, through a NAND foldback term, through a NAND output term and out through the different output gates.

It is intriguing that subtracting one from the other yields a NAND foldback gate delay of 5 to 6ns when the worst case gate delay of an internal foldback gate is listed as 8ns. This is due to the fact that a gate has less of a delay when its output is falling (t_{PHL}) than when its output is rising (t_{PLH}). When passing a signal through two NAND gates one gate will have less of a delay than the other, and since the individual rise and fall delays are not specified, this causes the apparent discrepancy between the two delays.

Figure 1.3, Figure 1.4, Figure 1.5 and Figure 1.6 show graphically the timing paths listed in the PLHS501 data sheet.

PLHS501 TIMING

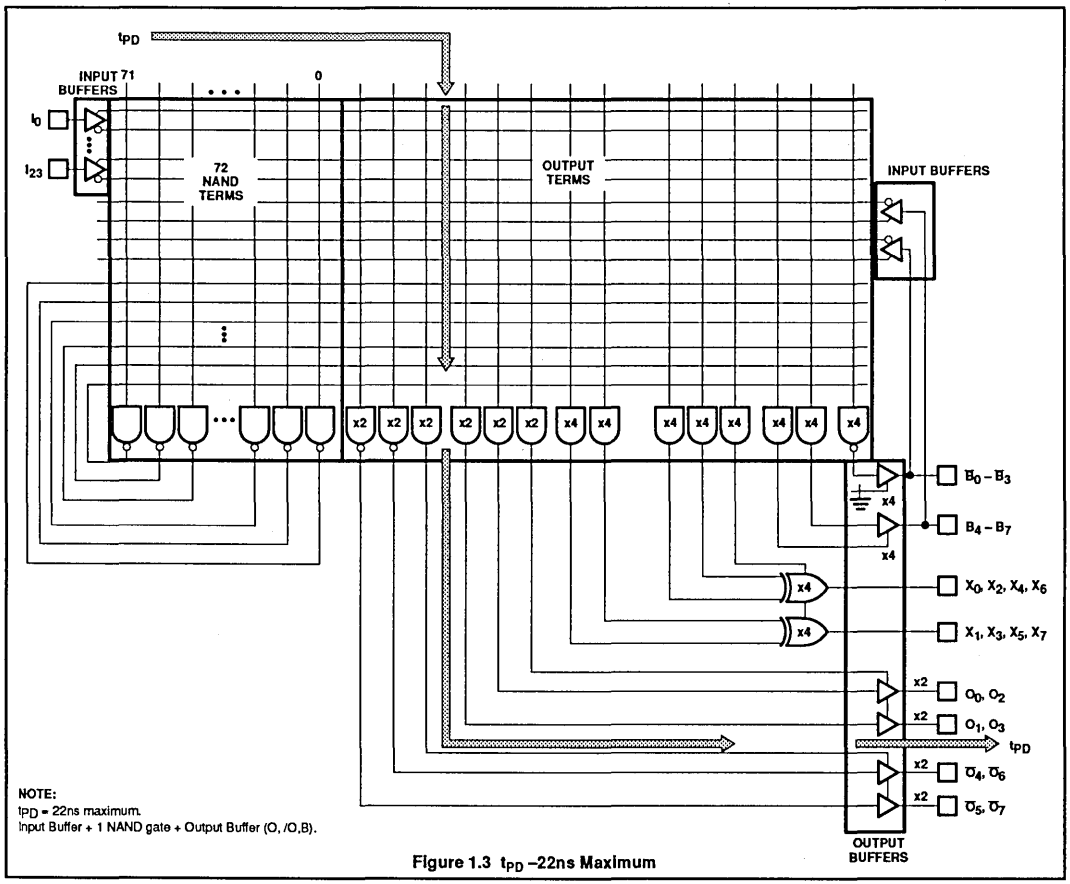


Figure 1.3 t_{PD} –22ns Maximum

Designing with Programmable Macro Logic

PLHS501 TIMING (Continued)

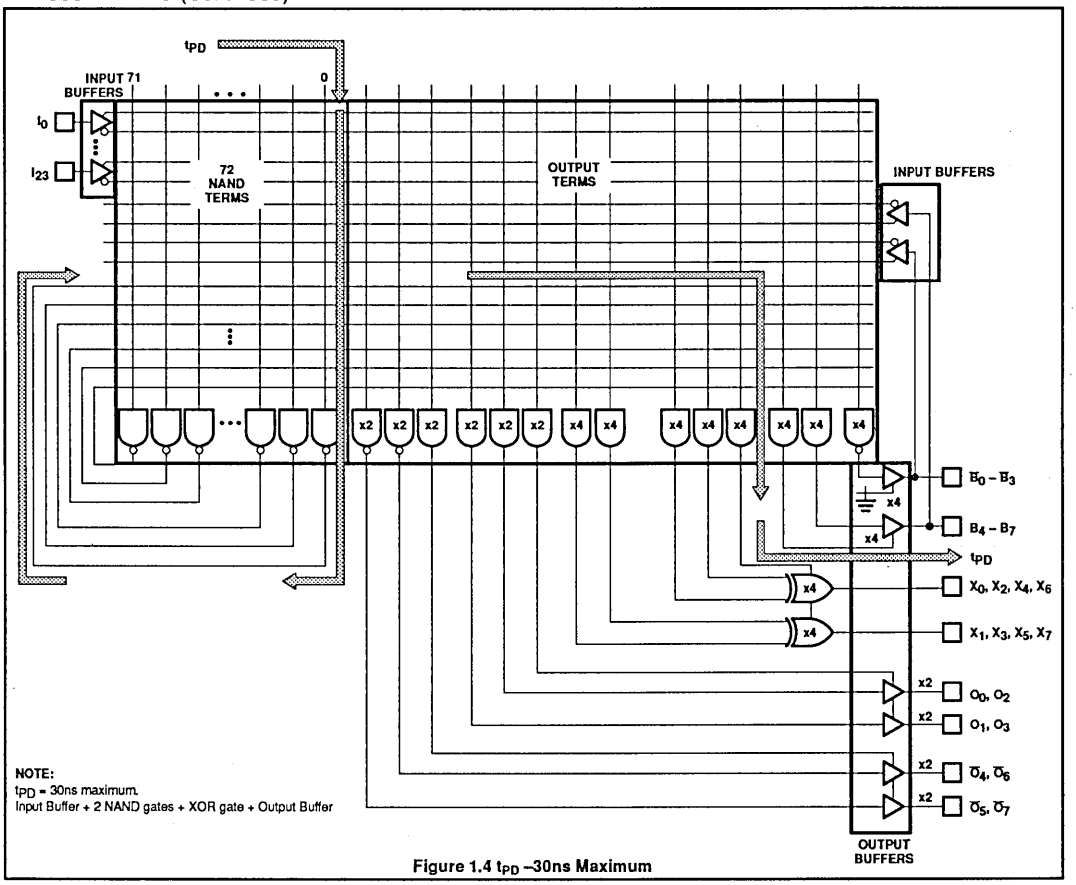


Figure 1.4 t_{PD} –30ns Maximum

Designing with Programmable Macro Logic

PLHS501 TIMING (Continued)

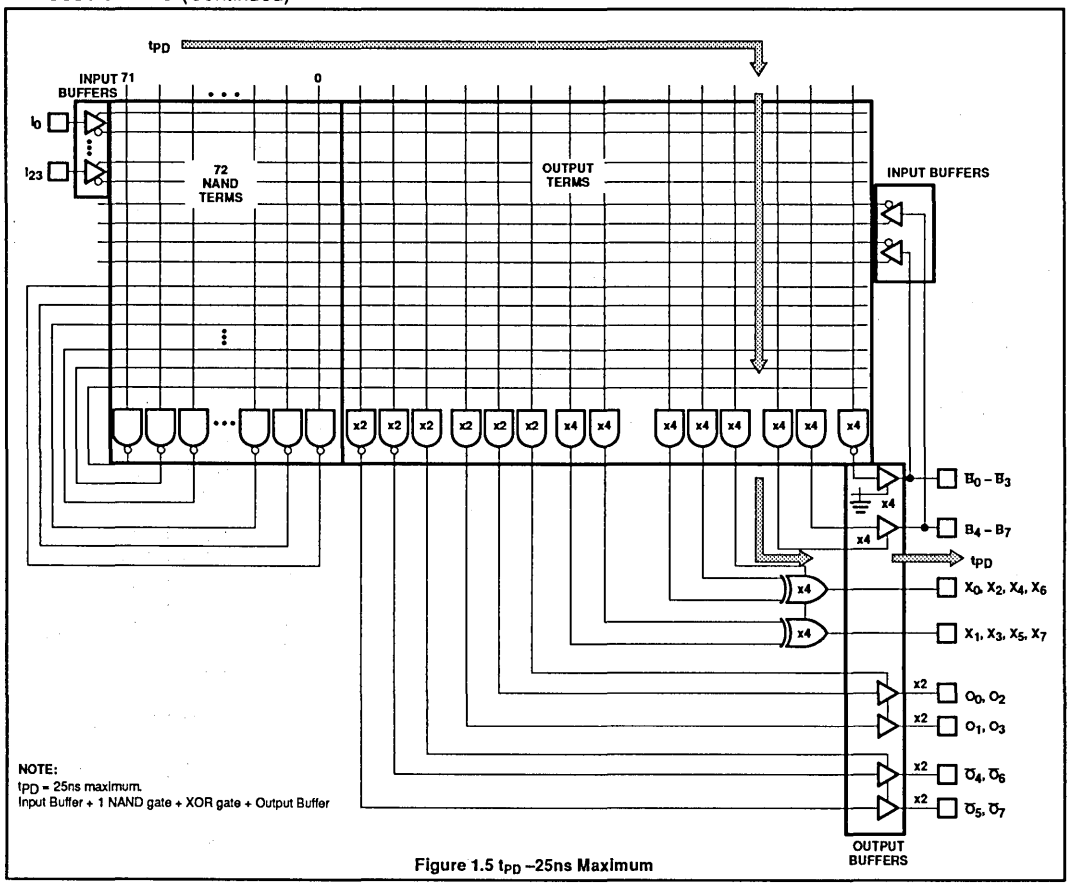


Figure 1.5 t_{PD} –25ns Maximum

Designing with Programmable Macro Logic

PLHS501 TIMING (Continued)

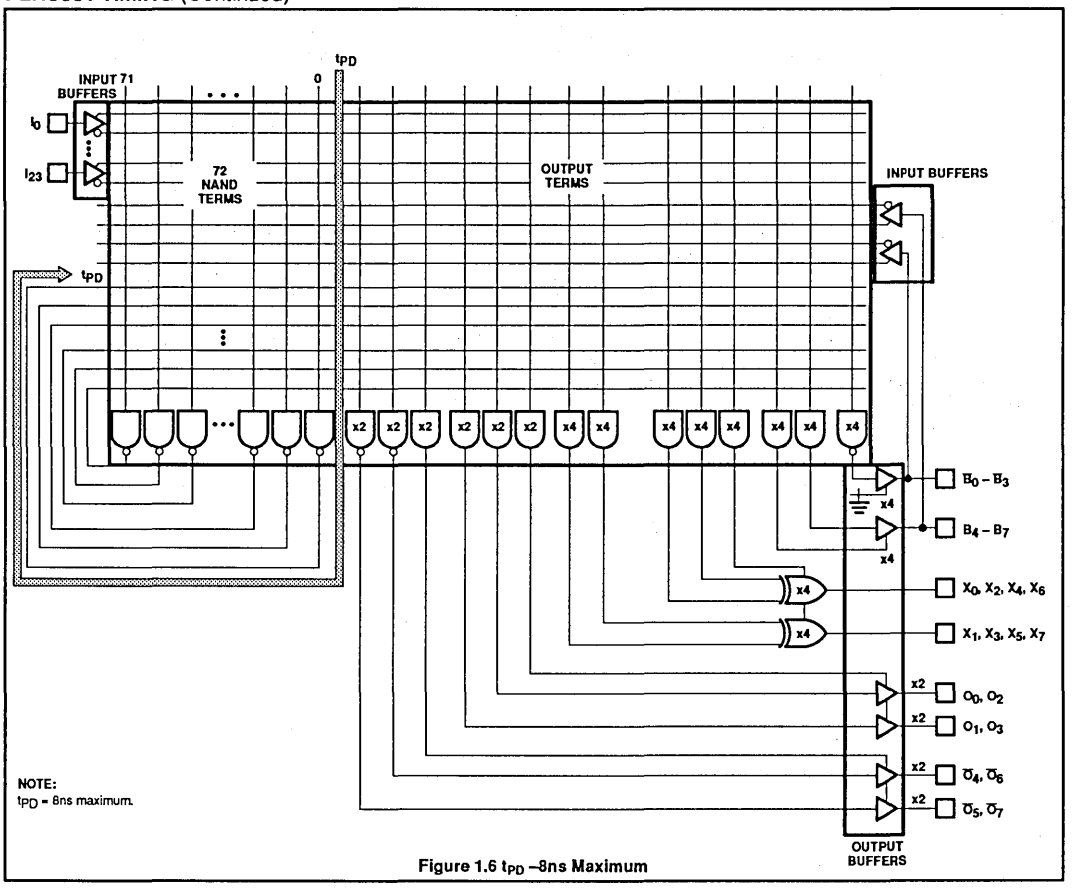


Figure 1.6 t_{PD} –8ns Maximum

Designing with Programmable Macro Logic

NAND GATE FLIP-FLOPS

Various types of flip-flops and latches may be constructed using the NAND gate building blocks of the PLHS501. A typical 7474 type of edge-triggered D flip-flop requires 6 NAND gates as shown in Figure 1.7.

No additional gates are required to implement asynchronous set and reset functions to the flip-flop. The equations necessary for AMAZE to implement the D flip-flop are shown in Figure 1.8. However, please note that the equations of Figure 1.8 define a D flip-flop configured as a divide by 2 (i.e., QN is connected to the data input) whereas Figure 1.7 shows a general case. Also note that flip-flops with some additional features may be constructed without using more than the six NAND gates. This is possible because of the large number of inputs associated with each NAND gate. For instance, a flip-flop may be required to have a clock gated by one or more signals. Using the PLHS501, it may be implemented by adding additional input signal names to NAND gate equations of gates #2 and #3 of Figure 1.7. If the data input is to the AND of several signals, extra inputs to NAND gate #4 may be used. Or if additional set or reset lines are required, they may be added simply by using more of the inputs of each NAND gate connected to the main set or reset.

Figure 1.10 shows two simulations of the same flip-flop. The first one is at a little less than maximum frequency, for clarity in following the waveforms, and the second is at the maximum toggling frequency. For these simulations each NAND gate has a maximum t_{PHL} or t_{PLH} of 8ns (which is the gate delay of a NAND gate in the PLHS501's foldback array). First of all, it can be seen from these simulations that for proper simulation or testing of such a device a set or reset input is mandatory. Both Q and QN outputs are unknown not matter what the inputs do, until they are put into a known state by either a set or reset input. Secondly, various timing parameters such as propagation delay, as well as setup and hold times may be determined.

Therefore, performance of the flip-flop depends a great deal on which gates in the PLHS501 are used, either NAND gates in the foldback array or output NAND gates, connected to bidirectional pins. As a test of the simulation, a D flip-flop connected as a divide by 2 was constructed using only the foldback NAND terms (see Figure 1.8). An output NAND terms was used to invert the QN output and drive an output buffer. The only inputs were the clock and a reset. The data input to the flop was driven internally by the QN output. According to the simulation, it was possible to drive the clock at a frequency of 25MHz and this small circuit also functioned at that frequency.

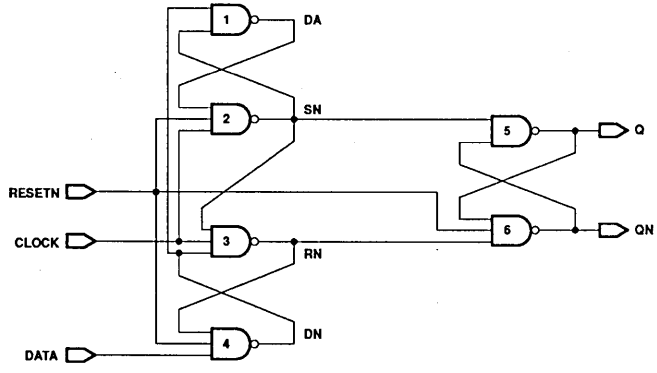


Figure 1.7 Edge-Triggered D Flip-Flop

Designing with Programmable Macro Logic

```
################## P I N   L I S T ##################

Left                                          Right
LABEL   ** FNC **PIN  _______  PIN** FNC **LABEL
VCC     ** +5V **  8-|       |-46 ** +5V **VCC
N/C     ** I   **  9-|       |-45 ** I   **N/C
N/C     ** I   ** 10-|       |-44 ** I   **N/C
N/C     ** I   ** 11-|   P   |-43 ** I   **N/C
N/C     ** I   ** 12-|   L   |-42 ** I   **N/C
CLK     ** I   ** 13-|   H   |-41 ** I   **N/C
RST     ** I   ** 14-|   S   |-40 ** /O  **N/C
N/C     ** I   ** 15-|   5   |-39 ** /O  **N/C
N/C     ** I   ** 16-|   0   |-38 ** /O  **N/C
N/C     ** I   ** 17-|   1   |-37 ** /O  **O0
N/C     ** I   ** 18-|       |-36 ** O   **N/C
N/C     ** O   ** 19-|       |-35 ** O   **N/C
GND     ** 0V  ** 20-|_______|-34 ** 0V  **GND

File Name: 1flop
Date: 12/3/1987
Time: 10:1:22
@DEVICE TYPE PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
Single D flip-flop connected as
divide-by-two for test
@INTERNAL NODE
CSN0,CRN0,CQ0,CQN0;
@COMMON PRODUCT TERM
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION
O0 = /(CQN0);
CSN0 = /(CLK*RST*(/(CSN0*(/[CQN0*RST*CRN0]))));
CRN0 = /(CSN0*CLK*(/[CQN0*RST*CRN0]));
CQ0 = /(CSN0*CQN0);
CQN0 = /(CRN0*CQ0*RST);
```

Figure 1.8 PLHS501 Test Flip-Flop

```
PLHS501
77666666666655555555554444444444333333333322222222211111111110000000000
10987654321098765432109876543210987654321098765432109876543210987654321O
IIIIIIIIIIIIIIIIIIIIIIIBBBBBBBB
22221111111111000000000000000000
32109876543210987654321076543210
--------------------------------   Col for   /B 0
................................................................A..
HH------------------------------   Col for    P 0
...............................................................A...
-H------------------------------   Col for    P 1
..............................................................A...A
H-------------------------------   Col for    P 2
.............................................................A...A.
--------------------------------   Col for    P 3
..............................................................A...A
H-------------------------------   Col for    P 4
................................................................AA.
--------------------------------   Col for    P 5
.................................................................A.A
```

Figure 1.9 Partial PLHS501 Fusemap Showing Test Flip-Flop Fusing

Designing with Programmable Macro Logic

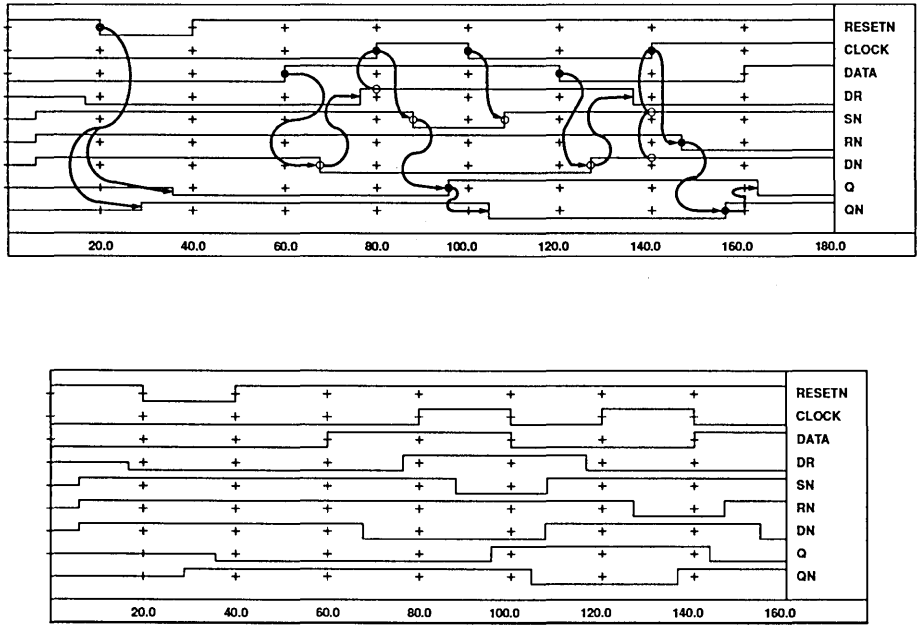


Figure 1.10 Waveforms of Test Flip-Flop

Designing with Programmable Macro Logic

Programmable Logic Devices

FUNCTIONAL FIT

In the late 1960's and early 1970's designers used SSI, MSI and small amounts of early LSI to generate logic solutions. Frustrated by the lack of wide input gates to accommodate a lot of product terms for two level solutions, they turned toward the budding ROM and PROM products. These devices relied on literally realizing a function by generating its truth table in silicon. The logic function had to have each logical one and zero realized distinctly as an entry for a particular combination of input variables, usually supplied on the address lines of the memory. Observing that many such truth tables were dense in ones or zeroes and sparse in the remainder, a cadre of initial manufacturers emerged with focus on supplying a programmable product with a few AND gates and OR gates which were versatile enough to compete against the ROM/PROM parts. The gimmick supplied these PLA manufacturers was to illustrate the functional equivalency of the PLA to the PROM by comparing the number of product terms (to be shortened to "p-terms") the PLA supplied and comparing this to the width and depth of available PROMs. P-terms became the "currency" of the PLA world and a designer only had to assess the equivalent number of Boolean product terms required by his function to determine whether a particular PLA was a suitable candidate for his design.

Almost in parallel, gate arrays became available. These provided an array of identical, fixed input gates (usually two input NANDs or NORs). These were generated in a regular fashion on substrate which has a fixed input/output pin arrangement. Also recognizing that all logic functions could be built from the appropriate two input gate, when interconnected correctly, manufacturers offered these devices to customers who required increased density.

The designer's responsibility was to generate what would ultimately be a metal interconnect pattern of his design. Special tools were required to allow an untrained system designer to do this successfully. Flop-flops, decoders, registers, adders, etc., could all be generated from the low level gate building blocks.

The currency of gate arrays became known as gate equivalent functions. That is with limited number of available gates on a substrate, the user needed to know precisely how many gates were used up, on a function by function basis, to generate each piece of his design. A D flip-flop requires about six gates, a D latch four, a 3 to 8 decoder takes about 14 gates and so forth. This allowed estimation regarding whether the function could conceivable be fit onto a particular substrate or not. Manufacturers had to offer multiple foundations to that a designer could be assured that his design would result in a working IC.

The classic method of estimating whether a logic function would fit into a PLA was to determine the number of I/O pads required and the number of product terms required to generate the logical function, then select the PLA. For a gate array, the required measure included the I/O pad arrangement but substituted the number of available gates to generate the logical function (usually by table lookup). In an attempt to reconcile the two measures, Hartman[4] has evolved a formula for his product line. A calculation using this method and developing an appropriate "exchange rate: is shown in Table 2.1 for the PLHS501 and PLHS502. An alternate method of generating an estimate is to consider the gate equivalent of generating, say for the PLHS501, a gate equivalent of the part in an optimistic functional configuration (72 occurrences of a 32 input NAND gate).

Figure 2.1 shows how this will result in over 2000 equivalent gates. Conversely, by stacking the NAND gates into D flip-flops, its least efficient function, the PLHS501 will have a gate equivalent of only about 100 gates.

The most rational method of assessing fit is to isolate functions and identify the correct configuration in terms of gates, to allow direct tally of the gates used, to generate the proposed configuration. Table 2.2 may assist in doing this analysis. Note that all basic gates require precisely one gate to generate the function. Also note the occurrence of functions in the table which could never be generated as standard ICs previously. The procedure is to tally the design against a total budget of 72 multiple input NAND gates.

Table 2.2 is illustrative only, and should by no means be taken as complete. It may be simply expanded by designing the proposed function with disregard to the usual restrictions on the number of inputs to a gate, realize the function as one, two, three, or more levels of interconnected logic and count the number of gate occurrences required. Special software has been provided to allow pyramided logic structures to be generated under the designer's control. These structures may, however, be no deeper than 72 levels for the PLHS501. Functions should be generated in accord with the guidelines mentioned before, for selecting an optimal 2 level logical solution.

It is an interesting observation that manufacturers of gate arrays and standard cell products which offer embedded PROMs, ROMs or RAMs have not successfully described these embedded functions in terms of equivalent gates, but rather resort to other means (such as divulging their relative area with respect to the area of a basic gate). There is, as yet, no standard in this arena.

Designing with Programmable Macro Logic

Table 2.1 Equivalency Ratio

Hartman's method is based on a CMOS gate array equivalency wherein 4 transistors constitute a 2 input NAND or NOR gate, equal to one gate. Thus, his "exchange rate" is as follows:

E.R. = 4 × # inputs
+9 × # FFs
+7 × # 3-State outputs
+(15 to 30) × # OR outputs from the AND/OR array.

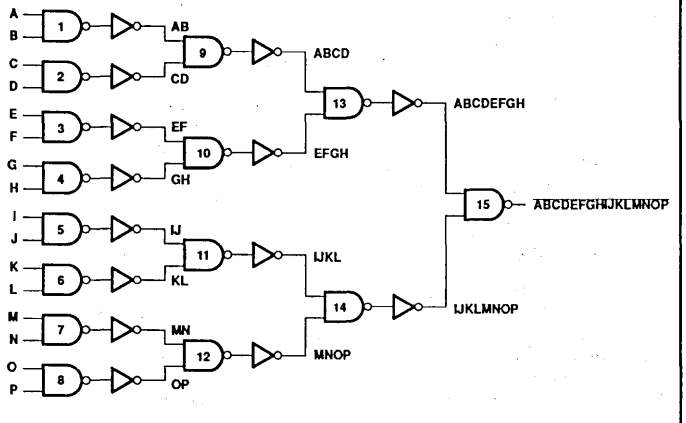
For the PLHS501: (using CMOS numbers which may be inappropriate)

E.R. = 4 × 32
+9 × 0
+7 × 24
+(15 to 30) × 50% of 72 feedbacks = 836 to 1376 gates

For the PLHS502:

E.R. = 4 × 32
+9 × 16
+7 × 12
+(15 to 30) × 50% of 64 feedbacks = 962 to 1502 gates

Being for two bipolar ICs, in this case, the method may be inappropriate, but may be taken as an estimating procedure.



NOTE:
Double this and add one for a 32 input NAND.

Figure 2.1 16 Input NAND Formed from 2 Input Gates

Designing with Programmable Macro Logic

Table 2.2 PLHS501 Gate Count Equivalents

FUNCTION	INTERNAL NAND EQUVALENT	COMMENTS
Gates		
NANDs	1	For 1 to 32 input variables
ANDs	1	For 1 to 32 input variables
NORs	1	For 1 to 32 input variables
ORs	1	For 1 to 32 input variables
Decoders		
3-to-8	8	Inverted inputs available
4-to-16	16	Inverted inputs available
5-to-32	32	Inverted inputs available (24 chip outputs only)
Encoders		
8-to-3	15	Inverted inputs, 2 logic levels
16-to-4	32	Inverted inputs, 2 logic levels
32-to-5	41	Inverted inputs, 2 logic levels, factored solution.
Multiplexers		
4-to-1	5	Inverted inputs available
8-to-1	9	
16-to-1	17	
27-to-1	28	Can address only 27 external inputs - more if internal
Flip-Flops		
D-type Flip-Flop	6	With asynchronous S-R
T-type Flip-Flop	6	With asynchronous S-R
J-K-type Flip-Flop	10	With asynchronous S-R
Adders		
8-bit	45	Full carry-lookahead (four levels of logic)
Barrel Shifters		
8-bit	72	2 levels of logic
Latches		
D-latch	3	2 levels of logic with one shared gate

Designing with Programmable Macro Logic

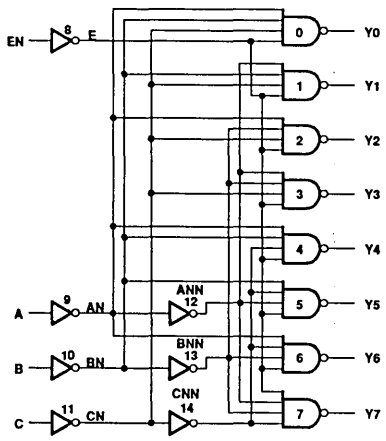
Programmable Logic Devices

DESIGN EXAMPLES

Most designers tend to view a PLD as a mechanism for collecting logical glue within a system. That is, those pieces which tie together the larger LSI microprocessors, controllers, RAMs, ROMs, UARTs, etc. However, there is a tendency of viewing a gate array as an entire system on a chip. PML based products will fit well in either casting as will be demonstrated by a series of small but straightforward examples. For starters, we shall examine how the fusing process embeds function, progress to glue-like decoding operations and finally demonstrate some coprocessor like functions as well as homemade "standard products".

The method of associating gates within the NAND foldback structure is depicted in Figure 3.1 wherein a simple three to eight decoder is fused into the array. The corresponding inputs are on the left and outputs at the top. This figure shows inputs and their inverse formed in the array resulting in a solution that requires 6 inverting NANDs that would probably be best generated at the input receivers. Hence, this diagram could be trimmed by six gates, down to eight to achieve the function. Figure 3.2 shows two consecutive D flip-flop fusing images. Note that asynchronous sets and resets may be achieved for free, in this version. In both Figures 3.1 and 3.2 the gates are numbered in a one-to-one arrangement. As well, the accompanying equations are in the format used by Signetics AMAZE design software. For clarity, consider the gate labeled 2A in Figure 3.1. Schematically, this is shown as a 3 input NAND. However, in the fused depiction, it combines from three intermediate output points with the dot intersect designation. Hence, all gates are drawn as single input NANDs whose inputs span the complete NAND gate foldback structure.

1 OF 8 DECODER/DEMULITPLEXER




```
@LOGIC EQUATION
AN = /A;
BN = /B;
CN = /C;
ANN = /AN;
BNN = /BN;
CNN = /CN;
E = /EN;
Y0 = / (AN * BN * CN * E);
Y1 = / ANN * BN * CN * E);
Y2 = / AN * BNN * CN * E);
Y3 = / ANN * BNN * CN * E);
Y4 = / AN * BN * CNN * E);
Y5 = / ANN * BN * CNN * E);
Y6 = / AN * BNN * CNN * E);
Y7 = / ANN * BNN * CNN * E);
```

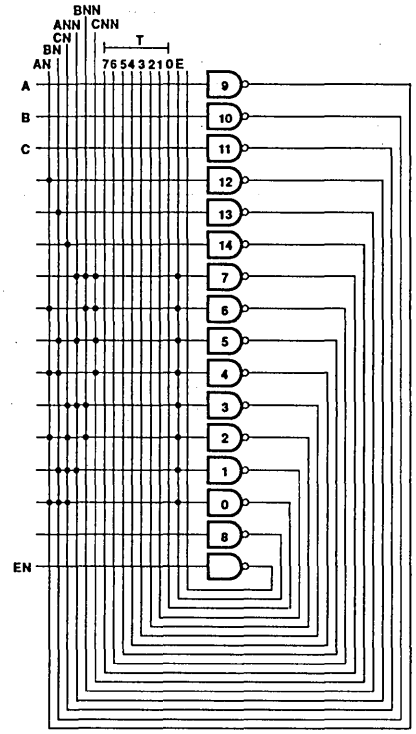


Figure 3.1 Decoder Implementation in NAND Foldback Structure

Designing with Programmable Macro Logic

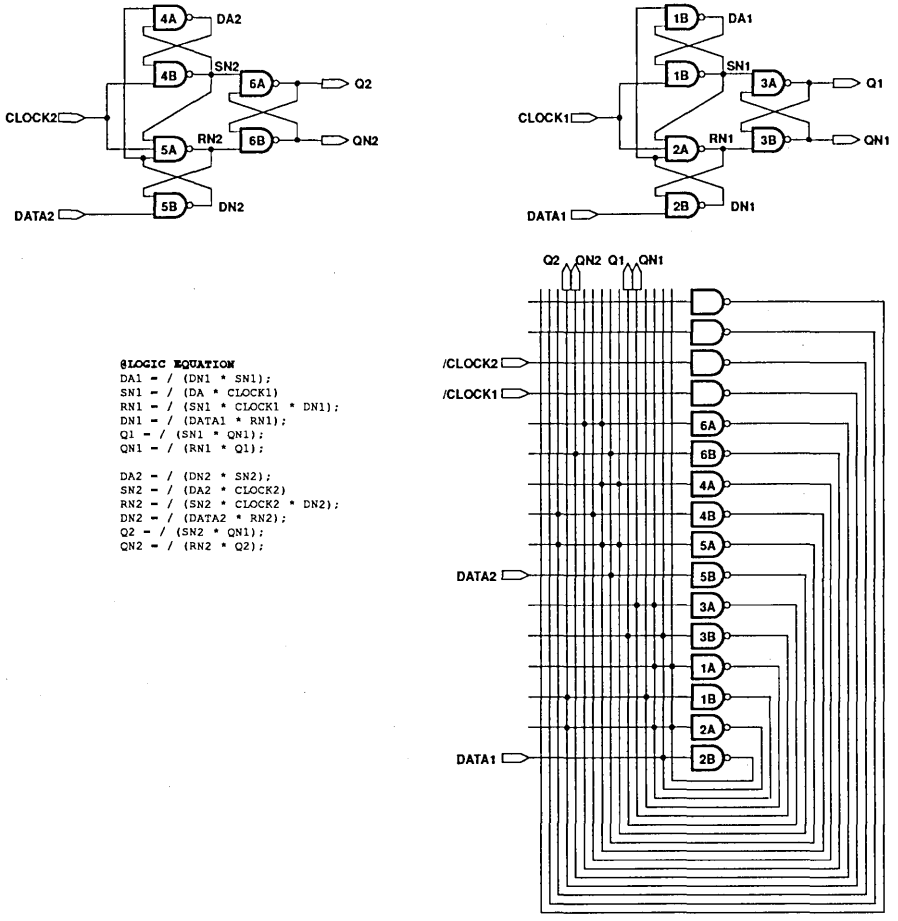


Figure 3.2 Two Flip-Flops Implemented in the NAND Foldback Strucutre

One straightforward example of using a PLHS501 is shown in Figure 3.3. Here, the device is configured to accept the 23 upper address lines generated by a 68000 microprocessor. By selecting the direct and complemented variables, at least 16 distinct address selections can be made using only the dedicated outputs. The designer can combine additional VME bus strobes, or other control signals to qualify the decode or, define 8 additional outputs for expanded selection.

As well, the designer could transform the bidirectionals to inputs and decode over a 32 bit space, selecting combinations off of a 32 bit wide address bus. Because this simple level of design requires only NAND output terms plus 4 NAND gates in the foldback array (for inversion of signals connected to O3.O0), there may be as many as 68 remaining gates to accomplish additional handshaking or logical operations on the input variables.

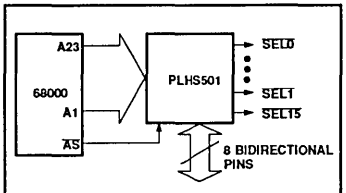


Figure 3.3 68000 Microprocessor Address Decode

Designing with Programmable Macro Logic

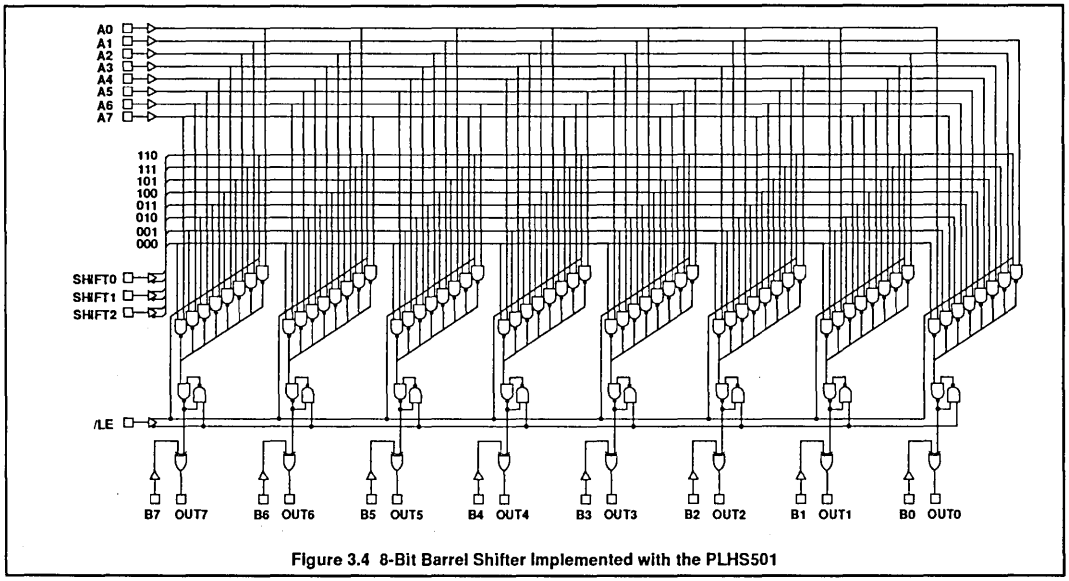


Figure 3.4 8-Bit Barrel Shifter Implemented with the PLHS501

An eight bit barrel shifter exploits most of the PLHS501 as depicted in Figure 3.4. This implementation utilizes all 72 internal foldback NANDs in a relatively brute force configuration as well as 8 output NANDs to generate transparent latched and shifted results. The shift position here is generated by the shift 0, shift 1 and shift 2 inputs which are distinguished and selected from the input cells. Variations on this idea of data manipulation could include direct passing data, mirror imaged data (bit reversal) or byte swapping to name a few.

Part of an eight bit, look-ahead parallel adder is shown in Figure 3.5. Gates necessary to from the level-0 generate and propagate, as well as the XOR output gates generating the resulting sum are not shown. The reader should be aware that this solution exploits four layers of pyramided gates and only utilizes a total of about 58 gates. Additional comparison or Boolean operations could still be generated with remaining NAND functions to achieve additional arithmetic operations. This application should make the reader aware of a new class of applications achievable with third generation PLDs - user definable I/O coprocessors. The approach of increasing microprocessor performance by designing dedicated task coprocessors is now within the grasp of user definable single chip solutions.

Designing with Programmable Macro Logic

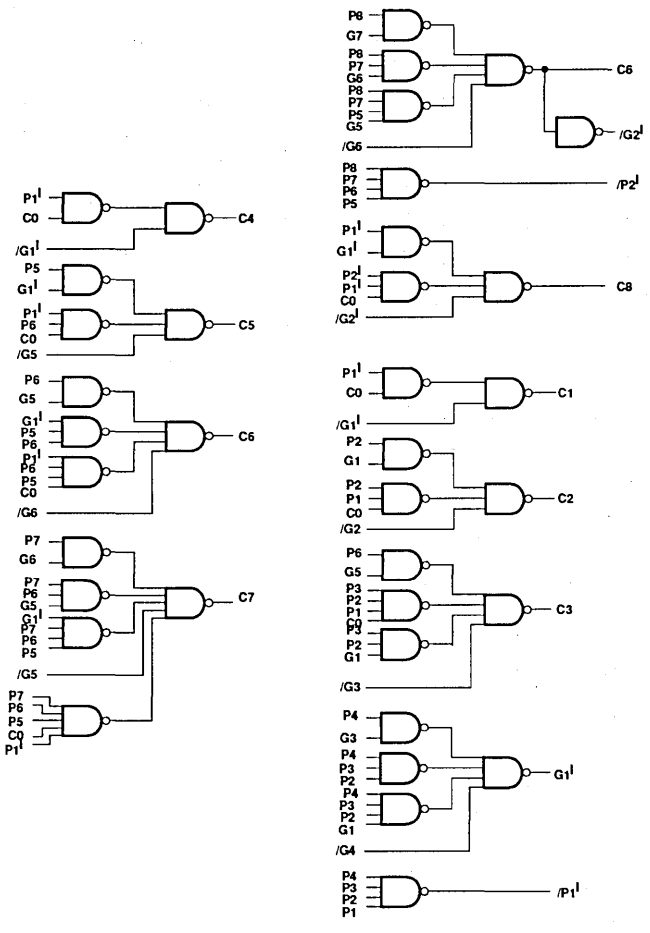


Figure 3.5 Partial NAND Gate Equivalence of the 8-Bit Look-Ahead Adder

An example of one of the least efficient structures realizable on the PLHS501 is shown in Figure 3.6. Here, a cascade of 12 flip-flops are formed into a toggle chain that used all available NAND gates in the main logic array. In the PLHS501 simple cross-coupled latches or transparent D latches are preferred over edge triggered flip-flops simply because they conserve NAND gates. Applications for structures like this include timing generators, rate multiplication, etc. Additional output gates exist on the output terms as shown in Figure 1.2, which could gate the output in multiple state detection configurations. As well, rearranging Figure 3.6 as a 12-bit shifter, picking off states at the output terms could result in a general purpose sequence recognizer capable of recognizing binary string sequences. These strings could be up to 13 bits long (in a Mealy configuration) and 24 distinct sequences could be sensed and detected.

Figure 3.7 shows a 32 to 5-bit priority encoder. This sort of device could generate encoded vector interrupts for 32 contending devices. Of particular interest is the fact that ordinary encoders are not this wide. The designer is, of course, not constrained to generating combinational functions in even powers of two. Thus, the PLHS501 can easily perform customized functions like a 5 to 27 decoder or a 14 to 4 encoder or, even an 18 to 7 multiplexor. For the sake of optimization, the designer is encouraged to implement precisely the function he needs, no more and no less!

The design examples given are illustrative of some typical operations used in ordinary systems. In each case, the example could be thought of as simply an "off the shelf" standard solution to an every day problem (i.e., a de facto standard product).

Designing with Programmable Macro Logic

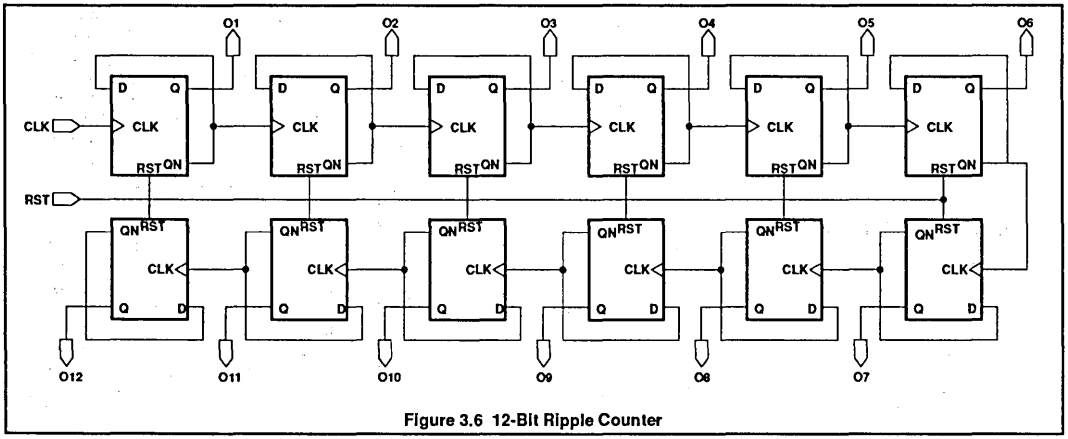


Figure 3.6 12-Bit Ripple Counter

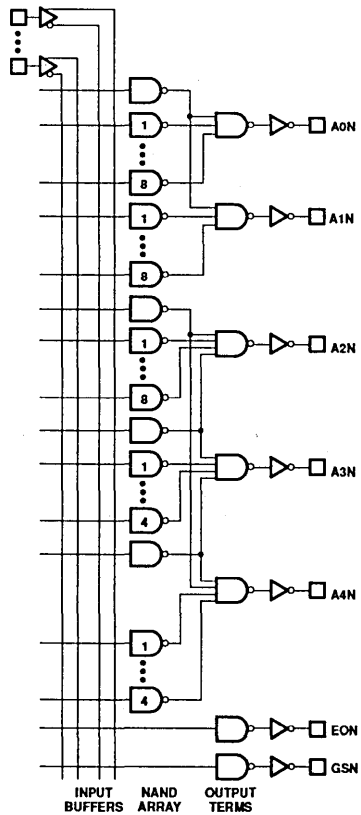


Figure 3.7 Encoder

Designing with Programmable Macro Logic

Programmable Logic Devices

SUCCESSOR ARCHITECTURES

The design examples described and Table 2.2 illustrate the combinational power and the sequential limitations of the PLHS501 – Signetics first PML entry. Clearly the next family members must address the flip-flop issue, and they do. The PLHS502 (Figure 4.1) shows a similar NAND function array of 64 gates with the vital addition of 8 buried D flip-flops and 8 buried S-R flip-flops. Again, 16 pins are devoted to dedicated outputs, 20 straight inputs, 4 clock or general inputs and 8 bidirectionals can be configured to expand input or output capabilities. Slated to operate in the middle 30MHz clocking range, this part greatly expands the sequencer capability beyond the PLHS501. The PLHS502 application range includes state machines like CRC generation/detection, Bus handshakers, LAN handshaking, arithmetic coprocessors, single chip systems and a complete bevy of general sequencer operations such as sequence generation and detection. It should be emphasized that the NAND array is fully connected and circumvents limitations on connectivity as found in other PLD products.

Expanding on the functional capabilities of the PLHS502, the CMOS PML2552 offers 52 flip-flops in a variety of natural configurations with a NAND array near 200 gates. Due to complexity and density, the part will combine a distinctive power-save option and the benefits of scan-design.

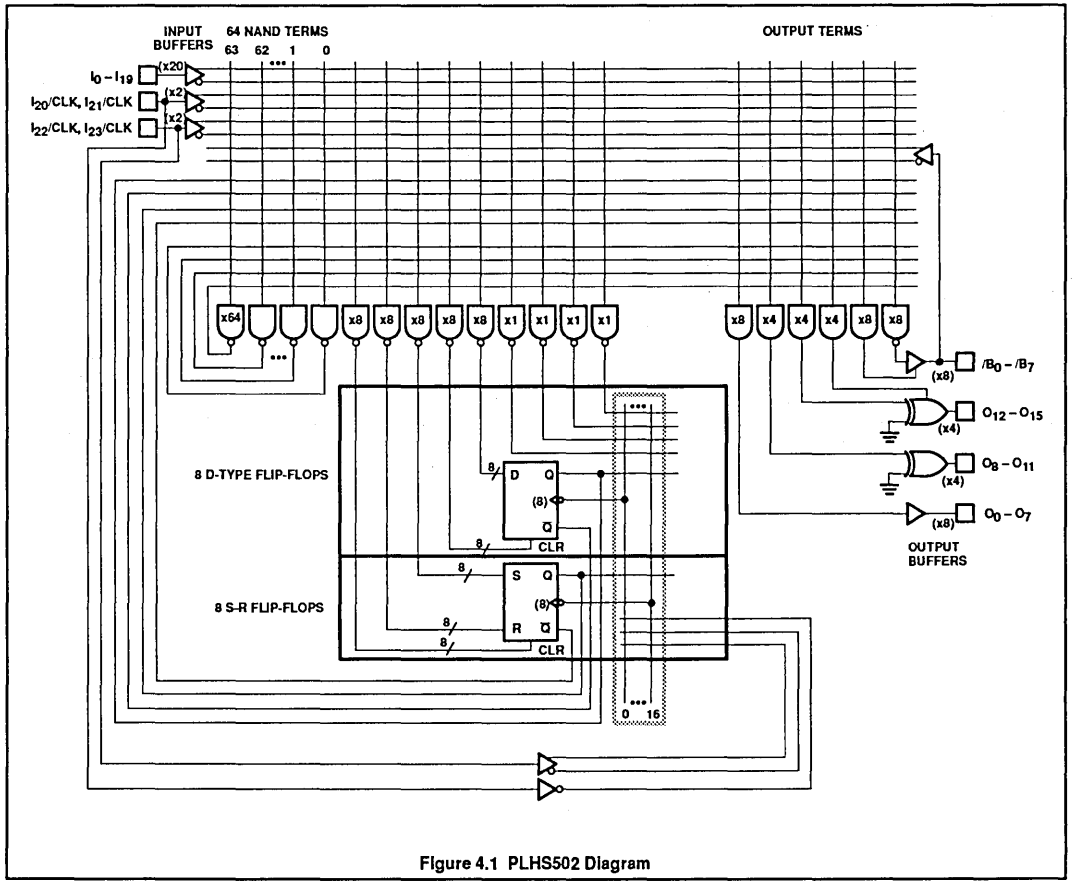


Figure 4.1 PLHS502 Diagram

Designing with Programmable Macro Logic

DEVELOPMENT SUPPORT

AMAZE

The current PML architecture, the PLHS501, is adequately supported by Signetics AMAZE software. Offered free to qualified users, AMAZE can generate the required design files, fusemaps and simulations within the appropriate modules of AMAZE. From a simple menu driven environment on an IBM personal computer (or compatible under MS/DOS), the user can generate a design with logic equations, state equations or schematic entry (using FutureNet DASH or OrCAD SDT software). Once the design is entered, the user must "assemble" it prior to fusing the PML product. If required, the user may simulate the assembled file to determine the accuracy and functional operation of his design. Iteration between design entry, assembly and simulation may be required, depending on the users expectations and the completeness of design. Automatic test vector generation is a simulation option. Currently, the designer may fuse his design using either a DATA I/O Unisite programmer, a Stag ZL–30A or a STREBOR fusing system with corresponding configuration modules.

SNAP

Because the architecture encourages deep functional nesting, a new support tool has been developed. Synthesis, Netlist, Analysis and Program (SNAP) software defines a gate array type development environment. SNAP permits several forms of design capture (schematic, Boolean equations, state equations, etc.), a gate array simulator with back annotation, waveform display and a complete fault analyzer and final fusemap compilation and model extraction. SNAP comes with a library of cells, and designs may be captured independently of the ultimate device that will implement the design. This permits the designer to migrate his design among a family of PML devices just as gate array designs can be moved to larger foundations when they do not route on smaller ones. Figure 2–1 shows the SNAP user interface "Shell" which dictates one sequence of operations to complete a design. Other sequences may be used.

The top portion of the shell depicts the paths available for design entry. Any design may be implemented in any one or a blend of all methods. For instance, a shift register might best be described schematically but a decoder by logic equations. These may be united with a multiplexor described by a text netlist as well. Ultimately, each form of input will be transformed to a function netlist and passed either to the simulation section or to the compiler section. Waveform entry is for simulation stimuli.

The simulator portion of SNAP is a 5-State gate array simulator with full timing information, setup and hold time checking, toggle and fault grade analysis and the ability to display in a wide range of formats, any set of nodes within the design. This permits a designer to zoom in with a synthetic logic state analyzer and view the behavior of any point in the design. Simulations can occur with unit delays, estimations or exact delays. The sequence of operations depicted in Figure 5.1 is entirely arbitrary, as many other paths exist.

It should be noted that the output of the "merger" block represents the composite design, but as yet is not associated to a PML device. This occurs in the compiler portion wherein association to the device occurs and a fusemap is compiled. This is analogous to placement and routing in a gate array environment. Because of the interconnectibility of PML, this is not difficult. Once compiled, the exact assignment of pins, gates and flip-flops is known, so timing parameters may be associated and a new simulation model generated with exact detailed timing embedded. The design may be simulated very accurately at this point, and if correct, a part should be programmed.

To facilitate future migration to workstations, SNAP has been written largely in C. The internal design representation is EDIF (Electronic Design Interchange Format) compatible which permits straightforward porting to many commercially viable environments. SNAP currently utilizes OrCAD for schematic entry with eminent availability of FutureNet™ DASH.

FutureNet and DASH are trademarks of DATA I/O

Designing with Programmable Macro Logic

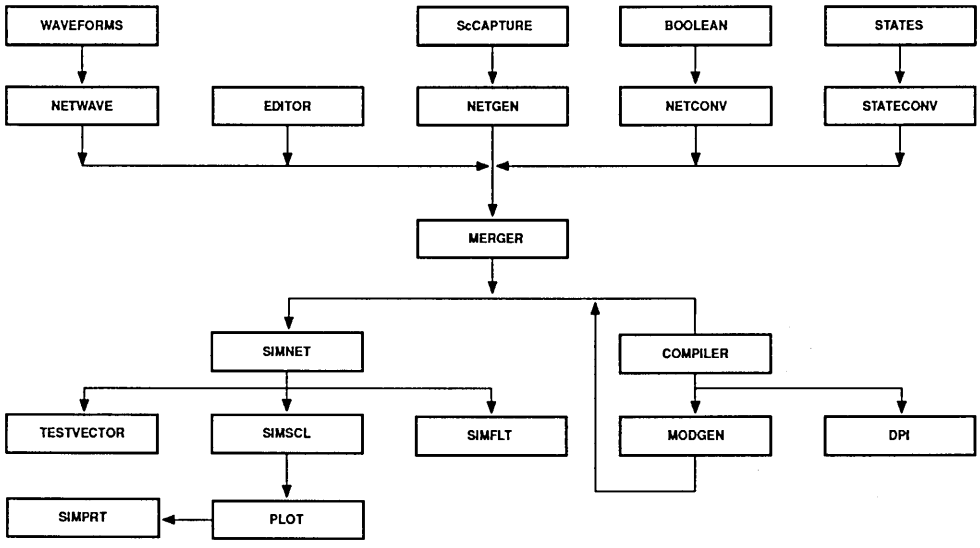


Figure 5.1 SNAP Shell Design

Designing with Programmable Macro Logic

Programmable Logic Devices

PLHS501 EXAMPLES USING AMAZE REVISION 1.6

- Simple gate implementations
- 8-bit barrel shifter
- 12-bit comparator with dual 1 of 8 decoders
- 8-bit carry look-ahead adder
- 32 to 5 priority encoder
- 8-bit shift register with 3-bit counter and sequence detector
- 4-bit synchronous counter

Following are six example applications for the PLHS501 using AMAZE Rev. 1.6. They should not be viewed as showing all possible capabilities of the device. They have been designed to demonstrate some of the PLHS501 features, syntax of AMAZE, and to give the reader some ideas for possible circuit implementations.

Note that these examples were written using AMAZE Rev. 1.6. Although Signetics will try to keep succeeding versions of AMAZE compatible, it may be necessary to change some syntax rules. Therefore, please refer to your AMAZE manual for any notes on differences, if using a revision later than Rev. 1.6.

SIMPLE GATE IMPLEMENTATIONS

In this example six functions were implemented for each of the three major types of output structures. The six functions are AND/OR, AOI, NAND, AND, OR and NOR. A requirement for the AND/OR and AOI gates was to use only two gates each from the foldback array and to combine these product terms in one NAND output gate. To achieve this result, it was necessary for the /B and /O outputs to write equations using internal nodes and brackets around the equation. Refer to Figures 6.1 and 6.2.

For the simulation (Figures 6.3 and 6.4) a binary count of 0 through 15 hex was applied to the input D-A. Each output of the log file was checked against anticipated and other device outputs of the same function for correct operation.

```
################## P I N   L I S T ##################

Left                                                  Right
LABEL   ** FNC **PIN            PIN** FNC **  LABEL
VCC     ** +5V **  8-|        |-46 ** +5V **  VCC
A       ** I   **  9-|        |-45 ** I   **  N/C
B       ** I   ** 10-|        |-44 ** I   **  N/C
C       ** I   ** 11-|   P    |-43 ** I   **  N/C
D       ** I   ** 12-|   L    |-42 ** I   **  N/C
N/C     ** I   ** 13-|   H    |-41 ** I   **  N/C
N/C     ** I   ** 14-|   S    |-40 ** /O  **  F1
F2      ** O   ** 15-|   5    |-39 ** /O  **  F3
F4      ** O   ** 16-|   0    |-38 ** /O  **  F5
F6      ** O   ** 17-|   1    |-37 ** /O  **  N/C
N/C     ** B   ** 18-|        |-36 ** O   **  N/C
F20     ** O   ** 19-|        |-35 ** O   **  F7X
GND     ** 0V  ** 20-|        |-34 ** 0V  **  GND


Bottom                                                Top
LABEL   ** FNC **PIN            PIN** FNC **  LABEL
F4O     ** O   ** 21-|        |- 7 ** I   **  N/C
F6O     ** O   ** 22-|        |- 6 ** I   **  N/C
N/C     ** O   ** 23-|        |- 5 ** I   **  N/C
F1O     ** /O  ** 24-|   P    |- 4 ** I   **  N/C
F3O     ** /O  ** 25-|   L    |- 3 ** I   **  N/C
F5O     ** /O  ** 26-|   H    |- 2 ** I   **  N/C
N/C     ** /O  ** 27-|   S    |- 1 ** I   **  N/C
F1X     ** O   ** 28-|   5    |-52 ** I   **  N/C
F2X     ** O   ** 29-|   0    |-51 ** I   **  N/C
F3X     ** O   ** 30-|   1    |-50 ** I   **  N/C
F4X     ** O   ** 31-|        |-49 ** I   **  N/C
F5X     ** O   ** 32-|        |-48 ** I   **  N/C
F6X     ** O   ** 33-|        |-47 ** I   **  N/C
```

Figure 6.1 Gates Pin List

Designing with Programmable Macro Logic

```
File Name : GATES
Date : 9/15/1987
Time : 9:30:35

@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION

Various single and two level gate implementations
using different output structures

@INTERNAL NODE
@COMMON PRODUCT TERM
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION

"AND-OR using /B output"
f1 = /(/[a*b] * /[c*d]) ;
"AND-OR-INVERT using B output"
f2 = /[(a*b) + (c*d)];
"NAND using /B output"
f3 = /(a*b*c*d);
"AND using B output"
f4 = a*b*c*d;
"OR using /B output"
f5 = /(/a*/b*/c*/d);
"NOR using B output"
f6 = /a*/b*/c*/d;

"AND-OR using XOR output:
f1x = (a*b)+(c*d);
"AND-OR-INVERT using XOR output"
f2x = /[(a*b) + (c*d)];
"NAND using XOR output"
f3x = /(a*b*c*d);
"AND using XOR output"
f4x = a*b*c*d;
"OR using XOR output"
f5x = /(/a*/b*/c*d);
"NOR using XOR output"
f6x = /a*/b*/c*d;
"XOR using XOR output"
f7x : xr1 = a;
      xr2 = b;

"AND-OR using /O output"
f1o = /(/[a*b] * /[c*d]) ;
"AND-OR-INVERT using O output"
f2o = /[(a*b) + (c*d)];
"NAND using /O output"
f3o = /(a*b*c*d);
"AND using O output"
f4o = a*b*c*d;
"OR using /O output"
f5o = /(/a*/b*/c*/d);
"NOR using O output"
f6o = /a*/b*/c*/d;
```

Figure 6.2 Gates Boolean Equations

```
" Simulation input file for GATES
"
" <---------INPUTS---------> < B,/B >
" 222211111111111
" 32198976543210987654321O 76543210
"
"   DCBA  A is MSB, D is LSB
"
  HHLLLLHHHHHHHHHHHHHHHH -------- "Input all 0's A-D"
  HHHLLLHHHHHHHHHHHHHHHH -------- "Input count 1"
  HHLHLLHHHHHHHHHHHHHHHH -------- "Through...
  HHHHLLHHHHHHHHHHHHHHHH --------
  HHLLHLHHHHHHHHHHHHHHHH --------
  HHHLHLHHHHHHHHHHHHHHHH --------
  HHLHHLHHHHHHHHHHHHHHHH --------
  HHHHHLHHHHHHHHHHHHHHHH --------
  HHLLLLHHHHHHHHHHHHHHHH --------
  HHHLLLHHHHHHHHHHHHHHHH --------
  HHLHLLHHHHHHHHHHHHHHHH --------
  HHHHLLHHHHHHHHHHHHHHHH --------
  HHLLLLHHHHHHHHHHHHHHHH --------
  HHHLLLHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHH -------- "Count 15 hex"
QUIT
```

Figure 6.3 Gates Simulation Input File

Designing with Programmable Macro Logic

```
PLHS501           GATES           Time = 13:59:42 Date = 9/14/1987
"
" <--------INPUTS--------> < B,/B > < XOR  > < /O,O > TRACE TERMS
" 222211111111111
" 32109876543210987654321O 76543210 76543210 76543210
"
  11000011111111111111111111 HHLHLHLL LLHLLHHL LLHLHHLH ;
  11100011111111111111111111 HLLHLHHL LLLHLHHL LHHLHLLH ;
  11010011111111111111111111 HLLHLHHL LLLHLHHL LHHLHLLH ;
  11110011111111111111111111 HLLLHHHL LLLHLHLH LHHHHLLL ;
  11001011111111111111111111 HLLHLHHL LHLHLHHL LHHLHLLH ;
  11101011111111111111111111 HLLHLHHL LHLHLHHL LHHLHLLH ;
  11011011111111111111111111 HLLHLHHL LHLHLHHL LHHLHLLH ;
  11111011111111111111111111 HLLLHHHL LHLHLHLH LHHHHLLL ;
  11000111111111111111111111 HLLHLHHL LHLHLHHL LHHLHLLH ;
  11100111111111111111111111 HLLHLHHL LHLHLHHL LHHLHLLH ;
  11010111111111111111111111 HLLHLHHL LHLHLHHL LHHLHLLH ;
  11110111111111111111111111 HLLLHHHL LHLHLHLH LHHHHLLL ;
  11001111111111111111111111 HLLLHHHL LLLHLHLH LHHHHLLL ;
  11101111111111111111111111 HLLLHHHL LLLHLHLH LHHHHLLL ;
  11011111111111111111111111 HLLLHHHL LLLHLHLH LHHHHLLL ;
  11111111111111111111111111 HLHLHLHL LLLHHLLH LHLHHLHL ;
"
" ------------------------ --------   I/O CONTROL LINES
"                          OOOOOOOO   DESIGNATED I/O USAGE
"                          OOOOOOOO   ACTUAL I/O USAGE
"
" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.4 Gates Simulation Log File

Designing with Programmable Macro Logic

8-BIT BARREL SHIFTER

This 8-bit shifter will shift to the right, data applied to A7 – A9 with the result appearing on OUT7 – OUT0. Data may be shifted by 1 to 7 places by indicating the desired binary count on pins SHIFT2 – SHIFT0. Data applied to the OUT0 position for a shift of 1. For a shift of 0, A7 will appear on OUT7.

Also included is a transparent latch for the output bits. The input 'COMPLMTO' will invert all output bits simultaneously and input /OE will 3-State all outputs.

```
File Name : BRLSHFT
Date : 9/15/1987
Time : 9:31:58

################## P I N   L I S T ##################

Left                                                Right
LABEL   ** FNC **PIN          PIN** FNC **  LABEL
VCC     ** +5V **  8-|       |-46 ** +5V **  VCC
A2      ** I   **  9-|       |-45 ** I   **  N/C
A3      ** I   ** 10-|       |-44 ** I   **  N/C
A4      ** I   ** 11-|   P   |-43 ** I   **  N/C
A5      ** I   ** 12-|   L   |-42 ** I   **  N/C
A6      ** I   ** 13-|   H   |-41 ** I   **  N/C
A7      ** I   ** 14-|   S   |-40 ** /O  **  L0
L4      ** O   ** 15-|   5   |-39 ** /O  **  L1
L5      ** O   ** 16-|   0   |-38 ** /O  **  L2
L6      ** O   ** 17-|   1   |-37 ** /O  **  L3
L7      ** O   ** 18-|       |-36 ** O   **  OUT7
N/C     ** O   ** 19-|       |-35 ** O   **  OUT6
GND     ** 0V  ** 20-|       |-34 ** 0V  **  GND


Bottom                                              Top
LABEL   ** FNC **PIN          PIN** FNC **  LABEL
N/C     ** O   ** 21-|       |- 7 ** I   **  A1
N/C     ** O   ** 22-|       |- 6 ** I   **  A0
N/C     ** O   ** 23-|       |- 5 ** I   **  SHIFT2
N/C     ** /O  ** 24-|   P   |- 4 ** I   **  SHIFT1
N/C     ** /O  ** 25-|   L   |- 3 ** I   **  SHIFT0
N/C     ** /O  ** 26-|   H   |- 2 ** I   **  COMPLMTO
N/C     ** /O  ** 27-|   S   |- 1 ** I   **  /LE
OUT0    ** O   ** 28-|   5   |-52 ** I   **  /OE
OUT1    ** O   ** 29-|   0   |-51 ** I   **  N/C
OUT2    ** O   ** 30-|   1   |-50 ** I   **  N/C
OUT3    ** O   ** 31-|       |-49 ** I   **  N/C
OUT4    ** O   ** 32-|       |-48 ** I   **  N/C
OUT5    ** O   ** 33-|       |-47 ** I   **  N/C
```

Figure 6.5 Barrel Shifter Pin List

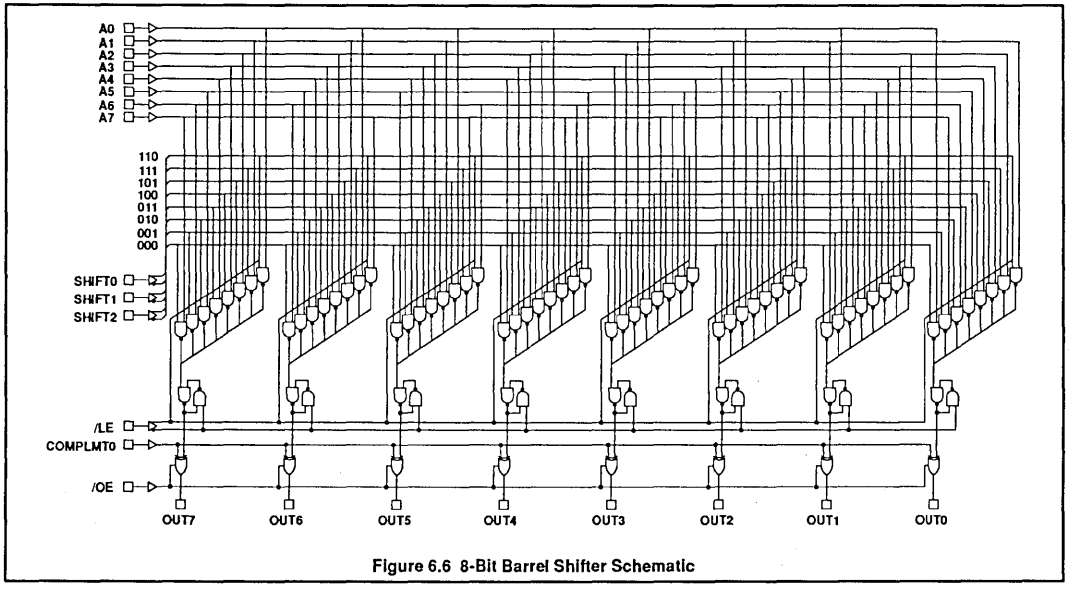


Figure 6.6 8-Bit Barrel Shifter Schematic

Designing with Programmable Macro Logic

```
File Name : BRLSHFT
Date : 9/15/1987
Time : 9:32:14

@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESTRIPTION

8 Bit Barrel Shifter
with 3-state latched outputs

@INTERNAL NODE
nod1,nod2,nod3,nod4,nod5,nod6,nod7,nod8;
nod9,nod10,nod11,nod12,nod13,nod14,nod15,nod16;
nod17,nod18,nod19,nod20,nod21,nod22,nod23,nod24;
nod25,nod26,nod27,nod28,nod29,nod30,nod31,nod32;
i3,i2,i1,i0;
@COMMON PRODUCT TERM
rot0 = /shift2 * /shift1 * /shift0;
rot1 = /shift2 * /shift1 *  shift0;
rot2 = /shift2 *  shift1 * /shift0;
rot3 = /shift2 *  shift1 *  shift0;
rot4 =  shift2 * /shift1 * /shift0;
rot5 =  shift2 * /shift1 *  shift0;
rot6 =  shift2 *  shift1 * /shift0;
rot7 =  shift2 *  shift1 *  shift0;
@I/O DIRECTION
xe0 = oe;
xe1 = oe;
xe2 = oe;
xe3 = oe;
@I/O STEERING
@LOGIC EQUATION

  17 = /[a7 * rot0 * /le +
         a6 * rot1 * /le +
         a5 * rot2 * /le +
         a4 * rot3 * /le +
         a3 * rot4 * /le +
         a2 * rot5 * /le +
         a1 * rot6 * /le +
         a0 * rot7 * /le +
         le * /17];

  16 = /[a6 * rot0 * /le +
         a5 * rot1 * /le +
         a4 * rot2 * /le +
         a3 * rot3 * /le +
         a2 * rot4 * /le +
         a1 * rot5 * /le +
         a0 * rot6 * /le +
         a7 * rot7 * /le +
         le * /16];

  15 = /[a5 * rot0 * /le +
         a4 * rot1 * /le +
         a3 * rot2 * /le +
         a2 * rot3 * /le +
         a1 * rot4 * /le +
         a0 * rot5 * /le +
         a7 * rot6 * /le +
         a6 * rot7 * /le +
         le * /15];

  14 = /[a4 * rot0 * /le +
         a3 * rot1 * /le +
         a2 * rot2 * /le +
         a1 * rot3 * /le +
```

Figure 6.7 Barrel Shifter Boolean Equations

Designing with Programmable Macro Logic

```
            a0 * rot4 * /le +
            a7 * rot5 * /le +
            a6 * rot6 * /le +
            a5 * rot7 * /le +
            le * /l4];

  nod1 = [a3 * rot0 * /le];
  nod2 = [a2 * rot1 * /le];
  nod3 = [a1 * rot2 * /le];
  nod4 = [a0 * rot3 * /le];
  nod5 = [a7 * rot4 * /le];
  nod6 = [a6 * rot5 * /le];
  nod7 = [a5 * rot6 * /le];
  nod8 = [a4 * rot7 * /le];
  i3 = [le *l3];

l3 = /([/nod1*/nod2*/nod3*/nod4*/nod5*/nod6*/nod7*/nod8*/i3]);

  nod9 = [a2 * rot0 * /le];
  nod10= [a1 * rot1 * /le];
  nod11= [a0 * rot2 * /le];
  nod12= [a7 * rot3 * /le];
  nod13= [a6 * rot4 * /le];
  nod14= [a5 * rot5 * /le];
  nod15= [a4 * rot6 * /le];
  nod16= [a3 * rot7 * /le];
  i2 = [le * l2];

l2 = /([/nod9*/nod10*/nod11*/nod12*/nod13*/nod14*/nod15*/nod16*/i2]);

  nod17= [a1 * rot0 * /le];
  nod18= [a0 * rot1 * /le];
  nod19= [a7 * rot2 * /le];
  nod20= [a6 * rot3 * /le];
  nod21= [a5 * rot4 * /le];
  nod22= [a4 * rot5 * /le];
  nod23= [a3 * rot6 * /le];
  nod24= [a2 * rot7 * /le];
  i1 = [le * l1];

l1 = /([/nod17*/nod18*/nod19*/nod20*/nod21*/nod22*/nod23*/nod24*/i1]);

  nod25= [a0 * rot0 * /le];
  nod26= [a7 * rot1 * /le];
  nod27= [a6 * rot2 * /le];
  nod28= [a5 * rot3 * /le];
  nod29= [a4 * rot4 * /le];
  nod30= [a3 * rot5 * /le];
  nod31= [a2 * rot6 * /le];
  nod32= [a1 * rot7 * /le];
  i0 = [le * l0];

l0 = /([/nod25*/nod26*/nod27*/nod28*/nod29*/nod30*/nod31*/nod32*/i0]);

  out7 : xr1 = /l7;
         xr2 = complmto;
  out6 : xr1 = /l6;
         xr2 = complmto;
  out5 : xr1 = /l5;
         xr2 = complmto;
  out4 : xr1 = /l4;
         xr2 = complmto;
  out3 : xr1 = /l3;
         xr2 = complmto;
  out2 : xr1 = /l2;
         xr2 = complmto;
  out1 : xr1 = /l1;
         xr2 = complmto;
  out0 : xr1 = /l0;
         xr2 = complmto;
```

Figure 6.7 Barrel Shifter Boolean Equations (Continued)

Designing with Programmable Macro Logic

```
"Simulation input for BRLSHFT
" <--------INPUTS--------> < B,/B >
  lllllllllllhlhhhhhhhhhh -------- "Shift0 - Input 00"
  hhhhhhhhllllhlhhhhhhhhhh -------- "               FF"
  hlhlhlhllllhlhhhhhhhhhh -------- "               AA"
  lhlhlhlhllllhlhhhhhhhhhh -------- "               55"
  lllllllhllllhlhhhhhhhhhh -------- "Shift0 - Input 01"
  lllllllhllhlhlhhhhhhhhhh -------- "     1           "
  lllllllhlhllhlhhhhhhhhhh -------- "     2           "
  lllllllhlhhlhlhhhhhhhhhh -------- "     3           "
  lllllllhhlllhlhhhhhhhhhh -------- "     4           "
  lllllllhhlhlhlhhhhhhhhhh -------- "     5           "
  lllllllhhhllhlhhhhhhhhhh -------- "     6           "
  lllllllhhhhlhlhhhhhhhhhh -------- "     7           "
  llllhhhhhlllhlhhhhhhhhhh -------- "Shift4 - Input 0F"
  hlhlhlhlllhlhlhhhhhhhhhh -------- "Shift1 - Input AA"
  lllllllllllhlhhhhhhhhhh -------- "Shift0 - Input 00"
  lllllllllllhhlhhhhhhhhhh -------- "Complement output"
  llllhhhhllllhlhhhhhhhhhh -------- "Shift0 - Input 0F"
  hlhlhlhlllllllhhhhhhhhhh -------- "start latch test"
  lllllllhlllllllhhhhhhhhhh -------- "continue...
  llllllhllllllhhhhhhhhhh --------
  lllllhlllllllhhhhhhhhhh --------
  llllhlllllllhhhhhhhhhh --------
  lllhlllllllllhhhhhhhhhh --------
  llhlllllllllhhhhhhhhhh --------
  lhlllllllllllhhhhhhhhhh --------
  hlllllllllllhhhhhhhhhh -------- "end"
  lllllllllllhlhhhhhhhhhh -------- "un-latch output"
  lllllllllllhhhhhhhhhhh -------- "3-state outputs"
QUIT
```

Figure 6.8 Barrel Shifter Simulation Input File

Designing with Programmable Macro Logic

```
PLHS501            BRLSHFT          Time = 14:12:36 Date = 9/14/1987
"
" <--------INPUTS--------> < B,/B > < XOR  > < /O,O >  TRACE TERMS
  222211111111111
" 32109876543210987654321O 76543210 76543210 76543210
"
  000000000000101111111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  111111110000101111111111 LLLLHHHH HHHHHHHH LLLLHHHH ;
  101010100000101111111111 LHLHLHLH HLHLHLHL LLLLHHHH ;
  010101010000101111111111 HLHLHLHL LHLHLHLH LLLLHHHH ;
  000000010000101111111111 HHHHLLLL LLLLLLLH LLLLHHHH ;
  000000010010101111111111 HHHHLHLL LLLLLLHL LLLLHHHH ;
  000000010100101111111111 HHHHLHHL LLLLLHLL LLLLHHHH ;
  000000010110101111111111 HHHHLLLH LLLLHLLL LLLLHHHH ;
  000000011000101111111111 HHHLLLLL LLLHLLLL LLLLHHHH ;
  000000011010101111111111 HHLHLLLL LLHLLLLL LLLLHHHH ;
  000000011100101111111111 HLHHLLLL LHLLLLLL LLLLHHHH ;
  000000011110101111111111 LHHHLLLL HLLLLLLL LLLLHHHH ;
  000011111000101111111111 LLLLLLLL HHHHLLLL LLLLHHHH ;
  101010100010101111111111 HLHLHLHL LHLHLHLH LLLLHHHH ;
  000000000000101111111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  000000000001101111111111 HHHHLLLL HHHHHHHH LLLLHHHH ;
  000011110000101111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  101010100000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  000000010000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  000000100000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  000001000000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  000010000000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  000100000000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  001000000000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  010000000000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  100000000000001111111111 HHHHHHHH LLLLHHHH LLLLHHHH ;
  000000000000101111111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  000000000000111111111111 HHHHLLLL ........ LLLLHHHH ;
"
" ------------------------ --------   I/O CONTROL LINES
"                          OOOOOOOO   DESIGNATED I/O USAGE
"                          OOOOOOOO   ACTUAL I/O USAGE
"
" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.9 Barrel Shifter Simulation Log File

Designing with Programmable Macro Logic

```
File Name : 12BITCMP
Date : 9/15/1987
Time : 9:36:0

################ P I N   L I S T ################

Left                                          Right
LABEL   ** FNC **PIN          PIN** FNC **    LABEL
VCC     ** +5V **  8-|        |-46 ** +5V **  VCC
B6      ** I   **  9-|        |-45 ** I   **  A4
B7      ** I   ** 10-|        |-44 ** I   **  A3
B8      ** I   ** 11-|   P    |-43 ** I   **  A2
B9      ** I   ** 12-|   L    |-42 ** I   **  A1
B10     ** I   ** 13-|   H    |-41 ** I   **  A0
B11     ** I   ** 14-|   S    |-40 ** /O  **  CMPOUT
ENCOMP  ** I   ** 15-|   5    |-39 ** I   **  DA2
DCDREN  ** I   ** 16-|   0    |-38 ** I   **  DA1
RW      ** I   ** 17-|   1    |-37 ** I   **  DA0
N/C     ** B   ** 18-|        |-36 ** O   **  R7
W0      ** O   ** 19-|        |-35 ** O   **  R6
GND     ** 0V  ** 20-|        |-34 ** 0V  **  GND


Bottom                                        Top
LABEL   ** FNC **PIN          PIN** FNC **    LABEL
W1      ** O   ** 21-|        |- 7 ** I   **  B5
W2      ** O   ** 22-|        |- 6 ** I   **  B4
W3      ** O   ** 23-|        |- 5 ** I   **  B3
W4      ** /O  ** 24-|   P    |- 4 ** I   **  B2
W5      ** /O  ** 25-|   L    |- 3 ** I   **  B1
W6      ** /O  ** 26-|   H    |- 2 ** I   **  B0
W7      ** /O  ** 27-|   S    |- 1 ** I   **  A11
R0      ** O   ** 28-|   5    |-52 ** I   **  A10
R1      ** O   ** 29-|   0    |-51 ** I   **  A9
R2      ** O   ** 30-|   1    |-50 ** I   **  A8
R3      ** O   ** 31-|        |-49 ** I   **  A7
R4      ** O   ** 32-|        |-48 ** I   **  A6
R5      ** O   ** 33-|        |-47 ** I   **  A5
```

Figure 6.10 12-Bit Comparator Pin List

12-BIT COMPARATOR WITH DUAL 1-OF-8 DECODERS

Two functions that are very often associated with controlling I/O parts are address comparison and address decoding. In this example, both functions are programmed into a PLHS501 using 52 out of the 72 foldback NAND terms.

The comparator compares 12 bits on inputs A11 – A0 to inputs B11 – B0 when the input 'ENCMP' is High. Output 'CMPOUT' will become Active-Low when all 12 bits of the A input match the B. Selection between the two decoders is done with input 'R/W'. Only one output may be active (Low) at a time. Although currently separate functions, the decoder enable may be derived internally from 'CMPOUT' freeing 2 bidirectional pins which together with available foldback NAND terms, may be used to incorporate a third function.

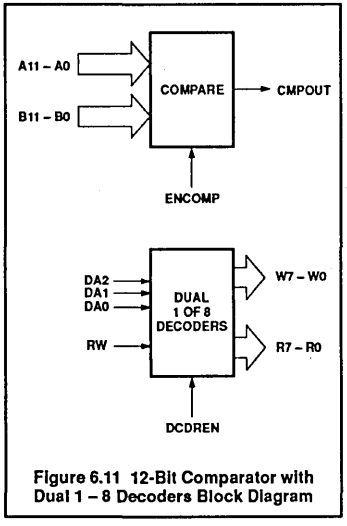


Figure 6.11 12-Bit Comparator with Dual 1 – 8 Decoders Block Diagram

Designing with Programmable Macro Logic

```
File Name : 12BITCMP
Date : 9/15/1987
Time : 9:36:17

@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION

12-bit address comparator and dual 1 of 8 decoders

@INTERNAL NODE
axb0,axb1,axb2,axb3,axb4,axb5,axb6;
axb7,axb8,axb9,axb10,axb11;
@COMMON PRODUCT TERM
ad0=/da2*/da1*/da0*dcdren;
ad1=/da2*/da1* da0*dcdren;
ad2=/da2* da1*/da0*dcdren;
ad3=/da2* da1* da0*dcdren;
ad4= da2*/da1*/da0*dcdren;
ad5= da2*/da1* da0*dcdren;
ad6= da2* da1*/da0*dcdren;
ad7= da2* da1* da0*dcdren;
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION

"12-Bit Address Comparator"

axb0 = a0*/b0 + /a0*b0;
axb1 = a1*/b1 + /a1*b1;
axb2 = a2*/b2 + /a2*b2;
axb3 = a3*/b3 + /a3*b3;
axb4 = a4*/b4 + /a4*b4;
axb5 = a5*/b5 + /a5*b5;
axb6 = a6*/b6 + /a6*b6;
axb7 = a7*/b7 + /a7*b7;
axb8 = a8*/b8 + /a8*b8;
axb9 = a9*/b9 + /a9*b9;
axb10 = a10*/b10 + /a10*b10;
axb11 = a11*/b11 + /a11*b11;

cmpout = /(/axb0*/axb1*/axb2*/axb3*/axb4*/axb5*/axb6*/axb7*/axb8*/axb9*
                  /axb10*/axb11*encomp);

"Dual 1 of 8 decoders

- da2-da0 are address inputs
- dcdren is an enable input
- rw selects which group of 8 outputs r7-r0 or w7-w0
     will have the decided active low output"

w7 = /(ad7*/rw);
w6 = /(ad6*/rw);
w5 = /(ad5*/rw);
w4 = /(ad4*/rw);
w3 = /(ad3*/rw);
w2 = /(ad2*/rw);
w1 = /(ad1*/rw);
w0 = /(ad0*/rw);

r7 = /(ad7* rw);
r6 = /(ad6* rw);
r5 = /(ad5* rw);
r4 = /(ad4* rw);
r3 = /(ad3* rw);
r2 = /(ad2* rw);
r1 = /(ad1* rw);
r0 = /(ad0* rw);
```

Figure 6.12 12-Bit Comparator Boolean Equations

Designing with Programmable Macro Logic

```
" Simulation inputs for 12BITCMP
" <--------INPUTS--------> < B,/B >
" 2222HHHHHHHHHH
" 32HL98765432HL98765432HL 765432HL
"
  LLLLLLLLLLLLLLLLLLLLLL -LLL-LLL "disable comp"
  LLLLLLLLLLLLLLLLLLLLLL -LLH-LLL "enable comp"
  HLHLHLHLHLHLHLHLHLHLHL -LLH-LLL "comp AA"
  LHLHLHLHLHLHLHLHLHLHLH -LLH-LLL "comp 55"
  HHHHHHHHHHHHHHHHHHHHHH -LLH-LLL "comp FF"
  HHHHHHHHHHHHLHHHHHHHHH -LLH-LLL "A not equal B"
  LLLLLLLLLLLLLLLLLLLLLL -LHL-LLL "enable decoder W"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-LLL "enable decoder R"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-LLH "count 1"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-LHL "      2"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-LHH "      3"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-HLL "      4"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-HLH "      5"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-HHL "      6"
  LLLLLLLLLLLLLLLLLLLLLL -HHL-HHH "      7"
  LLLLLLLLLLLLLLLLLLLLLL -LHL-HHH "enable decoder W"
QUIT
```

Figure 6.13 12-Bit Comparator Simulation Input File

```
PLHS501          12BITCMP        Time = 15:0:58 Date = 9/14/1987
"
" <--------INPUTS--------> < B,/B > < XOR  > < /O,O >  TRACE TERMS
" 222211111111111
" 321098765432109876543210 76543210 76543210 76543210
"
  000000000000000000000000 H000H000 HHHHHHHH HHHHHHHH ;
  000000000000000000000000 H001L000 HHHHHHHH HHHHHHHH ;
  101010101010101010101010 H001L000 HHHHHHHH HHHHHHHH ;
  010101010101010101010101 H001L000 HHHHHHHH HHHHHHHH ;
  111111111111111111111111 H001L000 HHHHHHHH HHHHHHHH ;
  111111111111011111111111 H001H000 HHHHHHHH HHHHHHHH ;
  000000000000000000000000 H010H000 HHHHHHHH HHHHHHHL ;
  000000000000000000000000 H110H000 HHHHHHHL HHHHHHHH ;
  000000000000000000000000 H110H001 HHHHHHLH HHHHHHHH ;
  000000000000000000000000 H110H010 HHHHHLHH HHHHHHHH ;
  000000000000000000000000 H110H011 HHHHLHHH HHHHHHHH ;
  000000000000000000000000 H110H100 HHHLHHHH HHHHHHHH ;
  000000000000000000000000 H110H101 HHLHHHHH HHHHHHHH ;
  000000000000000000000000 H110H110 HLHHHHHH HHHHHHHH ;
  000000000000000000000000 H110H111 LHHHHHHH HHHHHHHH ;
  000000000000000000000000 H010H111 HHHHHHHH LHHHHHHH ;
"
" ------------------------ --------   I/O CONTROL LINES
"                          OIIIOIII   DESIGNATED I/O USAGE
"                          OIIIOIII   ACTUAL I/O USAGE
"
" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.14 12-Bit Comparator Simulation Log File

Designing with Programmable Macro Logic

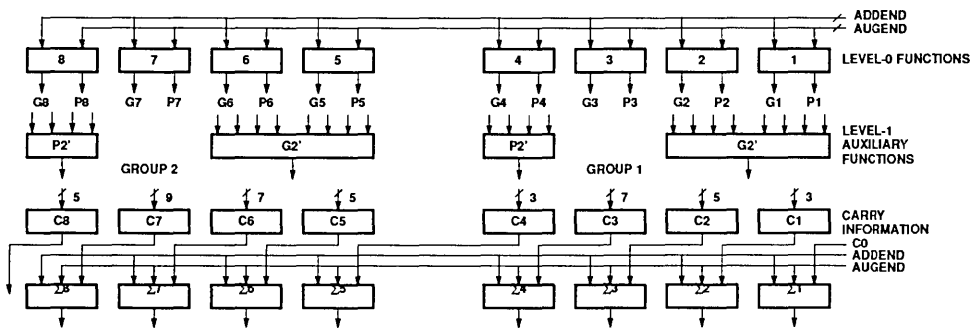
8-BIT CARRY LOOK-AHEAD ADDER

This function may be used as part of an ALU design or simply to off-load a microprocessor. Figure 6.16 is a block diagram showing the individual components needed for each bit.

A carry input (C0) is provided along with a carry output (C8). The result of an addition between the inputs A7 – A0 and B7 – B0 occurs on outputs SUM7 – SUM0.

```
File Name : 8BITADDR
Date : 9/15/1987
Time : 9:37:21

################## P I N   L I S T ##################

Left                                                Right
LABEL   ** FNC **PIN            PIN** FNC **        LABEL
VCC     ** +5V **  8-|        |-46 ** +5V **        VCC
A2      ** I   **  9-|        |-45 ** I   **        N/C
A3      ** I   ** 10-|        |-44 ** I   **        N/C
A4      ** I   ** 11-|   P    |-43 ** I   **        N/C
A5      ** I   ** 12-|   L    |-42 ** I   **        N/C
A6      ** I   ** 13-|   H    |-41 ** I   **        N/C
A7      ** I   ** 14-|   S    |-40 ** /O  **        N/C
N/C     ** B   ** 15-|   5    |-39 ** I   **        N/C
N/C     ** B   ** 16-|   0    |-38 ** I   **        N/C
N/C     ** B   ** 17-|   1    |-37 ** I   **        N/C
N/C     ** B   ** 18-|        |-36 ** O   **        SUM7
C8      ** O   ** 19-|        |-35 ** O   **        SUM6
GND     ** 0V  ** 20-|        |-34 ** 0V  **        GND


Bottom                                              Top
LABEL   ** FNC **PIN            PIN** FNC **        LABEL
N/C     ** O   ** 21-|        |- 7 ** I   **        A1
N/C     ** O   ** 22-|        |- 6 ** I   **        A0
N/C     ** O   ** 23-|        |- 5 ** I   **        B7
N/C     ** /O  ** 24-|   P    |- 4 ** I   **        B6
N/C     ** /0  ** 25-|   L    |- 3 ** I   **        B5
N/C     ** /O  ** 26-|   H    |- 2 ** I   **        B4
N/C     ** /O  ** 27-|   S    |- 1 ** I   **        B3
SUM0    ** O   ** 28-|   5    |-52 ** I   **        B2
SUM1    ** O   ** 29-|   0    |-51 ** I   **        B1
SUM2    ** O   ** 30-|   1    |-50 ** I   **        B0
SUM3    ** O   ** 31-|        |-49 ** I   **        C0
SUM4    ** O   ** 32-|        |-48 ** I   **        N/C
SUM5    ** O   ** 33-|        |-47 ** I   **        N/C
```

Figure 6.15 8-Bit Adder Pin List



NOTES:

G1' = G4 + P4 * G3 + P4 * P3 * G2 + P4 * P3 * P2 * G1;
P1' = P4 * P3 * P2 * P1
G2' = G8 + P8 * G7 + P8 * P7 * G6 + P8 * P7 * P6 * G5;
P2' = P8 * P7 * P6 * P5

C1 = G1 + P1 * C0;
C2 = G2 + P2 * G1 + P2 * P1 * C0;
C3 = G3 + P3 * G2 + P3 * P2 * G1 + P3 * P2 * P1 * C0;
C4 = G1' + P1' * C0;
C5 = G5 + P5 * G1' +P5 * P1' * C0;
C6 = G6 + P6 * G5 + P6 * P5 * G1' + P6 * P5 * P1' * C0;
C7 = G7 + P7 * G6 + P7 * P6 * G5 + P7 * P6 * P5 * G1' +P7 * P6 * P5 * P1' * C0;
C8 = G2' + P2' * G1' + P2' * P1' * C0;

Figure 6.16 8-Bit Carry Look-Ahead Adder Block Diagram and Equations

Designing with Programmable Macro Logic

```
File name : 8BITADDR
Date : 9/15/1987
Time : 9:37:36
@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
8 Bit Carry Look-Ahead Adder

@INTERNAL NODE
 g8, g1, g2, g3, g4, g5, g6, g7;
 p8, p1, p2, p3, p4, p5, p6, p7;
gn8,gn1,gn2,gn3,gn4,gn5,gn6,gn7;
c1,c2,c3,c4,c5,c6,c7;
g1_1,g2_1;
@COMMON PRODUCT TERM
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION

"level-0 functions"
gn1 = /(a0*b0);
p1 = /(/a0*/b0);
g1 = /gn1;

gn2 = /(a1*b1);
p2 = /(/a1*/b1);
g2 = /gn2;

gn3 = /(a2*b2);
p3 = /(/a2*/b2);
g3 = /gn3;

gn4 = /(a3*b3);
p4 = /(/a3*/b3);
g4 = /gn4;

gn5 = /(a4*b4);
p5 = /(/a4*/b4);
g5 = /gn5;
```

Figure 6.17 8-Bit Adder Boolean Equations

```
gn6 = /(a5*b5);
p6 = /(/a5*/b5);
g6 = /gn6;

gn7 = /(a6*b6);
p7 = /(/a6*/b6);
g7 = /gn7;

gn8 = /(a7*b7);
p8 = /(/a7*/b7);
g8 = /gn8;

"level-1 functions"
g1_1 = g4 + p4*g3 + p4*p3*g2 + p4*p3*p2*g1;
g2_1 = g8 + p8*g7 + p8*p7*g6 + p8*p7*p6*g5;

"carry information"
c1 = g1 + p1*c0;
c2 = g2 + p2*g1 + p2*p1*c0;
c3 = g3 + p3*g2 + p3*p2*g1 + p3*p2*p1*c0;
c4 = g1_1 + p4*p3*p2*p1*c0;
c5 = g5 + p5*g1_1 + p5*p4*p3*p2*p1*c0;
c6 = g6 + p6*g5 + p6*p5*g1_1 + p6*p5*p4*p3*p2*p1*c0;
c7 = g7 + p7*g6 + p7*p6*g5 + p7*p6*p5*g1_1 +
          p7*p6*p5*p4*p3*p2*p1*c0;
c8 = g2_1 + p8*p7*p6*p5*g1_1 + p8*p7*p6*p5*p4*p3*p2*p1*c0;

"addition functions"
sum0 : xr1 = c0;
       xr2 = p1 * gn1;
sum1 : xr1 = c1;
       xr2 = p2 * gn2;
sum2 : xr1 = c2;
       xr2 = p3 * gn3;
sum3 : xr1 = c3;
       xr2 = p4 * gn4;
sum4 : xr1 = c4;
       xr2 = p5 * gn5;
sum5 : xr1 = c5;
       xr2 = p6 * gn6;
sum6 : xr1 = c6;
       xr2 = p7 * gn7;
sum7 : xr1 = c7;
       xr2 = p8 * gn8;
```

Figure 6.17 8-Bit Adder Boolean Equations (Continued)

```
" 8 Bit Adder Simulation input
"
" <--------INPUTS--------> < B,/B >
" 2222HHHHHHHHHH1
" 32109876543210987654321O 76543210
"
  LLLLLLLLLLLLLLLLLHHHHHHH -------- "0 + 0"
  HHHHHHHLLLLLLLLLHHHHHHH -------- "0 + FF"
  HHHHHHHLLLLLLLLLHLHHHHHH -------- "1 + FF"
  LHHHHHHHLLLLLLLLHHHHHHH -------- "0 + 7F + CARRY IN"
  HLHLHLHLLHLHLHLHHHHHHH -------- "AA + 55"
  HLHLHLHLLHLHLHLHHHHHHH -------- "AA + 55 + CARRY IN"
  LLHHHHHHLLLHHHHLHHHHHH -------- "3F + 1F"
QUIT
```

Figure 6.18 8-Bit Adder Simulation Input File

Designing with Programmable Macro Logic

```
PLHS501           8BITADDR        Time = 15:41:33 Date = 9/14/1987
"
" <--------INPUTS--------> < B,/B > < XOR  > < /O,O >  TRACE TERMS
" 222211111111111
" 32109876543210987654321O 76543210 76543210 76543210
"
  000000000000000001111111 HHHHLLLL LLLLLLLL LLLLHHHL ;
  111111100000000101111111 HHHHLLLL HHHHHHHH LLLLHHHL ;
  111111100000001001111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  011111110000000011111111 HHHHLLLL HLLLLLLL LLLLHHHL ;
  101010100101010101111111 HHHHLLLL HHHHHHHH LLLLHHHL ;
  101010100101010111111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  001111110001111101111111 HHHHLLLL LHLHHHHL LLLLHHHL ;
"
" ------------------------- --------   I/O CONTROL LINES
"                           OOOOOOOO   DESIGNATED I/O USAGE
"                           OOOOOOOO   ACTUAL I/O USAGE
"
" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.19 8-Bit Adder Simulation Log File

Designing with Programmable Macro Logic

```
File Name : ENCODER
Date : 9/15/1987
Time : 9:38:43

################## P I N   L I S T ##################

Left                                                    Right
LABEL   ** FNC **PIN            PIN** FNC **  LABEL
VCC     ** +5V **  8-|        |-46 ** +5V **  VCC
I18N    ** I   **  9-|        |-45 ** I   **  I4N
I19N    ** I   ** 10-|        |-44 ** I   **  I3N
I20N    ** I   ** 11-|   P    |-43 ** I   **  I2N
I21N    ** I   ** 12-|   L    |-42 ** I   **  I1N
I22N    ** I   ** 13-|   H    |-41 ** I   **  I0N
I23N    ** I   ** 14-|   S    |-40 ** I   **  I31N
I24N    ** I   ** 15-|   5    |-39 ** I   **  I30N
I25N    ** I   ** 16-|   0    |-38 ** I   **  I29N
I26N    ** I   ** 17-|   1    |-37 ** I   **  I28N
I27N    ** I   ** 18-|        |-36 ** O   **  N/C
N/C     ** O   ** 19-|        |-35 ** O   **  A4N
GND     ** 0V  ** 20-|        |-34 ** 0V  **  GND


Bottom                                                  Top
LABEL   ** FNC **PIN            PIN** FNC **  LABEL
N/C     ** O   ** 21-|        |- 7 ** I   **  I17N
N/C     ** O   ** 22-|        |- 6 ** I   **  I16N
N/C     ** O   ** 23-|        |- 5 ** I   **  I15N
N/C     ** /O  ** 24-|   P    |- 4 ** I   **  I14N
N/C     ** /O  ** 25-|   L    |- 3 ** I   **  I13N
N/C     ** /O  ** 26-|   H    |- 2 ** I   **  I12N
N/C     ** /O  ** 27-|   S    |- 1 ** I   **  I11N
GSN     ** O   ** 28-|   5    |-52 ** I   **  I10N
EON     ** O   ** 29-|   0    |-51 ** I   **  I9N
A0N     ** O   ** 30-|   1    |-50 ** I   **  I8N
A1N     ** O   ** 31-|        |-49 ** I   **  I7N
A2N     ** O   ** 32-|        |-48 ** I   **  I6N
A3N     ** O   ** 33-|        |-47 ** I   **  I5N
```

Figure 6.20 Encoder Pin List

32- to 5-BIT PRIORITY ENCODER

This relatively simple example demonstrates the capability of the PLHS501 to be programmed with functions that are not available in 'standard' device libraries. The equations may look difficult at first glance. However, there is a pattern to the encoding. Referring to figure 6.21, Lab4 – Lab1 are terms that are common to several outputs (A4n – A0n). Separating them from the main equations allows a total reduction in the numbers of gates used.

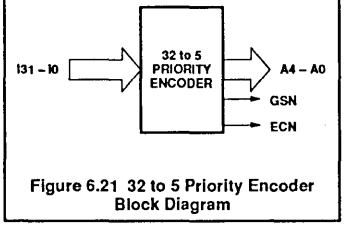


Figure 6.21 32 to 5 Priority Encoder Block Diagram

Designing with Programmable Macro Logic

```
File Name : ENCODER
Date : 9/15/1987
Time : 9:39:1

@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
32 TO 5 PRIORITY ENCODER

@COMMON PRODUT TERM
cpt1 = i26n*i27n*i28n*i29n*i30n*i31n;
cpt2 = i20n*i21n*i22n*i23n*i24n*i25n;
cpt3 = i14n*i15n*i16n*i17n*i18n*i19n;
cpt4 = i8n*i9n*i10n*i11n*i12n*i13n;

@INTERNAL NODE
eo lab1 lab2 lab3 lab4
@I/O DIRECITON
@I/O STEERING

@LOGIC EQUATION
lab1 =  ( /i31n
        + [/27n*i28n*i29n*i30n*i31n]
        + [/i23n*i24n*i25n*cpt1]
        + [/i9n*cpt2*cpt1]
        + [/i15n*i16n*i17n*i18n*i19n*cpt2*cpt1]
        + [/i11n*i12n*i13n*cpt3*cpt2*cpt1]
        + [/i7n*cpt4*cpt3*cpt2*cpt1]
        + [/i3n*i4n*i5n*i6n*i7n*cpt4*cpt3*cpt2*cpt1]);
lab2 =  ( [/i23n*i24n*i25n*cpt1]
        + [/i22n*i23n*i24n*i25n*cpt1]
        + [/i21n*i22n*i23n*i24n*i25n*cpt1]
        + [/i20n*i21n*i22n*i23n*i24n*i25n*cpt1]);
lab3 =  ( [/i15n*i16n*i17n*i18n*i19n*cpt2*cpt1]
        + [/i14n*i15n*i16n*i17n*i18n*i19n*cpt2*cpt1]
        + [/i13n*cpt3*cpt2*cpt1]
        + [/i12n*i13n*cpt3*cpt2*cpt1]);
lab4 =  ( /i31n
        + [/i30n*i31n]
        + [/i29n*i30n*i31n]
        + [/i28n*i29n*i30n*i31n]
        + [/i27n*i28n*i29n*i30n*i31n]
        + [/i26n*i27n*i28n*i29n*i30n*i31n]
        + [/i25n*cpt1]
        + [/i24n*i25n*cpt1]);
eo   = /(i0n*i1n*i2n*i3n*i4n*i5n*i6n*i7n
        *i8n*i9n*i10n*i11n*i12n*i13n*i14n*i15n
        *i16n*i17n*i18n*i19n*i20n*i21n*i22n*i23n
        *i24n*i25n*cpt1);
```

Figure 6.22 Encoder Boolean Equations

Designing with Programmable Macro Logic

```
gsn = /eo;
eon = eo;
a0n =  /( lab1
        + [/i29n*i30n*i31n]
        + [/i25n*cpt1]
        + [/i21n*i22n*i23n*i24n*i25n*cpt1]
        + [/i17n*i18n*i19n*cpt2*cpt1]
        + [/i13n*cpt3*cpt2*cpt1]
        + [/i9n*i10n*i11n*i12n*i13n*cpt3*cpt2*cpt1]
        + [/i5n*i6n*i7n*cpt4*cpt3*cpt2*cpt1]
        + [/i1n*i2n*i3n*i4n*i5n*i6n*i7n*cpt4*cpt3*cpt2*cpt1]);
a1n =  /( lab1
        + [/i30n*i31n]
        + [/i26n*i27n*i28n*i29n*i30n*i31n]
        + [/i22n*i23n*i24n*i25n*cpt1]
        + [/i18n*i19n*cpt2*cpt1]
        + [/i14n*i15n*i16n*i17n*i18n*i19n*cpt2*cpt1]
        + [/i10n*i11n*i12n*i13n*cpt3*cpt2*cpt1]
        + [/i6n*i7n*cpt4*cpt3*cpt2*cpt1]
        + [/i2n*i3n*i4n*i5n*i6n*i7n*cpt4*cpt3*cpt2*cpt1]);
a2n =  /( /i31n
        + [/i30n*i31n]
        + [/i29n*i30n*i31n]
        + [/i28n*i29n*i30n*i31n]
        + lab2
        + lab3
        + [/i7n*cpt4*cpt3*cpt2*cpt1]
        + [/i6n*i7n*cpt4*cpt3*cpt2*cpt1]
        + [/i5n*i6n*i7n*cpt4*cpt3*cpt2*cpt1]
        + [/i4n*i5n*i6n*i7n*cpt4*cpt3*cpt2*cpt1]);
a3n =  /( lab4
        + lab3
        + [/i11n*i12n*i13n*cpt3*cpt2*cpt1]
        + [/i10n*i11n*i12n*i13n*cpt3*cpt2*cpt1]
        + [/i9n*i10n*i11n*i12n*i13n*cpt3*cpt2*cpt1]
        + [/i8n*i9n*i10n*i11n*i12n*i13n*cpt3*cpt2*cpt1]);
a4n =  /( lab4
        + lab2
        + [/i19n*cpt2*cpt1]
        + [/i18n*i19n*cpt2*cpt1]
        + [/i17n*i18n*i19n*cpt2*cpt1]
        + [/i16n*i17n*i18n*i19n*cpt2*cpt1]);
```

Figure 6.22 Encoder Boolean Equations (Continued)

Designing with Programmable Macro Logic

```
" 32 to 5 Priority Encoder Simulation Input
"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHHHHHH "Inputs all high"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHHLHHH "I31 - LSB low"
  HHHHHHHHHHHHHHHHHHHHHHHL HHHHHHHH "I0  - MSB low"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHHHLHH "I30 low"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHHHHLH "I29"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHHHHHL "I28"
  HHHHHHHHHHHHHHHHHHHHHHHH LHHHHHHH "I27"
  HHHHHHHHHHHHHHHHHHHHHHHH HLHHHHHH "I26"
  HHHHHHHHHHHHHHHHHHHHHHHH HHLHHHHH "I25"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHLHHHH "I24"
  LHHHHHHHHHHHHHHHHHHHHHHH HHHHHHHH "I23"
  HLHHHHHHHHHHHHHHHHHHHHHH HHHHHHHH "I22"
  HHLHHHHHHHHHHHHHHHHHHHHH HHHHHHHH "I21"
  HHHLHHHHHHHHHHHHHHHHHHHH HHHHHHHH "I20"
  HHHHLHHHHHHHHHHHHHHHHHHH HHHHHHHH "I19"
  HHHHHLHHHHHHHHHHHHHHHHHH HHHHHHHH "I18"
  HHHHHHLHHHHHHHHHHHHHHHHH HHHHHHHH "I17"
  HHHHHHHLHHHHHHHHHHHHHHHH HHHHHHHH "I16"
  HHHHHHHHLHHHHHHHHHHHHHHH HHHHHHHH "I15"
  HHHHHHHHLHHHHHHHHHHHHHHH HHHHHHHH "I14"
  HHHHHHHHHLHHHHHHHHHHHHHH HHHHHHHH "I13"
  HHHHHHHHHHLHHHHHHHHHHHHH HHHHHHHH "I12"
  HHHHHHHHHHHLHHHHHHHHHHHH HHHHHHHH "I11"
  HHHHHHHHHHHHLHHHHHHHHHHH HHHHHHHH "I10"
  HHHHHHHHHHHHHLHHHHHHHHHH HHHHHHHH "I9"
  HHHHHHHHHHHHHHLHHHHHHHHH HHHHHHHH "I8"
  HHHHHHHHHHHHHHHLHHHHHHHH HHHHHHHH "I7"
  HHHHHHHHHHHHHHHHLHHHHHHH HHHHHHHH "I6"
  HHHHHHHHHHHHHHHHHLHHHHHH HHHHHHHH "I5"
  HHHHHHHHHHHHHHHHHHLHHHHH HHHHHHHH "I4"
  HHHHHHHHHHHHHHHHHHHLHHHH HHHHHHHH "I3"
  HHHHHHHHHHHHHHHHHHHHLHHH HHHHHHHH "I2"
  HHHHHHHHHHHHHHHHHHHHHLHH HHHHHHHH "I1"
  HHHHHHHHHHHHHHHHHHHHHHLH HHHHHHHH "I0"
  HHHHHHHHHHHHHHHHHHHHHHHH HHHHHHHH "ALL HIGH"
  LLLLLLLLLLLLLLLLLLLLLLLL LLLLLLLL "ALL LOW
  HHHLHHHLHHHHLHHHHLHHHLHH HHHLHHLH "Several simultaneously"
QUIT
```

Figure 6.23 Encoder Simulation Input File

Designing with Programmable Macro Logic

```
PLHS501           ENCODER          Time = 15:50:28 Date = 9/14/1987
"
" <---------INPUTS--------> < B,/B > < XOR  > < /O,O >  TRACE TERMS
" 222211111111111
" 32109876543210987654321O 76543210 76543210 76543210
"
  111111111111111111111111 11111111 LHHHHHLH LLLLHHHH ;
  111111111111111111111111 11110111 LLLLLLHL LLLLHHHH ;
  111111111111111111111110 11111111 LHHHHHHL LLLLHHHH ;
  111111111111111111111111 11111011 LLLLLHHL LLLLHHHH ;
  111111111111111111111111 11111101 LLLLHLHL LLLLHHHH ;
  111111111111111111111111 11111110 LLLLHHHL LLLLHHHH ;
  111111111111111111111111 01111111 LLLHLLHL LLLLHHHH ;
  111111111111111111111111 10111111 LLLHLHHL LLLLHHHH ;
  111111111111111111111111 11011111 LLLHHLHL LLLLHHHH ;
  111111111111111111111111 11101111 LLLHHHHL LLLLHHHH ;
  011111111111111111111111 11111111 LLHLLLHL LLLLHHHH ;
  101111111111111111111111 11111111 LLHLLHHL LLLLHHHH ;
  110111111111111111111111 11111111 LLHLHLHL LLLLHHHH ;
  111011111111111111111111 11111111 LLHLHHHL LLLLHHHH ;
  111101111111111111111111 11111111 LLHHLLHL LLLLHHHH ;
  111110111111111111111111 11111111 LLHHLHHL LLLLHHHH ;
  111111011111111111111111 11111111 LLHHHLHL LLLLHHHH ;
  111111101111111111111111 11111111 LLHHHHHL LLLLHHHH ;
  111111110111111111111111 11111111 LHLLLLHL LLLLHHHH ;
  111111111011111111111111 11111111 LHLLLHHL LLLLHHHH ;
  111111111101111111111111 11111111 LHLLHLHL LLLLHHHH ;
  111111111110111111111111 11111111 LHLLHHHL LLLLHHHH ;
  111111111111011111111111 11111111 LHLHLLHL LLLLHHHH ;
  111111111111101111111111 11111111 LHLHLHHL LLLLHHHH ;
  111111111111110111111111 11111111 LHLHHLHL LLLLHHHH ;
  111111111111111011111111 11111111 LHLHHHHL LLLLHHHH ;
  111111111111111101111111 11111111 LHHLLLHL LLLLHHHH ;
  111111111111111110111111 11111111 LHHLLHHL LLLLHHHH ;
  111111111111111111011111 11111111 LHHLHLHL LLLLHHHH ;
  111111111111111111101111 11111111 LHHLHHHL LLLLHHHH ;
  111111111111111111110111 11111111 LHHHLLHL LLLLHHHH ;
  111111111111111111111011 11111111 LHHHLHHL LLLLHHHH ;
  111111111111111111111101 11111111 LHHHHLHL LLLLHHHH ;
  111111111111111111111110 11111111 LHHHHHHL LLLLHHHH ;
  111111111111111111111111 11111111 LHHHHHLH LLLLHHHH ;
  000000000000000000000000 00000000 LLLLLLHL LLLLHHHH ;
  111011101111011110111011 11101101 LLLLHLHL LLLLHHHH ;
"
" ------------------------ --------   I/O CONTROL LINES
"                          IIIIIIII   DESIGNATED I/O USAGE
"                          IIIIIIII   ACTUAL I/O USAGE
"
" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.24 Encoder Simulation Log File

Designing with Programmable Macro Logic

8-BIT SHIFT REGISTER WITH SEQUENCE DETECTOR

This example demonstrates an application using D type edge-triggered flip-flops. Six NAND gates are used for each flip-flop (Figure 3.2). Note that to add an asynchronous reset and/or set to any flip-flop requires no additional gates. Also, every flip-flop must have a reset or set line to initialize it. Without being initialized, the simulator will not be able to determine the output states as it could power-up in either a set or reset condition. An uninitialized flip-flop will cause AMAZE 1.6 to display a message indicating the outputs are not stabilized within a certain time period.

As can be seen from the block diagram (Figure 6.26) this design consists fo an 8-bit shift register, 3-bit ripple counter and 2 flip-flops that are set only upon detection of specific patterns. The patterns are read from the Q and QN outputs of the shift register. Since the input to the second flip-flop has the output of the first flip-flop as a product term, detection of the first pattern is a requirement for the detection of the second.

```
################## P I N   L I S T ##################

Left                                                Right
LABEL   ** FNC **PIN            PIN** FNC **  LABEL
VCC     ** +5V **  8-|        |-46 ** +5V **  VCC
N/C     ** I   **  9-|        |-45 ** I   **  N/C
N/C     ** I   ** 10-|        |-44 ** I   **  N/C
N/C     ** I   ** 11-|   P    |-43 ** I   **  N/C
DATA    ** I   ** 12-|   L    |-42 ** I   **  N/C
CLK     ** I   ** 13-|   H    |-41 ** I   **  N/C
RST     ** I   ** 14-|   S    |-40 ** /O  **  N/C
DET1    ** O   ** 15-|   5    |-39 ** /O  **  CQ2
DET1N   ** O   ** 16-|   0    |-38 ** /O  **  CQ1
DET2    ** O   ** 17-|   1    |-37 ** /O  **  CQ0
DET2N   ** O   ** 18-|        |-36 ** O   **  O7
N/C     ** O   ** 19-|        |-35 ** O   **  O6
GND     ** 0V  ** 20-|        |-34 ** 0V  **  GND


Bottom                                              Top
LABEL   ** FNC **PIN            PIN** FNC **  LABEL
N/C     ** O   ** 21-|        |- 7 ** I   **  N/C
N/C     ** O   ** 22-|        |- 6 ** I   **  N/C
N/C     ** O   ** 23-|        |- 5 ** I   **  N/C
N/C     ** /O  ** 24-|   P    |- 4 ** I   **  N/C
N/C     ** /O  ** 25-|   L    |- 3 ** I   **  N/C
N/C     ** /O  ** 26-|   H    |- 2 ** I   **  N/C
N/C     ** /O  ** 27-|   S    |- 1 ** I   **  N/C
O0      ** O   ** 28-|   5    |-52 ** I   **  N/C
O1      ** O   ** 29-|   0    |-51 ** I   **  N/C
O2      ** O   ** 30-|   1    |-50 ** I   **  N/C
O3      ** O   ** 31-|        |-49 ** I   **  N/C
O4      ** O   ** 32-|        |-48 ** I   **  N/C
O5      ** O   ** 33-|        |-47 ** I   **  N/C
```

Figure 6.25 8-Bit Shifter Pin List

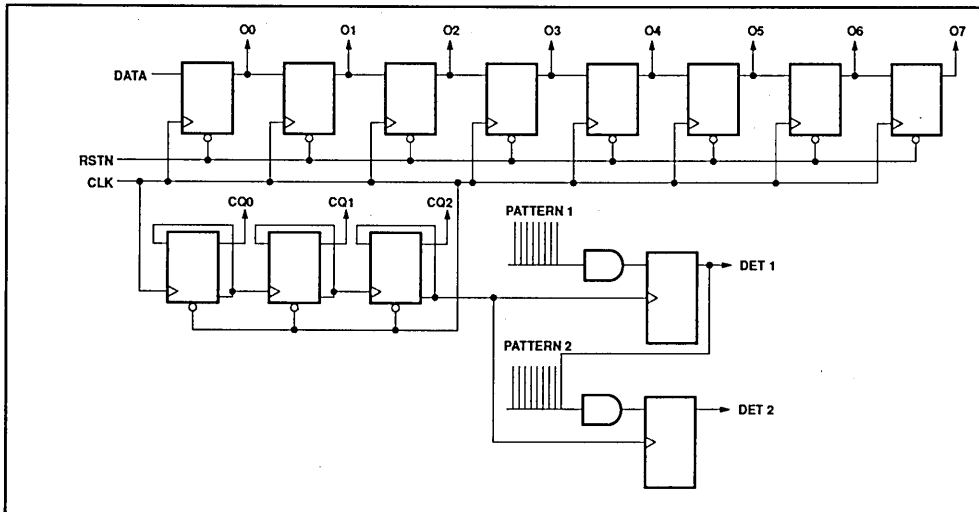


Figure 6.26 Shift Register with Sequence Detector Block Diagram

Designing with Programmable Macro Logic

```
File Name : 8BTSHFT
Date : 9/15/1987
Time : 9:41:35

@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION

8 Bit Shift Register with 16 bit Sequence Detector

@INTERNAL NODE
SN0,RN0,Q0;
SN1,RN1,Q1;
SN2,RN2,Q2;
SN3,RN3,Q3;
SN4,RN4,Q4;
SN5,RN5,Q5;
SN6,RN6,Q6;
SN7,RN7,Q7;
CSN0,CRN0,CQN0;
CSN1,CRN1,CQN1;
CSN2,CRN2,CQN2;
PSN0,PRN0;
PSN1,PRN1;
@COMMON PRODUCT TERM
PAT1 = Q7*Q6*Q5*Q4*Q3*Q2*Q1*Q0;
PAT2 = Q7*Q6*Q5*Q4*Q3*Q2*Q1*Q0;
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION

"8 D-TYPE FLIP FLOPS CONNECTED AS SHIFT REGISTER"

SN0 = /(CLK*RST*(/(SN0*(/[DATA*RST*RN0]))));
RN0 = /(SN0*CLK*(/[DATA*RST*RN0]));
Q0  = /(SN0*(/[RN0*Q0*RST]));

SN1 = /(CLK*RST*(/(SN1*(/[Q0*RST*RN1]))));
RN1 = /(SN1*CLK*(/[Q0*RST*RN1]));
Q1  = /(SN1*(/[RN1*Q1*RST]));

SN2 = /(CLK*RST*(/(SN2*(/[Q1*RST*RN2]))));
RN2 = /(SN2*CLK*(/[Q1*RST*RN2]));
Q2  = /(SN2*(/[RN2*Q2*RST]));

SN3 = /(CLK*RST*(/(SN3*(/[Q2*RST*RN3]))));
RN3 = /(SN3*CLK*(/[Q2*RST*RN3]));
Q3  = /(SN3*(/[RN3*Q3*RST]));
```

Figure 6.27 8-Bit Shifter Boolean Equations

Designing with Programmable Macro Logic

```
SN4 = /(CLK*RST*(/(SN4*(/[Q3*RST*RN4]))));
RN4 = /(SN4*CLK*(/[Q3*RST*RN4]));
Q4  = /(SN4*(/[RN4*Q4*RST]));

SN5 = /(CLK*RST*(/(SN5*(/[Q4*RST*RN5]))));
RN5 = /(SN5*CLK*(/[Q4*RST*RN5]));
Q5  = /(SN5*(/[RN5*Q5*RST]));

SN6 = /(CLK*RST*(/(SN6*(/[Q5*RST*RN6]))));
RN6 = /(SN6*CLK*(/[Q5*RST*RN6]));
Q6  = /(SN6*(/[RN6*Q6*RST]));

SN7 = /(CLK*RST*(/(SN7*(/[Q6*RST*RN7]))));
RN7 = /(SN7*CLK*(/[Q6*RST*RN7]));
Q7  = /(SN7*(/[RN7*Q7*RST]));

O0 = Q0;
O1 = Q1;
O2 = Q2;
O3 = Q3;
O4 = Q4;
O5 = Q5;
O6 = Q6;
O7 = Q7;

"3 D-TYPE FLIP FLOPS CONNECTED AS A RIPPLE COUNTER"

CSN0 = /(CLK*RST*(/(CSN0*(/[CQN0*RST*CRN0]))));
CRN0 = /(CSN0*CLK*(/[CQN0*RST*CRN0]));
CQ0  = /(CSN0*CQN0);
CQN0 = /(CRN0*CQ0*RST);

CSN1 = /(CQN0*RST*(/(CSN1*(/[CQN1*RST*CRN1]))));
CRN1 = /(CSN1*CQN0*(/[CQN1*RST*CRN1]));
CQ1  = /(CSN1*CQN1);
CQN1 = /(CRN1*CQ1*RST);

CSN2 = /(CQN1*RST*(/(CSN2*(/[CQN2*RST*CRN2]))));
CRN2 = /(CSN2*CQN1*(/[CQN2*RST*CRN2]));
CQ2  = /(CSN2*CQN2);
CQN2 = /(CRN2*CQ2*RST);

"2 D-TYPE FLIP FLOPS USED FOR PATTERN SEQUENCE DETECTION.
Sequence to be detected is 16 bits - 55AA Hex.
When the pattern is detected, pin det2 will go high.

In this example, both pattern 1 and pattern 2 are set to FF hex. To
change the pattern to 55AA, the STD file (P68 and P70) was edited using
FTE. This was necessary because AMAZE 1.6 only allows 40 internal labels,
so it was not possible to reference the QN nodes of the shift register
flip-flops."

PSN0  = /(CQN2*RST*(/(PSN0*(/[PAT1*RST*PRN0]))));
PRN0  = /(PSN0*CQN2*(/[PAT1*RST*PRN0]));
DET1N = (PSN0*/DET1);
DET1  = (PRN0*/DET1N*RST);

PSN1  = /(CQN2*RST*(/(PSN1*(/[PAT2*DET1*RST*PRN1]))));
PRN1  = /(PSN1*CQN2*(/[PAT2*DET1*RST*PRN1]));
DET2N = (PSN1*/DET2);
DET2  = (PRS1*/DET2N*RST);
```

Figure 6.27 8-Bit Shifter Boolean Equations (Continued)

Designing with Programmable Macro Logic

```
--------------------------------   Col for    P 67
...A.................................A.................................
H--------------------------------   Col for    P 68
.....................................A..........A..A..A..A..A..A..A..A..
--------------------------------   Col for    P 69
.A...................................A.................................
H--------------------------H-----   Col for    P 70
.....................................A............A..A..A..A..A..A..A..A..
--------------------------------   Col for    P 71
.....................................................................
HHHH
```

Original STD file showing P 68 and P 70

```
--------------------------------   Col for    P 67
...A.................................A.................................
H--------------------------------   Col for    P 68
...........A.....A.....A.....A.......A.............A.....A.....A.....A..
--------------------------------   Col for    P 69
.A...................................A.................................
H--------------------------H-----   Col for    P 70
..............A.....A.....A.....A..A..................A.....A.....A.....
--------------------------------   Col for    P 71
.....................................................................
HHHH
```

Figure 6.28 Portion of STD Files

Designing with Programmable Macro Logic

```
" 8 Bit Shift Register Simulation Input
"
  LLLHHHHHHHHHHHHHHHHHHHHH --------
  LHLHHHHHHHHHHHHHHHHHHHHH --------
  LLHHHHHHHHHHHHHHHHHHHHHH --------
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 7 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 6 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 5 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 4 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 3 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 2 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 1 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 0 H"
  HLHHHHHHHHHHHHHHHHHHHHHH -------- "END OF FIRST SEQ"
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 7 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 6 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 5 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 4 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 3 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 2 L"
  HLHHHHHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "BIT 1 H"
  HLLHHHHHHHHHHHHHHHHHHHHH --------
  HHLHHHHHHHHHHHHHHHHHHHHH -------- "BIT 0 L"
  HLLHHHHHHHHHHHHHHHHHHHHH -------- "END OF SECOND SEQ"
  HHLHHHHHHHHHHHHHHHHHHHHH --------
  LHHHHHHHHHHHHHHHHHHHHHHH -------- "RESET"
QUIT
```

Figure 6.29 8-Bit Shifter Simulation Input File

Designing with Programmable Macro Logic

```
PLHS501           8BTSHFT          Time = 16:34:21 Date = 9/14/1987
"
" <--------INPUTS--------> < B,/B > < XOR  > < /O,O >  TRACE TERMS
" 222211111111111
" 32109876543210987654321O 76543210 76543210 76543210
"
  00011111111111111111111111 HLHLLLLL LLLLLLLL LLLLHHHH ;
  01011111111111111111111111 HLHLLLLL LLLLLLLL LLLLHHHH ;
  00111111111111111111111111 HLHLLLLL LLLLLLLL LLLLHHHH ;
  10011111111111111111111111 HLHLLLLL LLLLLLLL LLLLHHHH ;
  11011111111111111111111111 HLHLLLLH LLLLLLLL LLLLHHHH ;
  10111111111111111111111111 HLHLLLLH LLLLLLLL LLLLHHHH ;
  11111111111111111111111111 HLHLLLHL LLLLLLLH LLLLHHHH ;
  10011111111111111111111111 HLHLLLHL LLLLLLLH LLLLHHHH ;
  11011111111111111111111111 HLHLLLHH LLLLLLHL LLLLHHHH ;
  10111111111111111111111111 HLHLLLHH LLLLLLHL LLLLHHHH ;
  11111111111111111111111111 HLHLLHLL LLLLLHLH LLLLHHHH ;
  10011111111111111111111111 HLHLLHLL LLLLLHLH LLLLHHHH ;
  11011111111111111111111111 HLHLLHLH LLLLHLHL LLLLHHHH ;
  10111111111111111111111111 HLHLLHLH LLLLHLHL LLLLHHHH ;
  11111111111111111111111111 HLHLLHHL LLLHLHLH LLLLHHHH ;
  10011111111111111111111111 HLHLLHHL LLLHLHLH LLLLHHHH ;
  11011111111111111111111111 HLHLLHHH LLHLHLHL LLLLHHHH ;
  10111111111111111111111111 HLHLLHHH LLHLHLHL LLLLHHHH ;
  11111111111111111111111111 HLLHLLLL LHLHLHLH LLLLHHHH ;
  10111111111111111111111111 HLLHLLLL LHLHLHLH LLLLHHHH ;
  11111111111111111111111111 HLLHLLLH HLHLHLHH LLLLHHHH ;
  10011111111111111111111111 HLLHLLLH HLHLHLHH LLLLHHHH ;
  11011111111111111111111111 HLLHLLHL LHLHLHHL LLLLHHHH ;
  10111111111111111111111111 HLLHLLHL LHLHLHHL LLLLHHHH ;
  11111111111111111111111111 HLLHLLHH HLHLHHLH LLLLHHHH ;
  10011111111111111111111111 HLLHLLHH HLHLHHLH LLLLHHHH ;
  11011111111111111111111111 HLLHLHLL LHLHHLHL LLLLHHHH ;
  10111111111111111111111111 HLLHLHLL LHLHHLHL LLLLHHHH ;
  11111111111111111111111111 HLLHLHLH HLHHLHLH LLLLHHHH ;
  10011111111111111111111111 HLLHLHLH HLHHLHLH LLLLHHHH ;
  11011111111111111111111111 HLLHLHHL LHHLHLHL LLLLHHHH ;
  10111111111111111111111111 HLLHLHHL LHHLHLHL LLLLHHHH ;
  11111111111111111111111111 HLLHLHHH HHLHLHLH LLLLHHHH ;
  10011111111111111111111111 HLLHLHHH HHLHLHLH LLLLHHHH ;
  11011111111111111111111111 LHHLLLLL HLHLHLHL LLLLHHHH ;
  10011111111111111111111111 LHHLLLLL HLHLHLHL LLLLHHHH ;
  11011111111111111111111111 LHHLLLLH LHLHLHLL LLLLHHHH ;
  01111111111111111111111111 HLHLLLLL LLLLLLLL LLLLHHHH ;
"
" ------------------------ --------   I/O CONTROL LINES
"                          OOOOOOOO   DESIGNATED I/O USAGE
"                          OOOOOOOO   ACTUAL I/O USAGE
"
" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.30 8-Bit Shifter Simulation Log File

Designing with Programmable Macro Logic

4-BIT SYNCHRONOUS COUNTER

This counter produces a binary count on outputs Count3 – Count0. Note the required reset (RST) input to initialize all of the flip-flops. The inputs for each flip-flop were first determined by drawing the desired output waveforms. Next, Karnaugh maps were used to reduce the number of terms and determine the logic equations for the input to each flip-flop. This technique could be used to construct a counter whose outputs produce some count other than binary.

The simulation only consists of a reset, followed by a number of clocks to count from 0 through 15 and back to 0.

```
File Name : 4BTCOUNT
Date : 9/15/1987
Time : 9:57:5

################## P I N   L I S T ##################

Left                                                Right
LABEL   ** FNC **PIN            PIN** FNC **   LABEL
VCC     ** +5V **  8-|        |-46 ** +5V **   VCC
CLK     ** I   **  9-|        |-45 ** I   **   N/C
RST     ** I   ** 10-|        |-44 ** I   **   N/C
N/C     ** I   ** 11-|   P    |-43 ** I   **   N/C
N/C     ** I   ** 12-|   L    |-42 ** I   **   N/C
N/C     ** I   ** 13-|   H    |-41 ** I   **   N/C
N/C     ** I   ** 14-|   S    |-40 ** /O  **   N/C
COUNT0  ** O   ** 15-|   5    |-39 ** /O  **   N/C
COUNT1  ** O   ** 16-|   0    |-38 ** /O  **   N/C
COUNT2  ** O   ** 17-|   1    |-37 ** /O  **   N/C
COUNT3  ** O   ** 18-|        |-36 ** O   **   N/C
TC      ** O   ** 19-|        |-35 ** O   **   N/C
GND     ** 0V  ** 20-|        |-34 ** 0V  **   GND


Bottom                                              Top
LABEL   ** FNC **PIN            PIN** FNC **   LABEL
N/C     ** O   ** 21-|        |- 7 ** I   **   N/C
N/C     ** O   ** 22-|        |- 6 ** I   **   N/C
N/C     ** O   ** 23-|        |- 5 ** I   **   N/C
N/C     ** /O  ** 24-|   P    |- 4 ** I   **   N/C
N/C     ** /O  ** 25-|   L    |- 3 ** I   **   N/C
N/C     ** /O  ** 26-|   H    |- 2 ** I   **   N/C
N/C     ** /O  ** 27-|   S    |- 1 ** I   **   N/C
N/C     ** O   ** 28-|   5    |-52 ** I   **   N/C
N/C     ** O   ** 29-|   0    |-51 ** I   **   N/C
N/C     ** O   ** 30-|   1    |-50 ** I   **   N/C
N/C     ** O   ** 31-|        |-49 ** I   **   N/C
N/C     ** O   ** 32-|        |-48 ** I   **   N/C
N/C     ** O   ** 33-|        |-47 ** I   **   N/C
```

Figure 6.31 4-Bit Counter Pin List

Designing with Programmable Macro Logic

```
File Name : 4BTCOUNT
Date : 9/15/1987
Time : 9:57:28

@DEVICE TYPE
PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION

   4 bit synchronous counter

@INTERNAL NODE
data1,data2,data3;
csn0,crn0,cq9,cqn0;
csn1,crn1,cq1,cqn1;
csn2,crn2,cq2,cqn2;
csn3,crn3,cq3,cqn3;
@COMMON PRODUCT TERM
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION

"INPUTS FOR EACH FLIP-FLOP"

DATA1 = [(CQ1*CQN0)+(CQN1*CQ0)];
DATA2 = [(CQ0*CQ1*CQN2)+(CQN0*CQ2)+(CQN1*CQ2)];
DATA3 = [(CQN2*CQ3)+(CQN0*CQ3)+(CQ0*CQ1*CQ2*CQN3)+(CQN1*CQA3)];

"4 D-TYPE FLIP FLOPS CONNECTED AS A SYNCHRONOUS COUNTER"

CSN0 = /(CLK*RST(/(CSN0*(/[CQN0*RST*CRN0]))));
CRN0 = /(CSN0*CLK*(/[CQN0*RST*CRN0]));
CQ0  = /(CSN0*CQN0);
CQN0 = /(CRN0*CQ0*RST);

CSN1 = /(CLK*RST(/(CSN1*(/[DATA1*RST*CRN1]))));
CRN1 = /(CSN1*CLK*(/[DATA1*RST*CRN1]));
CQ1  = /(CSN1*CQN1);
CQN1 = /(CRN1*CQ1*RST);

CSN2 = /(CLK*RST(/(CSN2*(/[DATA2*RST*CRN2]))));
CRN2 = /(CSN2*CLK*(/[DATA2*RST*CRN2]));
CQ2  = /(CSN2*CQN2);
CQN2 = /(CRN2*CQ2*RST);

CSN3 = /(CLK*RST(/(CSN3*(/[DATA3*RST*CRN3]))));
CRN3 = /(CSN3*CLK*(/[DATA3*RST*CRN3]));
CQ3  = /(CSN3*CQN3);
CQN3 = /(CRN3*CQ3*RST);

"Connection to output pins"

count0=cq0;
count1=cq1;
count2=cq2;
count3=cq3;

"TERMINAL COUNT PIN"

TC=(CQ0*CQ1*CQ2*CQ3);
```

Figure 6.32 4-Bit Counter Boolean Equations

Designing with Programmable Macro Logic

```
"
" 4 Bit Synchronous Counter Simulation Input
"
  HHHHLLHHHHHHHHHHHHHHHHHH -------- "RESET"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT1"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT2"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT3"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT4"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT5"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT6"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT7"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT8"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT9"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT10"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT11"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT12"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT13"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT14"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT15"
  HHHHHLHHHHHHHHHHHHHHHHHH --------
  HHHHHHHHHHHHHHHHHHHHHHHH -------- "COUNT0"
QUIT
```

Figure 6.33 4-Bit Counter Simulation Input File

Designing with Programmable Macro Logic

```
PLHS501           4BTCOUNT        Time = 16:57:51 Date = 9/14/1987
" <--------INPUTS--------> < B,/B > < XOR  > < /O,O >  TRACE TERMS
" 222211111111111
" 32109876543210987654321O 76543210 76543210 76543210
"
  11110011111111111111111 LLLLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LLLLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LLLHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LLLHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LLHLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LLHLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LLHHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LLHHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LHLLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LHLLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LHLHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LHLHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LHHLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LHHLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 LHHHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 LHHHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HLLLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HLLLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HLLHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HLLHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HLHLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HLHLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HLHHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HLHHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HHLLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HHLLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HHLHLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HHLHLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HHHLLLLL LLLLLLLL LLLLHHHL ;
  11111011111111111111111 HHHLLLLL LLLLLLLL LLLLHHHL ;
  11111111111111111111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  11111011111111111111111 HHHHLLLL LLLLLLLL LLLLHHHH ;
  11111111111111111111111 LLLLLLLL LLLLLLLL LLLLHHHL ;
"
" ------------------------ --------   I/O CONTROL LINES
"                          OOOOOOOO   DESIGNATED I/O USAGE
"                          OOOOOOOO   ACTUAL I/O USAGE

" PINLIST...
" 14 13 12 11 10 09 07 06 05 04 03 02 01 52 51 50 49 48 47 45 44 43 42 41
" 18 17 16 15 40 39 38 37 36 35 33 32 31 30 29 28
" 27 26 25 24 23 22 21 19 ;
```

Figure 6.34 4-Bit Counter Simulation Log File

Bibliography

1. Cavalan, N., 1984, Third Generation PLD Architecture Breaks AND-OR Bottlenecks. Proceedings of WESCON.
2. Wong, D., 1985, New Developments in Programmable Logic Devices. MIDCON Session 19. Chicago.
3. Gheissari, A. and Safari, B., 1987, High Speed, High Complexity PLDs and Applications. ELECTRO–87 Session 35.
4. Hartman, Robert, May 1984, Estimating Gate Complexity of Programmable Logic Devices. VLSI Design. PP 100 – 103.
5. Minnick, R.C., Dec. 1964, Cutpoint Cellular Logic, I.E.E.E. Transactions on Electronic Computers, PP 685 – 698.
6. Minnick, R.C., 1965, Cobweb Cellular Arrays, Proceedings of the Fall Joint Computer Conference, PP 327 – 341. Las Vegas, NV.
7. Canaday, R., 1965, Two-Dimensional Iterative Logic, Proceedings of the Fall Joint Computer Conference, PP 343 – 353. Las Vegas, NV.
8. Gardner, P.L., Dec. 1970, Functional Memory and its Microprogramming Implications, Technical Report TR.12. 091, IBM United Kingdom Laboratories Limited. Hursley Park, Winchester Hampshire.
9. Roth Jr., C.H., Fundamentals of Logic Design, Second Edition, West Publishing Co., 1979, PP 136 – 144. St. Paul, MN.

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

INTRODUCTION

This document is written assuming the reader is familiar with Signetics PLHS501. As well, we shall assume familiarity with the predecessor document "Designing with PML" and some exposure to Signetics AMAZE software. The goal of this document (i.e., Vol. 2) is to expand on the original ideas and present some cookbook solutions to some useful design problems. Vol. 2 also reflects nearly a year of experience through the multitude of design-ins achieved with the PLHS501. In fact, several of the design solutions presented here were contributions from our customers through our field applications organization. Designs we have encountered fell into a couple of interesting categories. First, many users view the part as a natural step in eliminating extraneous board "glue" (10 or more chips) or eliminating multiple programmable array logic devices (usually 3 to 5 units). Others recognized the PLHS501 capabilities of extremely wide logic functions and still others chose to invent their own solutions to standard bus interfaces. Commercially available bus interfaces often "miss the mark" and creative designers wish to implement exactly the functions they need in a concise, effective manner. To date, we have seen PLHS501 interfaces to the VME Bus II, FAST Bus, NuBus, GPIB and the IBM Micro Channel for the PS/2 system.

Before presenting these solutions however, it is appropriate to review the PML basics and expand on a number of issues which have been found to be important but which were previously treated lightly.

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

PLHS501 REVIEW

The PLHS501 is a 52-Pin, bipolar programmable logic device with a very powerful architecture. Unlike classic AND/OR based architectures, its basic building block is the NAND function which is configured in a foldback programming array. By cascading successive NAND functions through the array, both combinational and sequential structures may be obtained. The PLHS501 has 24 dedicated inputs, 16 outputs (with several varieties) and eight bidirectional pins. The internal NANDs may be cascaded to any depth needed, to achieve effective solutions using logic structures such as muxes, decoders and flip-flops without going off chip and wasting I/O pins to achieve cascading. To use the PLHS501 effectively, the designer should attempt to fold in function and remain within the chip as much as possible before exiting.

Figure 2–1 shows the PLHS501 architecture and illustrates several of the timing paths for internal signals to give the designer a feeling for maximum time delay within the part. These numbers are worst case maximums, regardless of switching directions, so the user may be assured that in general, the PLHS501 will be faster than these numbers.

The shorthand notation of Figure 2–1 hides something with which many designers have been impressed in the PLHS501, the wide input NAND gates. Figure 2–2 shows just how wide the internal NANDs are, from a logical viewpoint. Each NAND can accommodate up to 32 external inputs and 72 internal inputs. Hence the part is ideal for wide decoding of 32-bit address and data busses. With 72 copies of the wide NAND, the PLHS501 is often compared against low-end gate arrays. While flatter4ing, this gives no usable method to determine the degree to which functions can be fit into the device. As a rule of thumb, the PLHS501 can accommodate three or more PLA devices and usually four to five PAL® devices.

For any particular design, the user should refer to Table 2–1 and evaluate his/her design incrementally, tallying against a 72 gate budget. This is a ballpark estimation against the NAND capacity of the core of the part. The clever designer will find additional function by correctly exploiting the output logic.

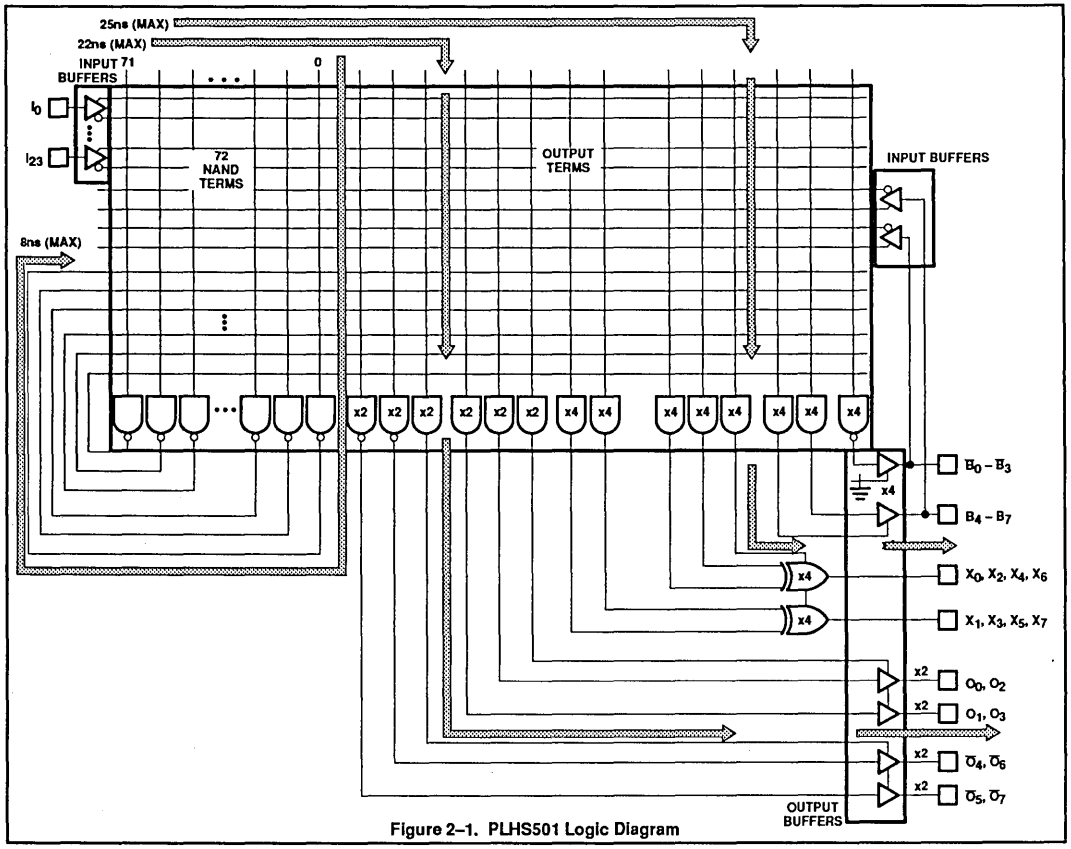


Figure 2–1. PLHS501 Logic Diagram

PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

Table 2–1. PLHS501 Gate Count Equivalents

FUNCTION	INTERNAL NAND EQUVALENT	COMMENTS
Gates		
NANDs	1	For 1 to 32 input variables
ANDs	1	For 1 to 32 input variables
NORs	1	For 1 to 32 input variables
ORs	1	For 1 to 32 input variables
Decoders		
3-to-8	8	Inverted inputs available
4-to-16	16	Inverted inputs available
5-to-32	32	Inverted inputs available (24 chip outputs only)
Encoders		
8-to-3	15	Inverted inputs, 2 logic levels
16-to-4	32	Inverted inputs, 2 logic levels
32-to-5	41	Inverted inputs, 2 logic levels, factored solution.
Multiplexers		
4-to-1	5	Inverted inputs available
8-to-1	9	
16-to-1	17	
27-to-1	28	Can address only 27 external inputs - more if internal
Flip-Flops		
D-type Flip-Flop	6	With asynchronous S-R
T-type Flip-Flop	6	With asynchronous S-R
J-K-type Flip-Flop	10	With asynchronous S-R
Adders		
8-bit	45	Full carry-lookahead (four levels of logic)
Barrel Shifters		
8-bit	72	2 levels of logic
Latches		
D-latch	3	2 levels of logic with one shared gate

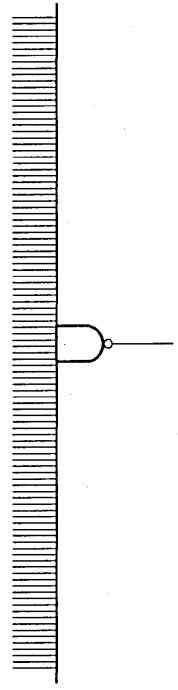


Figure 2–2. An Internal NAND Logic Equivalent

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

FLIP-FLOP BASICS

Most designers view flip-flops as black boxes with data inputs and outputs as well as additional control inputs. Some flip-flops are designed as primitive transistor structures, but in the past, gate array designers used their elementary building block, the NAND gate, to make flip-flops. Because the PLHS501 is also largely structured from NANDs, we can draw upon years of well known NAND-based flip-flop designs to readily implement flip-flops within the PLHS501.

Figures 3–1, 3–2, 3–3 and 3–4 give single sheet summaries of several flip-flop configurations. It should be noted that the transparent latch is recommended for data capturing, but not for state machines due to potential glitching. The edge triggered D-type is a convenient building block. Although external gates are saved with the J-K structure, it is at the expense of additional NANDs within the J-K flip-flop itself.

Notation

The delay of a NAND gate is most often designated as t_{PLH} or t_{PHL}, indicating that the gate output makes a High-to-Low (t_{PHL}) or Low-to-High (t_{PLH}) transition. For the flip-flops' transition, the High-to-Low ID is D0 and the Low-to-High ID is D1. This also holds true for structures fully contained within the foldback core, because input and output time delays will differ and change the performance. Knowing the basic concepts, the designer can expand these structures to include I/O pins and generate flip-flops wrapped around the part – but, he must derate his parameters accordingly to reflect the slower paths.

Because it will be lengthy to explain all of the flip-flop configurations given, we will show only one in some detail. The interested reader can verify the rest by manual analysis or by digital simulation. The Table 3-1 gives the typical and worst case values for an internal foldback NAND gate.

The single D-latch with enable Active-HIGH can be described in terms of the propagation delay formula given in Figure 3–1. For instance, the first propagation is for D to Q, where the Q output transitions from High-to-Low (i.e., t_{PD0}). To do this, assume Q is high so the /Q term is Low. To switch the state, /Q must be flipped first. Hence, the logic variable enters G2, then passes through G7, G4 and finally G3. This presents four transitions, two from Low-to-High (G2 and G4 outputs) and two from High-to-Low (G1 and G3 outputs). Hence, the formula reflects 2(d1&d0) which, using Table 3–1, gives 2(8+6.5)=29ns.

This is the worst case value, using typicals will give a value of 2(5.5+6.5)=24ns.

Switching in the other direction is a little different. Assuming t_{PD1} goes from Q=0 to Q=1, the /Q signal must be initially 1. Hence, G3 is armed for immediate transition. Hence, the time delay is simply traversing G2 and G3. One of them will go High-to-Low (G2) and the other Low-to-High (G3). The formula reflects the sum of the two transitions: t_{PD1} = d1+d0. From the table, this is 14.5ns (worst case) or 12ns (typical). The rest of the formula must be similarly analyzed, but the method is straightforward.

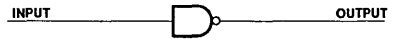
Flip-Flop Merging

Figure 3–5(A) shows the positive edge triggered D flip-flop structure. By putting a two-level AND/OR structure in front of the data input, the D flip-flop can be steered from state to state.

Figure 3–5(B) shows such an input structure realized from a two-level NAND gate section.

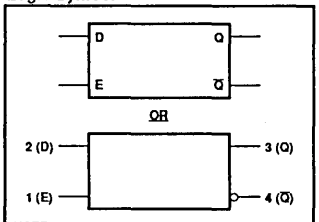
Figure 3–5(C) shows this "AND-OR" structure rolled inside of the flip-flop. The gating was merged with the flip-flop inwards to make a faster, composite function. Whereas this may appear as a trick to the uninitiated, this degree of flexibility allowed gate array designers to merge a multitude of logic into a fixed foundation. For highest efficiency, similar thinking allows the designer to break up decoders and multiplexers into their building blocks and generate only the pieces needed.

Table 3–1. Internal Fold Back NAND Gate



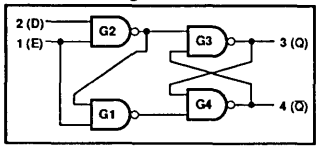
SYMBOL	PARAMETER		LIMITS		UNIT
	TO (OUTPUT)	FROM (INPUT)	Min	Max	
t_{PHL}		ANY	5.5	6.5	ns
t_{PLH}			6.5	8.0	

Logic Symbol



NOTE:
This part is functionally similar to the 7475.

NAND Gate Diagram



Function (Truth) Table

OPERATING MODE	E	D	Q	Q̄
DATA ENABLED	1	0	0	1
	1	1	1	0
DATA LATCHED	0	X	0_{n-1}	$\bar{0}_{n-1}$

NOTES:
1. When input Enable (E) is High, data enters the latch at "D" and appears at outputs "Q" and "Q̄"; the "Q" output follows the data as long as E is High. One setup time before the High-to-Low transition of E, D is stored in the latch; the latched outputs remain stable as long as E is Low.
2. 0_{n-1} = state before High-to-Low transition of E.
3. X = don't care.

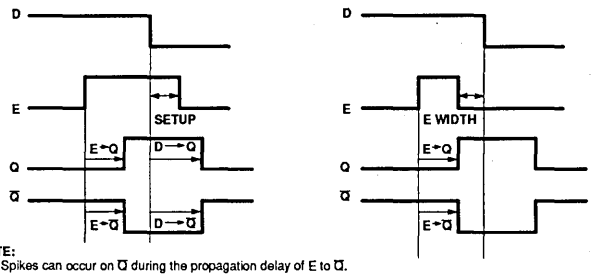
Propagation Delay Formulas:

FROM	TO	FORMULAS
D	Q	t_{PD0} = d1+d0 x 2
		t_{PD1} = d1+d0
D	Q̄	t_{PD0} = d1 + 2Xd0
		t_{PD1} = 2xd1 + d0
E	Q	t_{PD0} = d1 + 2Xd0
		t_{PD1} = d1+d0
E	Q̄	t_{PD0} = d1 + 2Xd0
		t_{PD1} = d1+d0
SETUP TIME		= d1 + 2Xd0
HOLD TIME		= 0
ENABLE PULSE		= E TO Q t_{PD0}

Propagation Delays:

PARAMETER	FROM	TO	DELAY in ns
t_{PD0} ↘	D	Q	21
	D	Q̄	21
	E	Q	21
	E	Q̄	21
t_{PD1} ↗	D	Q	14.5
	D	Q̄	22.5
	E	Q	14.5
	E	Q̄	14.5
D (SETUP TIME)			21
D (HOLD TIME)			0
ENABLE PULSE WIDTH			22

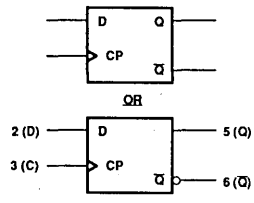
Timing Waveforms:



NOTE:
1. Spikes can occur on Q̄ during the propagation delay of E to Q̄.

Figure 3–1. Single D-Latch (Enable Active-HIGH)

Logic Symbol



NOTE:
This part is functionally similar to the 74LS74.

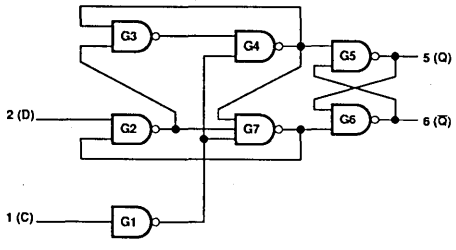
Propagation Delay Formulas:

FROM	TO	FORMULAS
C	Q	t_{PD1} = 2xd1+d0
		t_{PD0} = 2xd1+2xd0
C	$\overline{Q}$	t_{PD1} = 2xd1+d0
		t_{PD0} = 2 (d1+d0)
D (SETUP TIME)		= d0
D (HOLD TIME)		= d1+d0
CLOCK PULSE WIDTH		= 2xd0+d1

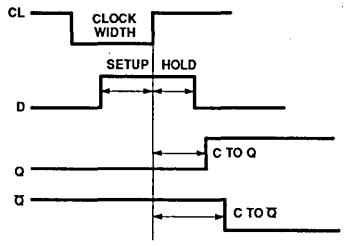
Propagation Delays:

PARAMETER	FROM	TO	DELAY in ns
t_{PD0} ↘	C	Q	29
	C	$\overline{Q}$	29
t_{PD1} ↗	C	Q	22.5
	C	$\overline{Q}$	22.5
D (SETUP TIME)	ns MIN		6.5
D (HOLD TIME)	ns MIN		14.5
ENABLE PULSE WIDTH	ns MIN		21

NAND Gate Diagram



Timing Waveforms:



NOTE:
1. Spikes can occur on $\overline{Q}$ during the propagation delay of C to $\overline{Q}$.

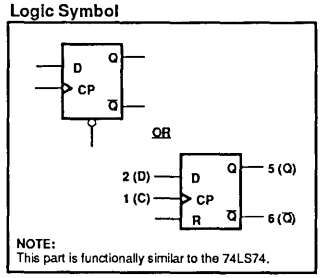
Function (Truth) Table

C	D	Q_{n+1}	$\overline{Q}_{n+1}$
↘	0	0	1
↘	1	1	0

NOTES:
1. Data is transferred to the outputs on the negative-going edge of the clock.
2. Q_{n+1} = state after High-to-Low transition of C.

Figure 3–2. D Flip-Flop (Negative Edge-Triggered)

Logic Symbol



NOTE:
This part is functionally similar to the 74LS74.

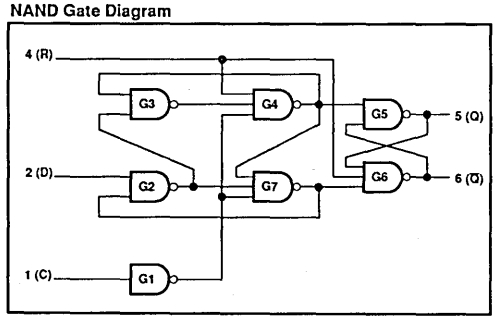
Propagation Delay Formulas:

FROM	TO	FORMULAS
C	Q	t_{PD1} = 2d1+d0
		t_{PD0} = 2(d1+d0)
C	Q̄	t_{PD1} = 2d1+d0
		t_{PD0} = 2(d1+d0)
R	Q	t_{PD0} = d1+d0
R	Q̄	t_{PD1} = d1
D (SETUP TIME)		= d0
D (HOLD TIME)		= d1 + d0
CL WIDTH (HIGH)		= d1 + 2d0
RESET WIDTH (LOW)		R→Q

Propagation Delays:

PARAMETER	FROM	TO	DELAY in ns
t_{PD1} ↗	C	Q	22.5
	C	Q̄	22.5
	R	Q	14.5
t_{PD0} ↘	C	Q	29
	C	Q̄	29
	R	Q̄	8
D (SETUP TIME)	ns Min		6.5
D (HOLD TIME)	ns Min		14.5
MINIMUM CLOCK WIDTH	ns		21
MINIMUM R WIDTH	ns		14.5

NAND Gate Diagram



Function (Truth) Table

C	D	Q_{n+1}	$\bar{Q}_{n+1}$
↘	0	0	1
↘	1	1	0
↘	0	0	1
↘	1	1	0

NOTES:
1. Data is transferred to the outputs on the negative-going edge of the clock; reset "R" (Active-LOW) is asynchronous and independent of
2. Q_{n+1} = state after High-to-Low transition of C.
3. X = don't care.

Timing Waveforms:

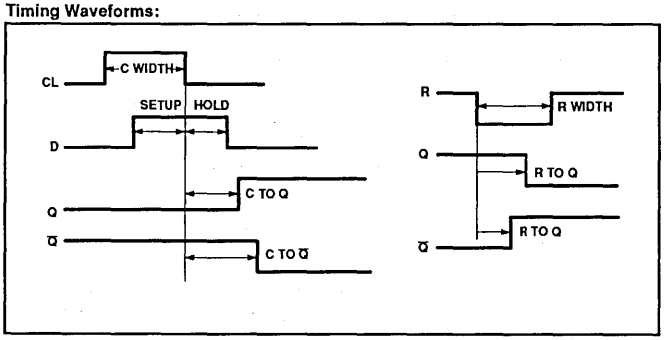
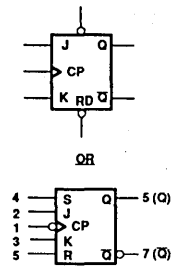


Figure 3–3. D Flip-Flop with Reset

Logic Symbol



NOTE:
This part is functionally similar to the 74LS109.

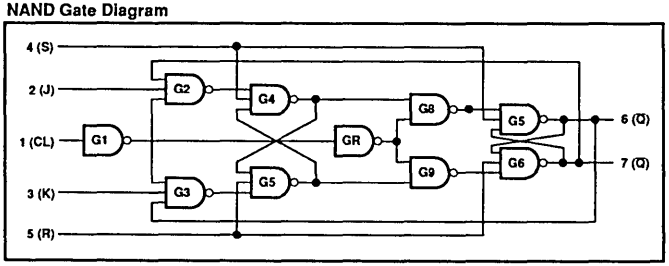
Propagation Delay Formulas:

FROM	TO	FORMULAS
C	Q, Q	t_{PD0} = 3d0+2d1
		t_{PD1} = 3(d1+d0)
S	Q	t_{PD1} = d1
R	Q	t_{PD1} = d1
S	Q	If C = 0 t_{PD0} = d1+d0
R	Q	If C = 1 t_{PD0} = 2(d1+d0)
J-K SETUP TIME		0
J-K HOLD TIME		= d0
PULSE WIDTH C (MIN)		= d1 + d0
S,R (MIN)		= (d1+d0)

Propagation Delays:

PARAMETER	FROM	TO	DELAY in ns
t_{PD0} ↘	C	Q & Q	35.5
	S	Q	14.5, 29
	R	Q	14.5, 29
t_{PD1} ↗	C	Q & Q	43.5
	S	Q	8
	R	Q	8
J-K SETUP TIME			0
J-K HOLD TIME			6.5
MINIMUM PULSE WIDTH			14.5
MINIMUM PULSE WIDTH S&R (LOW)			14.5

NAND Gate Diagram



Function (Truth) Table

C	J	K	S	R	Q_{n+1}	$\overline{Q}_{n+1}$
↑	0	0	1	1	Q_n	$\overline{Q}_n$
↑	1	0	1	1	1	0
↑	0	1	1	1	0	1
↑	1	1	1	1	Q	$\overline{Q}$
C	J	K	S	R	Q	$\overline{Q}$
X	X	X	1	0	0	1
X	X	X	0	1	1	0

NOTES:
1. Master/slave data is pulse-triggered and enters the J–K flip-flop when the clock (C) goes Low; the entered data is transferred to the outputs on the positive going edge of "C". Set (S) and Reset (R) inputs are asynchronous, independent signals and, when either is Active-LOW, the J–K and C inputs are overridden.
2. Q_{n+1} = next state of Q.
3. X = don't care.

Timing Waveforms:

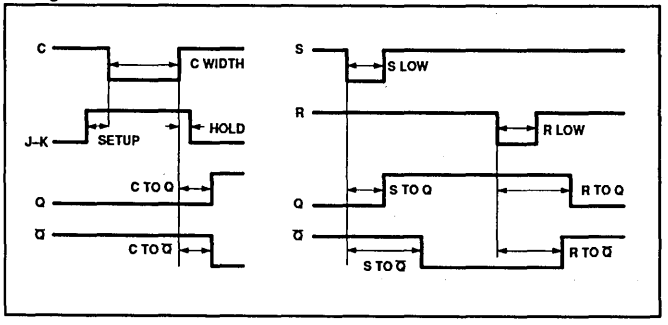


Figure 3–4. J-K Flip-Flop with Set and Reset

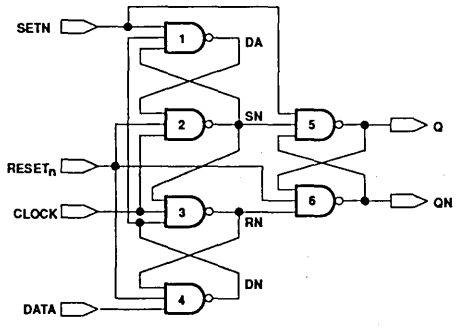


Figure 3–5(A). Positive Edge Triggered D-Flip-Flop with Reset and Set

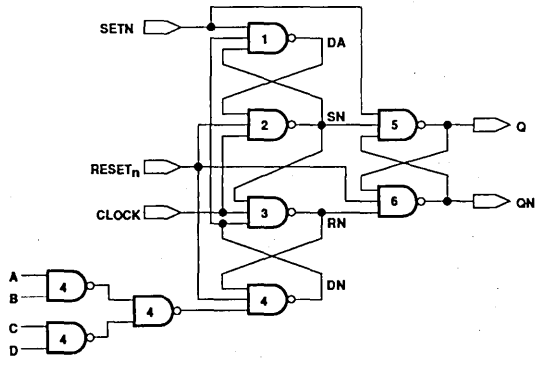


Figure 3–5(B). As in (A), with Input AND-OR Function

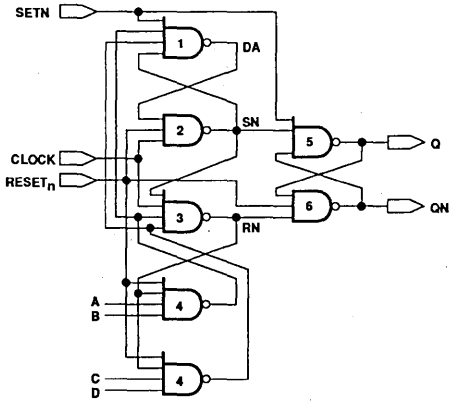


Figure 3–5(C). As Above, with Integral AND-OR Input Function

Figure 3–5. Flip-Flop Merging

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

VME Bus EXAMPLES

Omnibyte VSBC20 Mailbox Interrupt Structure

One of the more popular uses for the PLHS501 is interfacing with 32-bit microprocessors. This section illustrates some of the ways the part has been used with the popular VME Bus. The Omnibyte Corporation manufactures many VME Bus products (as well as others) and was kind enough to release a portion of their VSBC20 board design as an example of using the PLHS501 in a very flexible, user configurable interrupt generation device. The VSBC20 employs two PLHS501 parts, as shown in Figure 4–1. One device is used largely as an address decoder, the other, which is the object of this Section, is the configurable interrupt generator. The target micro- processor here is a 25MHz 68020 and the application is interrupt generation. The explanation is in the words of Glenn Case, the designer:

"Following the design philosophy of giving the user as much flexibility as possible, the local interrupt structure of the VSBC20 is implemented in a PLD. It is impossible to "optimize" the assignment of the local interrupts to the interrupt levels of the processor since they are application specific. One system may want the Serial I/O and Parallel I/O to have higher levels than the Omnimodule Interrupts while yet another, using a SCSI Omnimodule, may want it to have higher level interrupts. Arbitrarily assigning and hard wiring these levels would unnecessarily constrain the use of the VSBC20 for any given application. By using the Signetics PLHS501, the entire logic to implement the interrupt structure fits into one PLD. Furthermore, the AMAZE software to program the part is available free from Signetics. The PLHS501 can be reprogrammed until the unused feedback gates are all used. So, the user can get the software free and change the interrupt levels a couple of times before having to replace the PLHS501 with a new part. This appendix describes how the PLHS501 is used and how to change the interrupt levels.

There are a total of 17 possible interrupt sources to the processor on the VSBC20. There are up to seven possible VMEbus interrupts, nine possible local interrupts, and a Front Panel Non-Maskable interrupt. The local interrupts include: ACFAIL*, SYSFAIL*, parity error, mailbox interrupt, two Omnimodule interrupts, 24 bit timer interrupt, Parallel I/O, and Serial I/O interrupts. Although ACFAIL*, SYSFAIL* and mailbox interrupts are generated by VMEbus, they are referred to as local interrupts because they are acknowledged locally. That is, no VMEbus IACK cycle takes place. The local interrupts are latched during an IACK cycle to "freeze" the state of the interrupts. This allows the correct acknowledgement of the interrupts. The ACFAIL*, SYSFAIL* and Front Panel Non-Maskable Interrupt are assigned to Level 7. The Front Panel NMI has the highest priority followed by ACFAIL*, parity error, and SYSFAIL*. The front Panel interrupt is acknowledged by an autovector while the other three generate a vector that is encoded as described in the Error Interrupt Vector CSR. The local interrupts for the mailbox interrupt, Omnimodule Interrupt 0, Omnimodule Interrupt 1, 24 bit timer interrupt, Parallel I/O Interrupt, and Serial I/O Interrupt have been assigned by the user to the level best suited for the user's application. The mailbox interrupt uses the auto vector while the others provide interrupt acknowledge vectors. However, these may also be changed to generate autovectors. For example, if a unique Omnimodule is designed by the user and there is not enough room to provide an interrupt vector on the module, the PLD can be changed to issue an autovector instead of generating an IACK cycle to the Omnimodule. It is also possible to have two interrupts share the same level, although this seems unnecessary, since there are enough available interrupt levels.

The following examples illustrate how easy it is to change the local interrupt levels."

EXAMPLE 1:

Put the P10 interrupt in level 4 and the Omnimodule 0 interring on level 2.

```
LIRQ4 = /LIRQPIO;                                    |->
LIRQ2 = /LIRQ0OM;                                    |->  change these
IACK0OM = /(/LIRQ0M*/A3*A2*/A1*//IACK*/BAS);     <-|     equations to
IACKPIIO = /(/LIRQPIO*A3*/A2*/A1*/IACK*/BAS);    <-|
```

EXAMPLE 2:

Make both the Omnimodule Interrupt Autovectored instead of bus-vectored. (LITRQ0OM uses level 4 and LIRQIOM uses level 5.)

```
AUTOVECTOR    = [/FPNMIRQ*A3*A2*A1*/IACK*/BAS*RESET]
              + [/LIRQMBOX*A2*A2*/A1*/IACK*/BAS*RESET]
              + [/LIRQ0OM*A3*/A2*/A1*/IACK*/BAS*RESET]    <-- ADD
              + [/LIRQIOM*A3*/A2*A1*/IACK*/BAS*RESET]     <-- ADD
              + [autovector*              /BAS*RESET];

IACKIOM = /(O);  <--  change
IACK0OM = /(o);  <--  equation
```

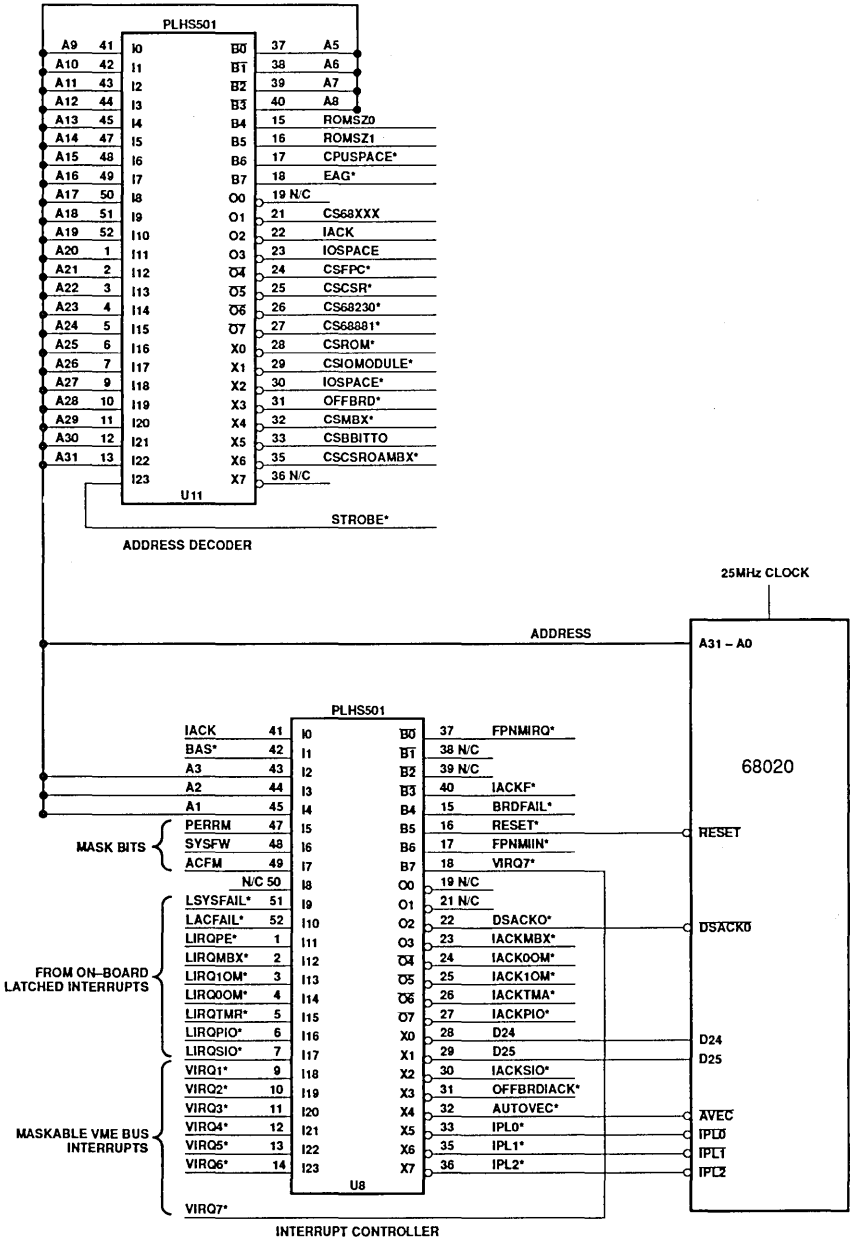


Figure 4–1. Portion of Omnibyte VSBC20 Highlighting PLHS501 Usage

```
File Name : vsbc20is
Date : 9/13/1988
Time : 9:4:2

###################### P I N   L I S T ######################

        Left                                        Right
   LABEL     ** FNC **PIN            PIN** FNC **     LABEL
VCC          ** +5V **  8-|        |-46 ** +5V **VCC
VIRQ1        ** I   **  9-|        |-45 ** I   **A1
VIRQ2        ** I   ** 10-|        |-44 ** I   **A2
VIRQ3        ** I   ** 11-|   P    |-43 ** I   **A3
VIRQ4        ** I   ** 12-|   L    |-42 ** I   **BAS
VIRQ5        ** I   ** 13-|   H    |-41 ** I   **IACK
VIRQ6        ** I   ** 14-|   S    |-40 ** /O  **IACKF
BRDFAIL      ** I   ** 15-|   5    |-39 ** /O  **N/C
RESET        ** I   ** 16-|   0    |-38 ** /O  **N/C
FPNMIIN      ** I   ** 17-|   1    |-37 ** /O  **N/C
VIRQ7        ** I   ** 18-|        |-36 ** O   **IPL2
N/C          ** O   ** 19-|        |-35 ** O   **IPL1
GND          ** 0V  ** 20-|        |-34 ** /O  **GND

        Bottom                                      Top
   LABEL     ** FNC **PIN            PIN** FNC **     LABEL
N/C          ** O   ** 21-|        |- 7 ** I   **LIRQSIO
DSACK0       ** O   ** 22-|        |- 6 ** I   **LIRQPIO
N/C          ** O   ** 23-|        |- 5 ** I   **LIRQTMR
IACK0OM      ** /O  ** 24-|   P    |- 4 ** I   **LIRQ0OM
IACK1OM      ** /O  ** 25-|   L    |- 3 ** I   **LIRQ1OM
IACKTMR      ** /O  ** 26-|   H    |- 2 ** I   **LIRQMBX
IACKPIO      ** /O  ** 27-|   S    |- 1 ** I   **LIRQPE
D24          ** O   ** 28-|   5    |-52 ** I   **LACFAIL
D25          ** O   ** 29-|   0    |-51 ** I   **LSYSFAIL
IACKSIO      ** O   ** 30-|   1    |-50 ** I   **N/C
OFFBRDIACK   ** O   ** 31-|        |-49 ** I   **ACFM
AUTOVEC      ** O   ** 32-|        |-48 ** I   **SYSFM
IPL0         ** O   ** 33-|        |-47 ** I   **PRRM
```

Figure 4–2. PLHS501 Pinlist for VSBC20 Interrupt Structure

```
@DEVICE TYPE
PLHS501
@DRAWING   1155
@REVISION  A
@DATE                    9-9-88
@SYMBOL
@COMPANY                 OMNIBYTE CORP.
@NAME                    GLENN CASE
@DESCRIPTION      VSBC20 INTERRUPT STRUCTURE PLD
@INTERNAL NODE
LIRQ7 ALLIRQ7 AHIACKF AHFPNMIRQ AUTOVECTOR FPNMIRQ

@COMMON PRODUCT TERM
LIRQ6 = /LIRQMBX;          "LIRQ6 goes high when LIRQMBX goes low"
LIRQ5 = /LIRQ1OM;
LIRQ4 = /LIRQ0OM;
LIRQ3 = /LIRQTMR;
LIRQ2 = /LIRQPIO;
LIRQ1 = /LIRQSIO;
```

Figure 4–3. VSCB20IS .BEE File (begins)

```
@I/O DIRECTION
DB4 = 0;
DB5 = 0;
DB6 = 0;
DB7 = 0;
XE0 = /IACKF;
XE1 = 1;
XE2 = 1;
XE3 = 1;
OE1 = /IACKF;
OE2 = 1;
OE3 = 1;

@I/O STEERING
@LOGIC EQUATION

LIRQ7 = [/LACFAIL * ACFM]                "LIRQ7 goes high when"
      + [/LYSFAIL * SYSFM * BRDFAIL]
      + [/LIRQPE * PERRM];

ALLIRQ7 = /[/[/LACFAIL * ACFM]
        + /[/LSYSFAIL * SYSFM * BRDFAIL]
        + /[/LIRQPE * PERRM]];

AHFPNMIRQ = [/FPNMIIN  */IACK * RESET]
          + [AHFPNMIRQ */IACK * RESET]
          + [AHFPNMIRQ *IACK *  BAS * RESET]
          + [AHFPNMIRQ *IACK * /BAS * /A1 * RESET]
          + [AHFPNMIRQ *IACK * /BAS * /A2 * RESET]
          + [AHFPNMIRQ *IACK * /BAS * /A3 * RESET]
          + [AHFPNMIRQ *IACK * /BAS * A3 * A2 * A1 * /AUTOVECTOR * RESET];

FPNMIRQ = /(AHFPNMIRQ);

IPLO: XR1= /VIRQ7
         +  LIRQ7
         + /FPNMIRQ
         + /LIRQ7 * /LIRQ6 * VIRQ7 * VIRQ6 * LIRQ5
         + /LIRQ7 * /LIRQ6 * VIRQ7 * VIRQ6 * /VIRQ5
         + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 * LIRQ3
         + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 * /VIRQ3
         + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * /LIRQ3 * /LIRQ2 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 *
VIRQ3
                  * VIRQ2 * LIRQ1
         + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * /LIRQ3 * /LIRQ2 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 *
VIRQ3
                  * VIRQ2 * /IRQ1;
XR2 = 1;

IPL1: XR1 = /VIRQ7
          +  LIRQ7
          + /FPNMIRQ
          + /LIRQ7 * VIRQ7 * LIRQ6
          + /LIRQ7 * VIRQ7 * /VIRQ6
          + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 * LIRQ3
          + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 * /VIRQ3
          + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * /LIRQ3 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 * /VIRQ3
                                          * LIRQ2
          + /LIRQ7 * /LIRQ6 * /LIRQ5 * /LIRQ4 * /LIRQ3 * VIRQ7 * VIRQ6 * VIRQ5 * VIRQ4 * /VIRQ3
                                          * /VIRQ2;
XR2 = 1;
```

Figure 4–3. VSCB20IS .BEE File (continued)

```
ILP2: XR1 = /VIRQ7
          +  LIRQ7
          + /FPNMIRQ
          + /LIRQ7 *  VIRQ7 *  LIRQ6
          + /LIRQ7 *  VIRQ7 * /VIRQ6
          + /LIRQ7 * /LIRQ6 *  VIRQ7 * VIRQ6 * LIRQ5
          + /LIRQ7 * /LIRQ6 *  VIRQ7 * VIRQ6 * /VIRQ5
          + /LIRQ7 * /LIRQ6 * /LIRQ5 * VIRQ7 * VIRQ6 * VIRQ5 * LIRQ4
          + /LIRQ7 * /LIRQ6 * /LIRQ5 * VIRQ7 * VIRQ6 * VIRQ5 * /VIRQ4;
XR2 = 1;

IACKF = /([A3 * A2 * A1 * FPNMIRQ * /LACFAIL  * IACK * /BAS]
      +   [A3 * A2 * A1 * FPNMIRQ * /LSYSFAIL * IACK * /BAS]
      +   [A3 * A2 * A1 * FPNMIRQ * /LIRQPE   * IACK * /BAS]);

OFFBRDIACK: XR1 = /LIRQ7 *  A3 *  A2 *  A1 * IACK * /BAS * FPNMIRQ * /AUTOVECTOR
                + /LIRQ6 *  A3 *  A2 * /A1 * IACK * /BAS
                + /LIRQ5 *  A3 * /A2 *  A1 * IACK * /BAS
                + /LIRQ4 *  A3 * /A2 * /A1 * IACK * /BAS
                + /LIRQ3 * /A3 *  A2 *  A1 * IACK * /BAS
                + /LIRQ2 * /A3 *  A2 * /A1 * IACK * /BAS
                + /LIRQ1 * /A3 * /A2 *  A1 * IACK * /BAS;
XR2 = 1;

AUTOVECTOR = [/FPNMIRQ * A3 * A2 *  A1 * IACK * /BAS * RESET]
           + [/LIRQMBX * A3 * A2 * /A1 * IACK * /BAS * RESET]
           + [ AUTOVECTOR                     * /BAS * RESET];

AUTOVEC; XR1 = AUTOVECTOR;
XR2 = 1;

D24: XR1 = /LACFAIL * ACFM
         + /LIRQPE  * PERRM;
XR2 = 1;

D25: XR1 = /LACFAIL * ACFM
         +  LIRQPE  *  PERRM * /LSYSFAIL * SYSFM
         +             /PERRM * /LSYSFAIL * SYSFM;
XR2 = 1;

IACK1OM = /(/LIRQ1OM *  A3 * /A2 *  A1 * IACK * /BAS);

IACK0OM = /(/LIRQ0OM *  A3 * /A2 * /A1 * IACK * /BAS);

IACKTMR = /(/LIRQTMR * /A3 *  A2 *  A1 * IACK * /BAS);

IACKPIO = /(/LIRQPIO * /A3 *  A2 * /A1 * IACK * /BAS);

IACKSIO = /(/LIRQSIO * /A3 * /A2 *  A1 * IACK * /BAS);

DSACK0 = BAS;
```

Figure 4–3. VSCB20IS .BEE File (end)

VME Bus EPROM Interface

The idea for this VMEbus EPROM board came from *WIRELESS WORLD CIRCUIT IDEAS*, January, 1988. The implementation was done by a Philips' FAE, John McNally.

The board contains two banks of EPROMs. Each bank consists of either two 27128s or two 27256s; each of which can be enabled by comparing the address location fo the board. Decoding three other address bits selects which of the banks is accessed. A 4-bit shift register combined with four jumpers provide wait states.

The circuit drawing was entered onto a PC using FutureNet DASH, a schematic capture package (Figures 4–4, 4–5, and 4–6). It was then converted to logic equations using AMAZE (Figure 4–9) and then assembled into a PLHS501.

This application, which needs eight ICs, used forty-four of the available seventy-two NAND Foldback Terms and forth of the available fifty-two pins. As the PLHS501 contains no registers, an edge-triggered D-type flip-flop was designed using NAND gates and this is used as a soft macro in order to implement the shift register function (Figure 4–6).

As suggested in the original article, the circuit could be expanded to access up to eight ROM banks (Figure 4–8). This was achieved by editing the logic equation file and adding extra equations (Figure 4–9). Modifying the drawing, although fairly easy to do, was not considered necessary as the object was to design with PML and not TTL. The expanded circuit would require another three TTL IC packages, brining the total to eleven. The number of foldback terms increased to fifty-five, with the number of pins rising to fifty. Figure 4–10 shows the pinout of both versions.

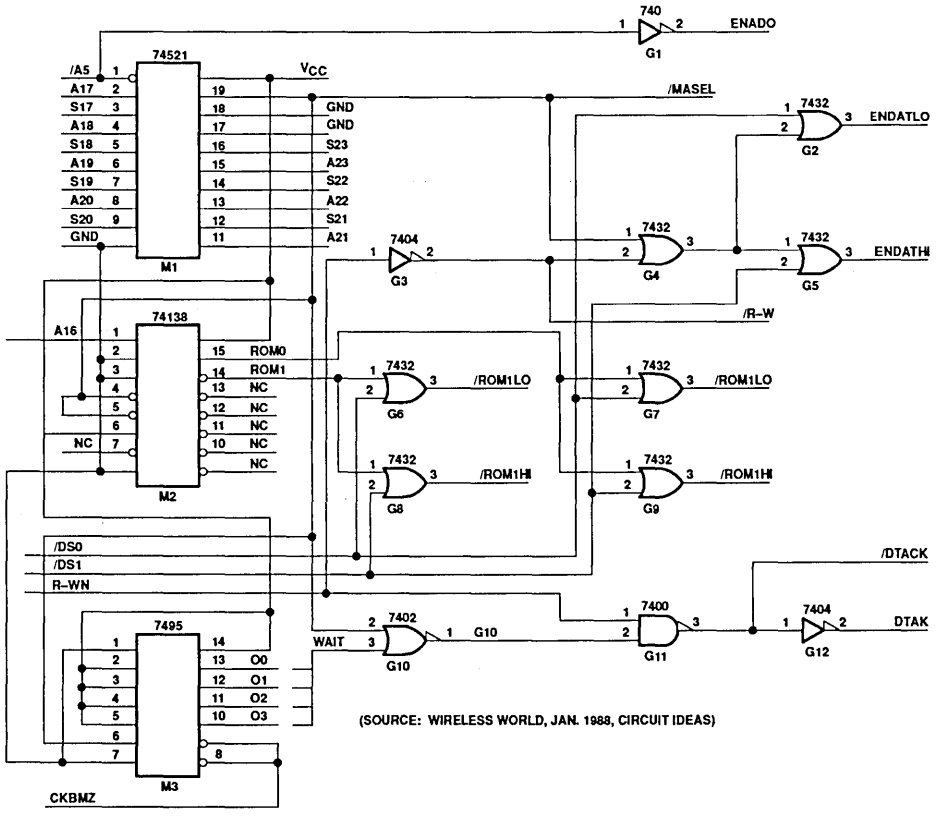


Figure 4–4. VME – EPROM Interface

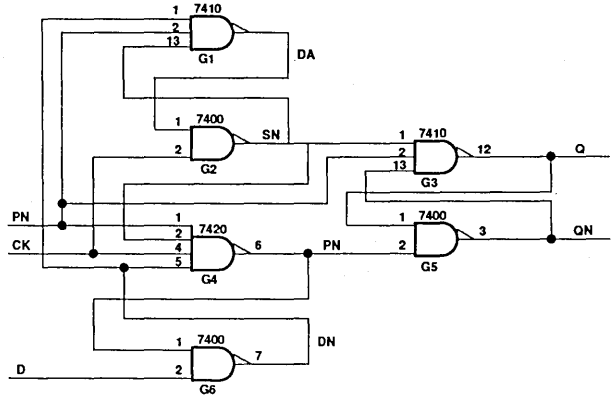


Figure 4–5. Edge-Triggered D Flip-Flop (DFFS)

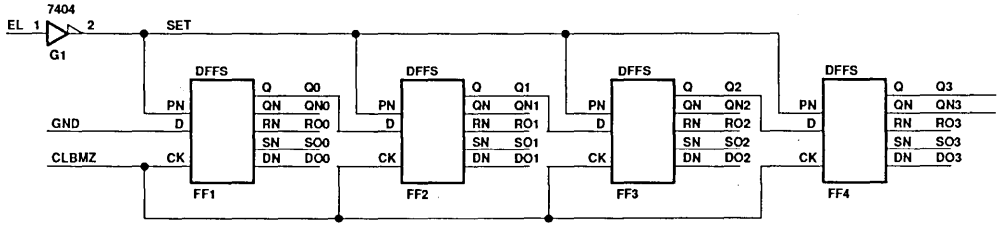
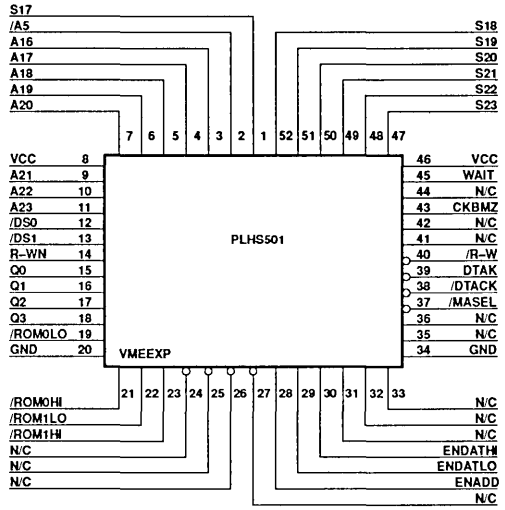


Figure 4–6. 4-Bit Shifter (7495)

2 ROM BANKS AS ORIGINAL CIRCUIT.
REPLACES 8 PACKAGES – USES 46 FOLDBACK TERMS



EXPANDED TO 8 ROM BANKS
REPLACES 11 PACKAGES – USES 55 FOLDBACK TERMS

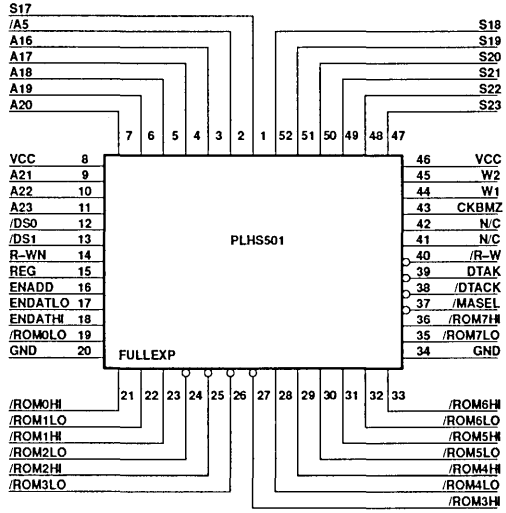


Figure 4–7. VMEEXP and FULLEXP

```
File Name : VMEEXP
Date : 2/13/1988
Time : 10:23:5

###################### P I N   L I S T ######################

        Left                                        Right
   LABEL     ** FNC **PIN           PIN** FNC **    LABEL
VCC          ** +5V **  8-|       |-46 ** +5V **VCC
A21          ** I   **  9-|       |-45 ** I   **WAIT
A22          ** I   ** 10-|       |-44 ** I   **N/C
A23          ** I   ** 11-|   P   |-43 ** I   **CKBMZ
/DS0         ** I   ** 12-|   L   |-42 ** I   **N/C
/DS1         ** I   ** 13-|   H   |-41 ** I   **N/C
R-WN         ** I   ** 14-|   S   |-40 ** /O  **/R-W
Q0           ** O   ** 15-|   5   |-39 ** /O  **DTAK
Q1           ** I   ** 16-|   0   |-38 ** /O  **/DTACK
Q2           ** I   ** 17-|   1   |-37 ** /O  **/MASEL
Q3           ** I   ** 18-|       |-36 ** 0   **N/C
/ROM0LO      ** O   ** 19-|       |-35 ** O   **N/C
GND          ** 0V  ** 20-|       |-34 ** 0V  **GND

       Bottom                                       Top
   LABEL     ** FNC **PIN           PIN** FNC **    LABEL
/ROM0HI      ** O   ** 21-|       |- 7 ** I   **A20
/ROM1LO      ** O   ** 22-|       |- 6 ** I   **A19
/ROM1HI      ** O   ** 23-|       |- 5 ** I   **A18
N/C          ** /O  ** 24-|   P   |- 4 ** I   **A17
N/C          ** /O  ** 25-|   L   |- 3 ** I   **A16
N/C          ** /O  ** 26-|   H   |- 2 ** I   **/A5
N/C          ** /O  ** 27-|   S   |- 1 ** I   **S17
ENADD        ** O   ** 28-|   5   |-52 ** I   **S18
ENDATLO      ** O   ** 29-|   0   |-51 ** I   **S19
ENDATHI      ** O   ** 30-|   1   |-50 ** I   **S20
N/C          ** O   ** 31-|       |-49 ** I   **S21
N/C          ** O   ** 32-|       |-48 ** I   **S22
N/C          ** O   ** 33-|       |-47 ** I   **S23
```

Figure 4–8. VMEEXP PLHS501 Pinlist

```
File Name: VMEEXP
Date:  2/13/1988
Time:  10:23:41

@DEVICE TYPE
     PLHS501
@DRAWING
     VMEEXP.DWG
@REVISION
@DATE
     2/12/1988
@SYMBOL
@COMPANY
@NAME
     VMEEXP
@DESCRIPTION
@INTERNAL NODE
     RO3 SO3 DO3 RO2 SO2
     DO2 Q0  RO1 SO1 DO1
     RO0 SO0 DO0
@I/O DIRECTION
     DB5 = 1 ;
     DB6 = 1 ;
     DB7 = 1 ;
     OE0 = 1 ;
     OE1 = 1 ;
@I/O STEERING
@LOGIC EQUATION
     RO3 = (/((//MASEL)*SO3*CK8MZ*DO3)) ;
     SO3 = (/(CKBMZ*(/(SO3*DO3*(//MASEL))))) ;
     DO3 = (/(Q2*RO3)) ;
     RO2 = (/((//MASEL)*SO2*CKBMZ*DO2)) ;
     SO2 = (/(CKBMZ*(/(SO2*DO2(//MASEL))))) ;
     DO2 = (/(Q1*RO2)) ;
     RO1 = (/((//MASEL)*SO1*CKBMZ*DO1)) ;
     SO1 = (/(CKBMZ*(/(SO1*DO1(//MASEL))))) ;
     DO1 = (/(Q0*RO1)) ;
     RO0 = (/((//MASEL)*SO0*CKBMZ*DO0)) ;
     SO0 = (/(CKBMZ*(/(SO0*DO0*(//MASEL))))) ;
     DO0 = (/(Q*RO0)) ;
     /ROM0LO = (/DS0+(/(/A16*/0*1*//MASEL))) ;
     /ROM0HI = (/DS1+(/(/A16*/0*1*//MASEL))) ;
     /ROM1LO = (/DS0+(/(A16*/0*1*//MASEL))) ;
     /ROM1HI = (/DS1+(/(A16*/0*1*//MASEL))) ;
     Q0 = (/((/(RO0*Q0))*SO0*(//MASEL))) ;
     Q1 = (/((/(RO1*Q1))*SO1*(//MASEL))) ;
     Q2 = (/((/(RO2*Q2))*SO2*(//MASEL))) ;
     Q3 = (/((/(RO3*Q3))*SO3*(//MASEL))) ;
     /MASEL = /(/(/[/((A17*S17+/A17*/S17)*(A18*S18+/A18*/S18)*(A19*S19+
                    /A19*/S19)*(A20*S20+/A20*/S20)*(A21*S21+/S21*/S21)*
                    (A22*S22+/A22*/S22)*(A23*S23+/A23*/S23)*//A5)])) ;
     /DTACK = /((/(/MASEL+WAIT))*R-WN) ;
      DTACK = /(/DTACK) ;
     /R-W = /(R-WN) ;
     ENADD = (//A5) ;
     ENDATLO = ((/R-W+/MASEL)+/DS0) ;
     ENDATHI = (/DS1+(/R-W+/MASEL)) ;
```

Figure 4–9. VMEEXP PLHS501 .BEE File

```
File Name : FULLEXP
Date : 2/13/1988
Time : 10:11:28

###################### P I N   L I S T ######################

        Left                                        Right
   LABEL     ** FNC **PIN          PIN** FNC **   LABEL
VCC          ** +5V **  8-|       |-46 ** +5V **VCC
A21          ** I   **  9-|       |-45 ** I   **W0
A22          ** I   ** 10-|       |-44 ** I   **W1
A23          ** I   ** 11-|   P   |-43 ** I   **CKBMZ
/DS0         ** I   ** 12-|   L   |-42 ** I   **N/C
/DS1         ** I   ** 13-|   H   |-41 ** I   **N/C
R-WN         ** I   ** 14-|   S   |-40 ** /O  **/R-W
REG          ** O   ** 15-|   5   |-39 ** /O  **DTAK
ENADD        ** O   ** 16-|   0   |-38 ** /O  **/DTACK
ENDATLO      ** O   ** 17-|   1   |-37 ** /O  **/MASEL
ENDATHI      ** O   ** 18-|       |-36 ** O   **ROM7HI
/ROM0LO      ** O   ** 19-|       |-35 ** O   **ROM7LO
GND          ** 0V  ** 20-|       |-34 ** 0V  **GND

        Bottom                                      Top
   LABEL     ** FNC **PIN          PIN** FNC **   LABEL
/ROM0HI      ** O   ** 21-|       |- 7 ** I   **A20
/ROM1LO      ** O   ** 22-|       |- 6 ** I   **A19
/ROM1HI      ** O   ** 23-|       |- 5 ** I   **A18
/ROM1LO      ** /O  ** 24-|   P   |- 4 ** I   **A17
/ROM1HI      ** /O  ** 25-|   L   |- 3 ** I   **A16
/ROM3LO      ** /O  ** 26-|   H   |- 2 ** I   **/A5
/ROM3HI      ** /O  ** 27-|   S   |- 1 ** I   **S17
/ROM4LO      ** O   ** 28-|   5   |-52 ** I   **S18
/ROM4HI      ** O   ** 29-|   0   |-51 ** I   **S19
/ROM5LO      ** O   ** 30-|   1   |-50 ** I   **S20
/ROM5HI      ** O   ** 31-|       |-49 ** I   **S21
/ROM6LO      ** O   ** 32-|       |-48 ** I   **S22
/ROM6HI      ** O   ** 33-|       |-47 ** I   **S23
```

Figure 4–10. FULLEXP Pinlist

```
File Name: FULLEXP
Date:  2/13/1988
Time:  10:11:30

@DEVICE TYPE
     PLHS501
@DRAWING
     VMEEXP.DWG
@REVISION
@DATE
     2/12/1988
@SYMBOL
@COMPANY
@NAME
     VMEEXP
@DESCRIPTION
@INTERNAL NODE
     RO3 SO3 DO3 RO2 SO2
     DO2 RO1 SO1 DO1 RO0
     SO0 DO0
     Q0 Q1 Q2 Q3
@I/O DIRECTION
     DB4 = 1 ;
     DB5 = 1 ;
     DB6 = 1 ;
     DB7 = 1 ;
     OE0 = 1 ;
     OE1 = 1 ;
     OE2 = 1 ;
     OE3 = 1 ;
     XE0 = 1 ;
     XE1 = 1 ;
     XE2 = 1 ;
     XE3 = 1 ;
@STEERING
     S0  = Q ;
     S1  = Q ;
     S2  = Q ;
     S3  = Q ;
```

Figure 4–11. FULLEXP PLHS501 .BEE File

```
@LOGIC EQUATION
     RO3 = (/(((//MASEL)*SO3*CKBMZ*DO3)) ;
     SO3 = (/(CKBMZ*(/(SO3*DO3*(//MASEL))))) ;
     DO3 = (/(Q2*RO3)) ;
     RO2 = (/(((//MASEL)*SO2*CKBMZ*DO2)) ;
     SO2 = (/(CKBMX*(/SO2*DO2*(//MASEL))))) ;
     DO2 = (/(Q1*RO2)) ;
     RO1 = (/(((//MASEL)*SO1*CKBMZ*DO1)) ;
     SO1 = (/(CKBMZ*(/(SO1*DO1*(//MASEL))))) ;
     DO1 = (/(Q0*RO1)) ;
     RO0 = (/(((//MASEL)*SO0*CKBMZ*DO0)) ;
     SO0 = (/(CKBMZ*(/SO0*DO0*(//MASEL))))) ;
     DO0 = (/(Q*RO0)) ;
     /ROM0LO = (/DS0+/(/A16*/A17*/A18*//MASEL)) ;
     /ROM0HI = (/DS1+/(/A16*/A17*/A18*//MASEL)) ;
     /ROM1LO = (/DS0+/(A16+/A17*/A18*//MASEL)) ;
     /ROM1HI = (/DS1+/(A16*/A17*/A18*//MASEL)) ;
     /ROM2LO = /(/(/DS0+/(/A16*A17*/A18*//MASEL))) ;
     /ROM2HI = /(/(/DS1+/(/A16*A17*/A18*//MASEL))) ;
     /ROM3LO = /(/(/DS0+/(A16*A17*/A18*//MASEL))) ;
     /ROM3HI = /(/(/DS1*/(A16*A17*/A18*//MASEL))) ;
     /ROM4LO = (/DS0+/(A16*/A17*A18*//MASEL)) ;
     /ROM4HI = (/DS1+/(A16*/A17*/A18*//MASEL)) ;
     /ROM5LO = (/SA0+/(A16*/A17(A18*//MASEL)) ;
     /ROM5HI = (/DS1+/(A16*/A17*A18*//MASEL)) ;
     /ROM6LO = (/DS0+/(A16*A17*A18*//MASEL)) ;
     /ROM6HI = (/DS1+/(A16*A17*A18*//MASEL)) ;
     /ROM7LO = (/DS0+/(A16*A17*A18*//MASEL)) ;
     /ROM7HI = (/DS1+/(A16*A17*A18*//MASEL)) ;
      ENADD  = (//A5) ;
     ENDATLO = ((/R-W+/MASEL)+/DS0) ;
     ENDATHI = (/DS1+(/R-W+/MASEL)) ;
     Q0 = /(((/(RO0*Q0))*SO0*(//MASEL)) ;
     Q1 = /(((/(RO1*Q1))*SO1*(//MASEL)) ;
     Q2 = /(((/(RO2*Q2))*SO2*(//MASEL)) ;
     Q3 = /(((/(RO3*Q3))*SO3*(//MASEL)) ;
     /MASEL = /(/([/((A17*S17+/A17*/A17*/S17*(A18*S18+/A18*/S18)
                  *(A19*S19+/A19*/S19)*(A20*S20+/A20*/S20)*(A21*S21
                  +/A21*/S21)*(A22*S22+/A22*/S22)*(A23**S23
                  +(A23*S23)*//A5)])) ;
     /DTACK = /((/(/MASEL+/(((Q0*W0*/W1)+(/Q1*W0*/W1)+(/Q2*/W0*W1)
                  +(/Q3*W0*W1))))*R-WN) ;
      DTAK  = /(DTACK) ;
     /R-W = /(R-WN) ;
     REG = Q0*Q1*Q2*Q3 ;
```

Figure 4–11. FULLEXP PLHS501 .BEE File (Continued)

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

MICRO CHANNEL INTERFACE

IBM's new Micro Channel Architecture (MCA) bus implements new features not found on the XT/AT bus. One new requirement for adapter designers is that of Programmable Option Select (POS) circuitry. It allows system software to configure each adapter card upon power on, thereby eliminating option select switches or jumpers on the main logic board and on adapter cards.

Each adapter card slot has its own unique –CDSETUP signal routed to it. This allows the CPU to interrogate each card individually upon power up. By activating a card's –CDSETUP line along with appropriate address and control lines two unique 8 bit ID numbers are first read from the adapter. Based upon the ID number, the system then writes into the card's option latches configuration information that has been stored in the system's CMOS RAM. The CPU also activates POS latch address 102h bit 0, which is designated as a card enable bit.

If a new card is added to the system, an auto-configuration utility will be invoked. Each adapter card has associated with it a standardized Adapter Description File with filename of @XXXX.ADF, where XXXX is the hex ID number of the card. The configuration utility prompts the user according to the text provided in the .ADF file and updates the card's latches and the system's CMOS RAM.

IBM reserves 8 addresses for byte-wide POS latches, however, depending on the card's function, not all addresses need to be used. In addition, of those addresses that are used, only the bits used need to be latched. The first two addresses which are reserved for reading the ID bytes, and bit 0 of the third address, which is defined as a card enable bit, are mandatory. Some of the remaining bits of the third address are suggested by IBM to be used as inputs to an I/O or memory address comparator to provide for alternate card addresses. Many adapter cards will not use more than these three POS locations.

The following example describes an implementation of POS circuitry realized in a PLHS501. It uses only 56 of the possible 72 internal foldback NAND gates and only a portion of the device pins, allowing additional circuitry to ba added. Figure 5–1 shows a block diagram of the circuit, and Figures 5–3 and 5–4 are the AMAZE files. Pins labeled D0O–D7O must be connected externally to pins D0I–D7I. They also must be connected through a 74F245 transceiver to the Micro Channel. External transceiver direction and enable control is provided for by circuitry within the PLHS501. The external transceiver may also be used by other devices on the adapter card.

In this application, edge-triggered registers are not required and therefore should not be used, as transparent latches use fewer NAND gates to implement. Figure 5–2 shows the various latch circuits described by the AMAZE equations. POS byte 2 was made using four of the /B device pins and four of the B pins. Notice however, from Figure 5–2(B) that the bits on the /B pins used the complement of the input pin, thereby implementing a non-inverting latch. Also, all 8 bits of this byte were brought to output pins. If some of the bits are not used by external circuitry, then the specific bit latch may not be needed or may be constructed entirely from foldback NAND gates freeing additional pins.

An external F521 may be added to provide for I/O address decoding. As the MCA bus requires all 16 bits of the I/O address to be decoded, 8 bits may be assigned to the F521 and 8 bits to the 501. Bit fields decoded in the 501 may be done so in conjunction with bits from POS byte 2 to provide for alternate I/O addressing. Additionally, some of the available 501 outputs may be used as device enables for other devices on the card.

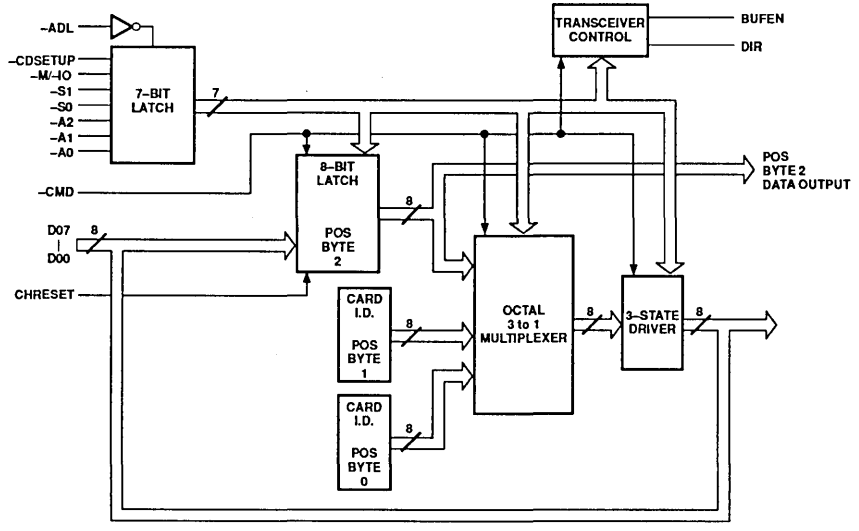
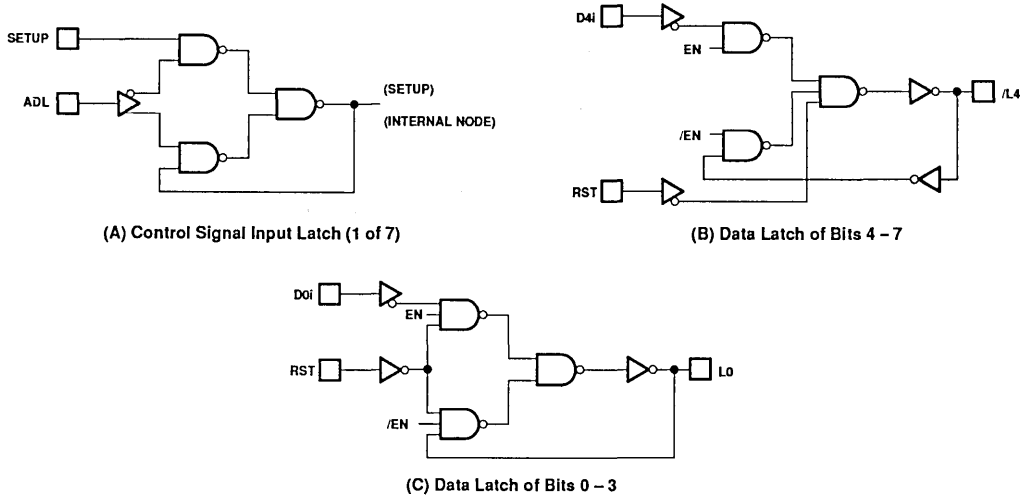


Figure 5–1. Block Diagram of Basic POS Implementation in PLHS501



(A) Control Signal Input Latch (1 of 7)

(B) Data Latch of Bits 4 – 7

(C) Data Latch of Bits 0 – 3

Figure 5–2. Latches Used in MCA Interface

```
File Name : MCPOSREG
Date : 5/31/1988
Time : 11:50:2

###################### P I N   L I S T ######################

        Left                                          Right
   LABEL     ** FNC **PIN           PIN** FNC **     LABEL
VCC          ** +5V **  8-|       |-46 ** +5V **VCC
N/C          ** I   **  9-|       |-45 ** I   **D4I
N/C          ** I   ** 10-|       |-44 ** I   **D3I
N/C          ** I   ** 11-|   P   |-43 ** I   **D2I
N/C          ** I   ** 12-|   L   |-42 ** I   **D1I
N/C          ** I   ** 13-|   H   |-41 ** I   **D0I
N/C          ** I   ** 14-|   S   |-40 ** /O  **L3
/L4          ** O   ** 15-|   5   |-39 ** /O  **L2
/L5          ** O   ** 16-|   0   |-38 ** /O  **L1
/L6          ** O   ** 17-|   1   |-37 ** /O  **L0
/L7          ** O   ** 18-|       |-36 ** O   **D7O
N/C          ** O   ** 19-|       |-35 ** O   **D6O
GND          ** 0V  ** 20-|       |-34 ** /O  **GND

        Bottom                                        Top
   LABEL     ** FNC **PIN           PIN** FNC **     LABEL
BUFEN        ** O   ** 21-|       |- 7 ** I   **SS1
N/C          ** O   ** 22-|       |- 6 ** I   **SS0
N/C          ** O   ** 23-|       |- 5 ** I   **SETUP
IOWB         ** /O  ** 24-|   P   |- 4 ** I   **CMD
N/C          ** /O  ** 25-|   L   |- 3 ** I   **A2
N/C          ** /O  ** 26-|   H   |- 2 ** I   **A1
N/C          ** /O  ** 27-|   S   |- 1 ** I   **A0
D0O          ** O   ** 28-|   5   |-52 ** I   **MIO
D1O          ** O   ** 29-|   0   |-51 ** I   **ADL
D2O          ** O   ** 30-|   1   |-50 ** I   **RST
D3O          ** O   ** 31-|       |-49 ** I   **D7I
D4O          ** O   ** 32-|       |-48 ** I   **D6I
D5O          ** O   ** 33-|       |-47 ** I   **D5I
```

Figure 5–3. PLHS501 MCPOSREG Pinlist

```
File Name: MCPOSREG
Date:  5/31/1988
Time:  11:50:17

@DEVICE TYPE
     PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
     Basic Programmable Option Select circuitry
         for a Micro Channel Adaptor card

@INTERNAL NODE
     /setup1,/mio1,a01,a11,/a21,ss01,ss11;
     /en,outen,/iow;
@COMMON PRODUCT TERM
     read0 = (setup1*/ss11*ss01*mio1*/cmd*a21*/a11*/a01);
     read1 = (setup1*/ss11*ss01*mio1*/cmd*a21*/a11* a01);
     read2 = (setup1*/ss11*ss10*mio1*/cmd*a21* a11*/a01);
"
     NOTE: In the above equations, setup1, mio1 and a21 all should be
     preceded by a slash (/). The slash was omitted to correct for
     a mapping error in AMAZE 1.65 when using active low internal node
     definitions in common product terms.
"

     b7h = 0;  " Define high ID byte "
     b6h = 1;  "   (POS byte #1)     "
     b5h = 1;  "      7E hex         "
     b4h = 1;
     b3h = 1;
     b2h = 1;
     b1h = 1;
     b0h = 0;

     b7l = 1;  " Define low ID byte "
     b6l = 1;  "    (POS byte #0)   "
     b5l = 1;  "       FF hex       "
     b4l = 1;
     b3l = 1;
     b2l = 1;
     b1l = 1;
     b0l = 1;
@I/O DIRECTION
       "3-state output control of d7o-d0o"
     xe0 = (/setup1*/ss11*ss01*/mio1*/cmd*a21*outen);
     xe1 = (/setup1*/ss11*ss01*/mio1*/cmd*a21*outen);
     xe2 = (/setup1*/ss11*ss01*/mio1*/cmd*a21*outen);
     xe3 = (/setup1*/ss11*ss01*/mio1*/cmd*a21*outen);
@I/O STEERING
```

Figure 5–4. PLHS501 MCPOSREG .BEE File (begins)

```
@LOGIC EQUATION

      " 7-Bit Input Latch for Control Signals "

    /setup1 = /setup*/ad1 + /setup1*ad1;
    /mio1   = /mio */ad1  + /mio1  *ad1;
    ss11    = ss1  */ad1  + ss11   *ad1;
    ss01    = ss0  */ad1  + ss01   *ad1;
    /a21    = /a2  */ad1  + /a21   *ad1;
    a11     = a1   */ad1  + a11    *ad1;
    a01     = a0   */ad1  + a01    *ad1;

      " Option Select Octal Data Latch (POS byte #2) "
      " 10 is to be used as a card enable signal"

    /en = /[/setup1*/ss01*ss11*/mio1*/cmd*/a21*a11*/a01]; "write to latch"

    /17 = /[/d7i * en]  * /[17 * /en] * [/rst];
    /16 = /[/d6i * en]  * /[16 * /en] * [/rst];
    /15 = /[/d5i * en]  * /[15 * /en] * [/rst];
    /14 = /[/d4i * en]  * /[14 * /en] * [/rst];
    13 = /(/[ d3i * en * /rst] * /[13 * /en  * /rst]);
    12 = /(/[ d2i * en * /rst] * /[12 * /en  * /rst]);
    11 = /(/[ d1i * en * /rst] * /[11 * /en  * /rst]);
    10 = /(/[ d0i * en * /rst] * /[10 * /en  * /rst]);

      " Octal 3 to 1 Multiplexer "
      " This miltiplexer selects between reading
        POS[0], POS[1] or POS[2] onto the data bus"

    d7o = (b7h*read1 + b7l*read0 + /17*read2);
    d6o = (b6h*read1 + b6l*read0 + /16*read2);
    d5o = (b5h*read1 + b5l*read0 + /15*read2);
    d4o = (b4h*read1 + b4l*read0 + /14*read2);
    d3o = (b3h*read1 + b3l*read0 +  13*read2);
    d2o = (b2h*read1 + b2l*read0 +  12*read2);
    d1o = (b1h*read1 + b1l*read0 +  11*read2);
    d0o = (b0h*read1 + b0l*read0 +  10*read2);
      "3-State output control for d7o-d0o:

    outen =/[a11*a01];

      "External F245 transceiver control"

    iowb  = /(/a21 * /setup1 * mio1 * ss11 * /ss01);
    /iow  = /(/a21 * /setup1 * mio1 * ss11 * /ss01);
    bufen = cmd * /iow;
```

Figure 5–4. PLHS501 MCPOSREG .BEE File (end)

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

NuBus INTERFACE

In Apple Computer's book* "*Designing Cards and Drivers for Macintosh II and Macintosh SE*", an application was described for interfacing an 8-bit I/O controller to the NuBus. The controller used was a SCSI controller of the type used on the main Macintosh logic board. Seven devices (three of which were PAL architecture) were used as control circuitry interfacing the SCSI controller and two RAM chips to the bus.

This example of using the PLHS501 shows a method of interfacing the same SCSI controller and RAM chips to the NuBus using only three parts. The adapter card schematic is shown in Figure 6–2 and the AMAZE listing is in Figure 6–6. Although the AMAZE listing may seem confusing at first glance, the circuitry fused into the PLHS501 can be broken down into small blocks of latches, flip-flops, and schematically in Figures 6–4 and 6–5. Circuit timing is shown in Figure 6–3.

Referring to Figure 6–4 and Figure 6–5, the circuitry starts a transaction by first detecting a valid address in either the slot or super slot range. The detection is accomplished by two wide-input NAND gates, and controlled by the /CLK signal. Following each NAND gate is an S-R latch to hold the signal until near the end of the cycle. The two S-R latch signals are combined into one signal named ST0 such that if either NAND gate output was low, then some delay time after the rising edge of /CLK, ST0 will go low. The next rising edge of /CLK will cause signal ST1 to go low. This sets signal DE2 low, which is an input to an external flip-flop to cause ST2 to go low at the next rising /CLK edge terminating the cycle. An external flip-flop was necessary to achieve a high-speed /CLK to /IOR and /ACK transition. Also, an external FI25 buffer was added to meed the soon to be approved IEEE P1196 specification requirement of 60mA I_{OL} for signal /NMRQ and 24mA I_{OL} for signals /TM0/TM1 and /ACK. Figure 6–5(B) shows an easily implemented latch which controls interrupts generated by the SCSI controller passing onto the bus. Upon /RESET the latch is put into a known state. Under software control, by writing to a decoded address, the latch may be set or reset, thereby gating or blocking the interrupt signals.

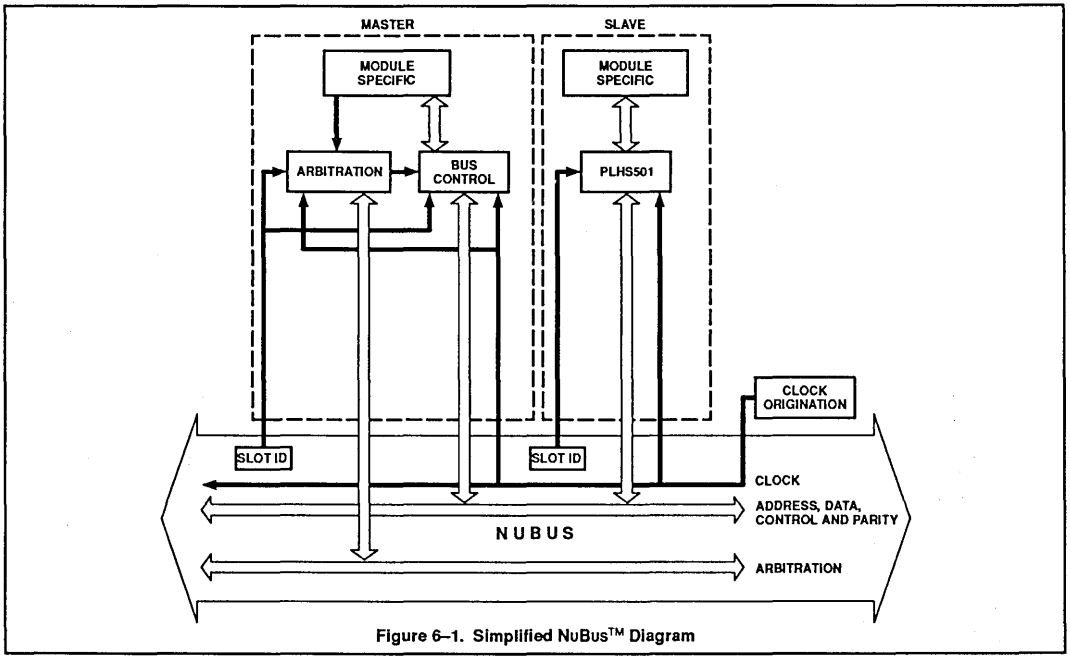


Figure 6–1. Simplified NuBus™ Diagram

* *Designing Cards and Drivers for Macintosh II and Macintosh SE*, Addison–Wesley Publishing Company, Inc. 1987.

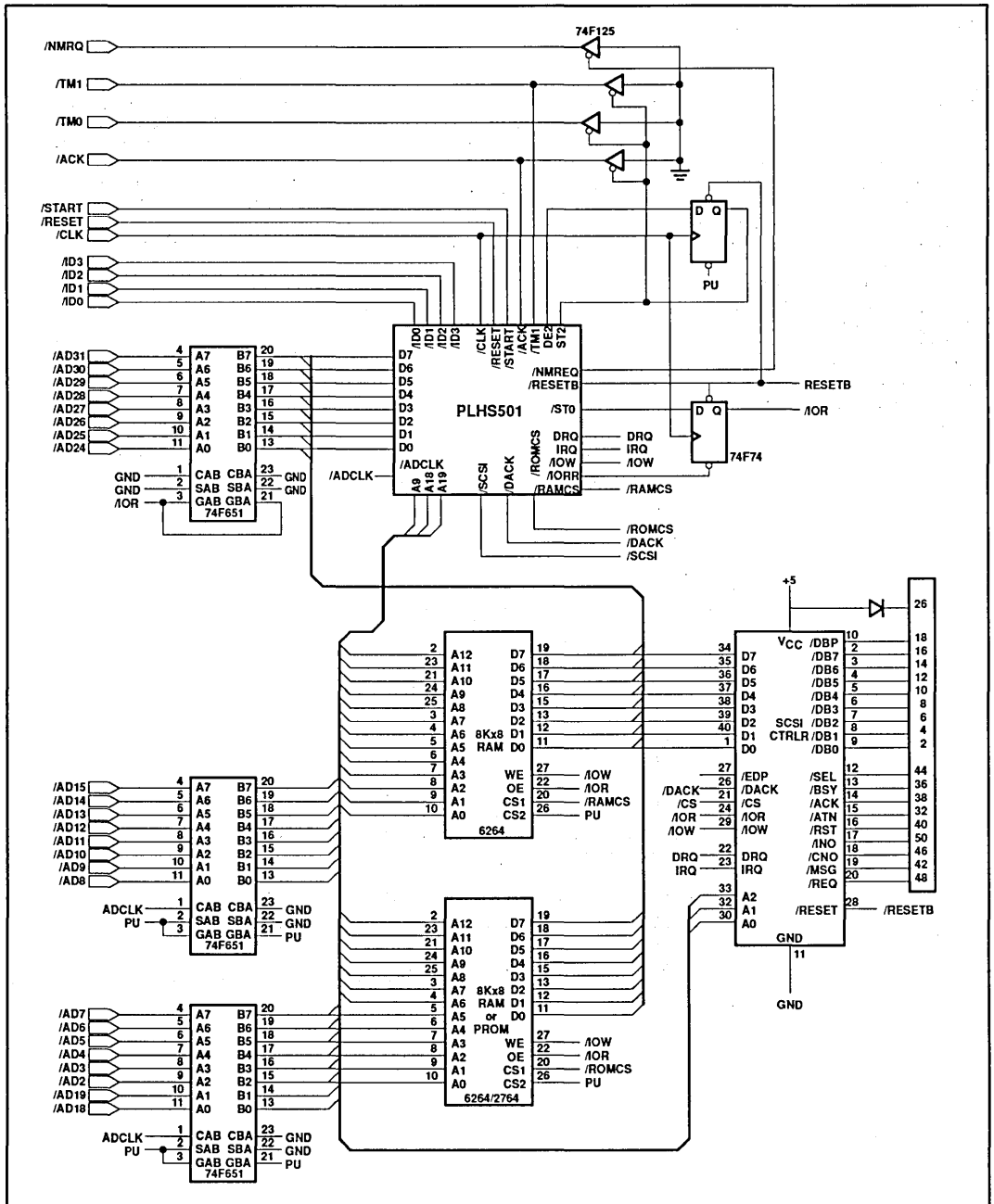


Figure 6–2. Adapter Card Schematic

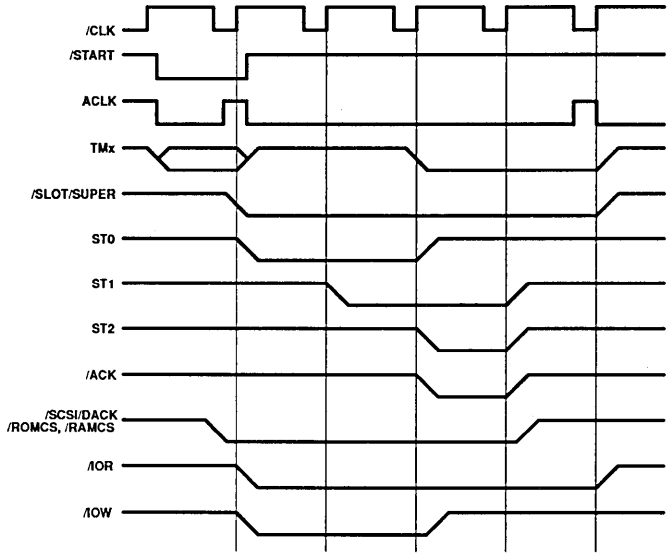


Figure 6–3. Timing Diagram

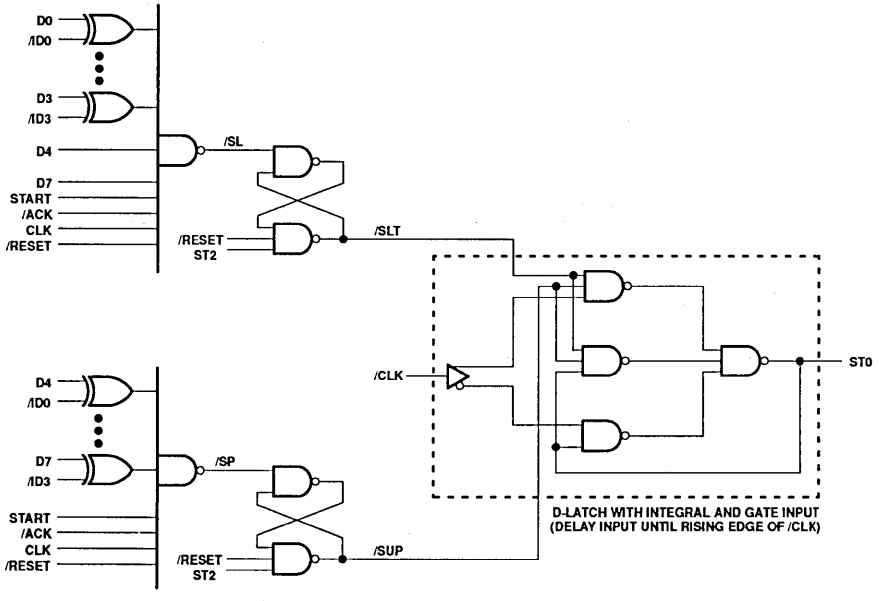
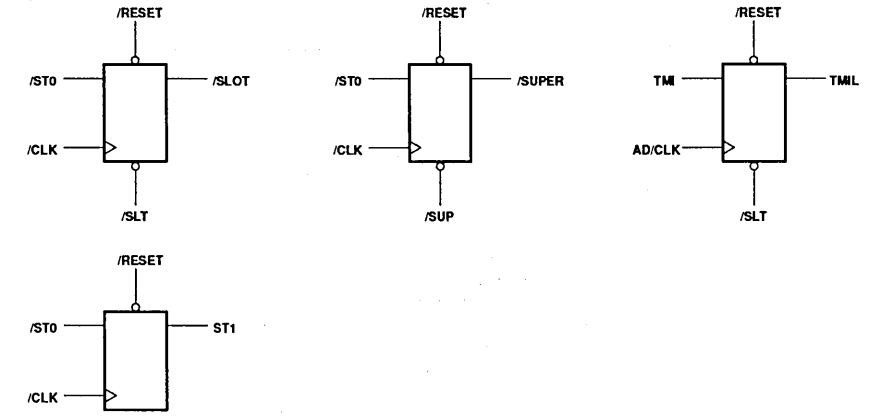
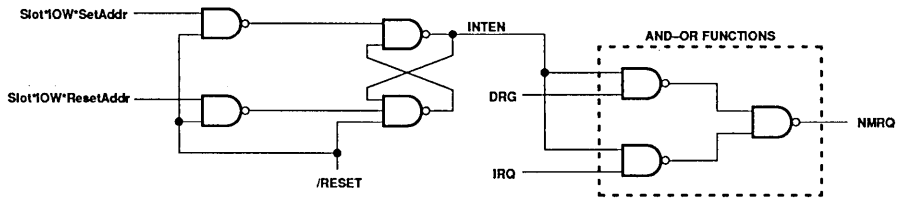


Figure 6–4. Decoding and Latch Circuitry



(A) Four Internal Flip-Flops Constructed from NAND Gates.



(B) Interrupt Enable Control Latch
Internal Flip-Flops and Latches

Figure 6–5. Internal Flip-Flops and Latches

```
                    ###################### P I N   L I S T ######################

                            Left                                        Right
                       LABEL     ** FNC **PIN           PIN** FNC **    LABEL
                    VCC          ** +5V **  8-|       |-46 ** +5V **VCC
                    /ID2         ** I   **  9-|       |-45 ** I   **D1
                    /ID3         ** I   ** 10-|       |-44 ** I   **D0
                    DRQ          ** I   ** 11-|   P   |-43 ** I   **A19
                    IRQ          ** I   ** 12-|   L   |-42 ** I   **A18
                    ST2          ** I   ** 13-|   H   |-41 ** I   **A9
                    N/C          ** I   ** 14-|   S   |-40 ** /O  **ST0
                    N/C          ** O   ** 15-|   5   |-39 ** /O  **N/C
                    N/C          ** O   ** 16-|   0   |-38 ** /O  **N/C
                    N/C          ** O   ** 17-|   1   |-37 ** /O  **N/C
                    N/C          ** O   ** 18-|       |-36 ** O   **N/C
                    N/C          ** O   ** 19-|       |-35 ** O   **N/C
                    GND          ** 0V  ** 20-|       |-34 ** /O  **GND

                            Bottom                                      Top
                       LABEL     ** FNC **PIN           PIN** FNC **    LABEL
                    N/C          ** O   ** 21-|       |- 7 ** I   **/ID1
                    N/C          ** O   ** 22-|       |- 6 ** I   **/ID0
                    ACLK         ** O   ** 23-|       |- 5 ** I   **/RESET
                    /ROMCS       ** /O  ** 24-|   P   |- 4 ** I   **/TM1
                    /RAMCS       ** /O  ** 25-|   L   |- 3 ** I   **/ACK
                    N/C          ** /O  ** 26-|   H   |- 2 ** I   **/START
                    /NMRQ        ** /O  ** 27-|   S   |- 1 ** I   **/CLK
                    DE2          ** O   ** 28-|   5   |-52 ** I   **D7
                    /RESET       ** O   ** 29-|   0   |-51 ** I   **D6
                    /SCSI        ** O   ** 30-|   1   |-50 ** I   **D5
                    /DACK        ** O   ** 31-|       |-49 ** I   **D4
                    /IORR        ** O   ** 32-|       |-48 ** I   **D3
                    /IOW         ** O   ** 33-|       |-47 ** I   **D2

@DEVICE TYPE
     PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
@NAME
@DESCRIPTION
     SCSI-NuBus Interface
@INTERNAL NODE
     /sl,/sp,/SLOT,/SUPER;
sn1,sn2,rn1,rn2;
sn3,rn3,st1;
sn4,rn4,tm11,tm11n;
CMP3a,CMP2a,CMP1a,CMP0a;
CMP3b,CMP2b,CMP1b,CMP0b;
/slt,/sup,stln,adclk;
setad,rstad,inten;
slotn,supern;
@COMMON PRODUCT TERM
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION

     "Address Decode"
cmp0a = (d0*id0+/d0*/id0);
cmp1a = (d1*id1+/d1*/id1);
```

Figure 6–6. AMAZE Listing (begins)

```
cmp2a = (d2*id2+/d2*/id2);
cmp3a = (d3*id3+/d3*/id3);
cmp0b = (d4*id0+/d4*/id0);
cmp1b = (d5*id1+/d5*/id1);
cmp2b = (d6*id2+/d6*/id2);
cmp3b = (d7*id3+/d7*/id3);

/sl = /(d7*d6*d4*cmp0a*cmp1a*cmp2a*cmp3a*start*/ack*clk);
/sp = /(cmp0b*cmp1b*cmp2b*cmp3b*start*/ack*clk);
     "latch slot signal"
/slt = /(/reset*st2*/[/sl*/slt]);
     "latch super signal"
/sup = /(/reset*st2*/[/sp*/sup]);
     "Let /slt or /sup through only
      until after the rising edge
      of /clk"
st0 = /(/[/slt*/sup*clk] * /[st0*clk] * /[/slt*/sup*st0] * /reset);
     "Slot signal D-type Flip-Flop"
sn1 = /(/clk*slt*(/[sn1*/reset*/super*(/[st0*rn1*/slt])]));
rn1 = /(/clk*sn1*(/[st0*rn1*/slt]));
/slot = /(/reset*/super*sn1*slotn);
slotn = /(//slot*rn1*/slt);
     "Super signal D-type Flip-Flop"
sn2 = /(/clk*/sup*(/[sn2*/reset*/slot*(/[st0*rn2*/sup])]));
rn2 = /(/clk*sn2*(/[st0*rn2*/sup]));
/super = /(/reset*/slot*sn2*supern);
supern = /(/super*rn2*/sup);
     "State 1 D-type Flip-Flop"
sn3 = /(/clk*(/[sn3*/reset*(/[st0*rn3])]));
rn3 = /(/clk*sn3*(/[st0*rn3]));
st1 = /(/reset*sn3*st1n);
st1n= /[st1*rn3];
     "output to external flop"
de2 = /(st1n * st2);
     "address latch clock"
adclk = clk*st0*st1;
aclk  = clk*st0*st1;
     "latch tm1 signal for r/w info"
sn4 = /(adclk*/reset*(/[sn4*(/[/tm1*rn4*/reset])]));
rn4 = /(adclk*sn4*(/[/tm1*rn4*/reset]));
tm1l = /(sn4*tm1ln);
tm1ln= /(rn4*/reset*tm1l);
     "
     tm1l  -> 1 read, 0 write
     tm1ln -> 0 read, 1 write
     "
     "straight decode stuff"
/iorr = /(/st0*tm1l                          * /reset);
/iow  = /(/tm1ln*st0                         * /reset);
/scsi = /(slotn*/a19*/a18*/a9                * /reset);
/dack = /(slotn*/a19*/a18* a9                * /reset);
/romcs= /(slotn* a19* a18                    * /reset);
/ramcs= /(supern                             * /reset);
/resetb= /reset;
     "interrupt control latch"
setad = /(tm1ln*/st0*slotn* a19*/a18* a9);
rstad = /(tm1ln*/st0*slotn* a19*/a18*/a9);
inten = /(setad*(/[inten*rstad*/reset]));
/nmrq = /(inten*drq+inten*irq);
```

Figure 6–6. AMAZE Listing (end)

PLHS501
Application Notes
Vol. 2

Programmable Logic Devices

NUGGETS

Much current focus for microprocessor design is on the address bus. Typically, most designers assume the processor will handle the data manipulation and the data bus is assumed to be a straight, clean path to and from the memory. Data transformations may be accomplished for specific purposes when the application requires it. For instance, a classic transformation from the early 70's was the bit reversal required to address operands for a Fast Fourier Transformation. When designers implemented bit reversal as a separate hardware process, the whole system improved. Likewise for hardware multipliers.

Also, a hidden "transformation" is the appending of parity and the calculation of E.C.C. polynomials. Clearly, when the designer recognizes that significant performance improvement can be achieved by realizing the payoff attainable with a special purpose hardware device, he should design it. For example, let's consider parity generation:

Data Bus Parity

The PLHS501 can span 32 bits of input data. It has four output Ex-OR gates, and the ability to generate literally any function of the inputs. It would seem that there must be some "best" way to generate and detect parity. Recall that the PLHS501 can generate both deep logic functions (lots of levels) and wide logic functions (lots of inputs). The best solution would require the fewest gates and the fewest number of logic levels. Let's review the basics, first. Table 7–1(A) shows the parity function for two variables and Table 7–1(B) shows it for three variables. The Ex-OR function generates even parity.

It is noticeable that there are precisely 50% logical 1 entries in the truth tables. This yields the famous checkerboard Karnaugh Maps. With a checkerboard K-map, no simplification of Ex-OR functions is possible by Boolean simplification. The two variable Ex-OR has two ones (implying 3 gates to generate), the 3 variable has four ones (implying 5 gates to generate). In general, $2^{n-1}+1$ product terms could generate Ex-OR functions in two levels of NAND gates (assuming complementary input variables exist). You must have an unlimited number of gate inputs for this to hold.

The PLHS501 could do this for 7 input variables in two levels ($2^6+1=65$), but cannot support 8 ($2^7+1=129$). Hence, it is appropriate to seek a cascaded solution, hopefully taking advantage of the available output Ex-OR functions. Let's solve a 16 input Ex-OR function, by subpartitioning. First, consider Figure 7–1(A) where two literals are Exclusive-ORed to generate an intermediate Ex-OR function. This requires available complementary inputs and generates even parity in two levels. Figure 7–1(B) also does this (by factoring), requiring 3 gate levels, but does not require complementary inputs.

Assuming inputs must get into the PLHS501 through the pin receivers, it is best to generate as wide of an initial Ex-OR as possible, so a structure like Figure 7–1(A) expanded is appropriate. Figure 7–1 shows a 2-level 4 input Ex-OR function which may be viewed as a building block. This structure may be repeated four times, across four sets of four input bits generating partial intermediate parity values which may then be treated through two boxes similar to Figure 7–1(B). These outputs are finally combined through an output Ex-OR at a PLHS501 output pin. Figure 7–3 shows the complete solution which requires 44 NANDs plus one Ex-OR.

Table 7–1. Even Parity Functions

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Table 7–1(A).

A	B	C	A ⊕ B
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Table 7–1(B).

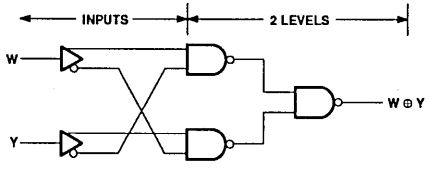


Figure 7–1(A).

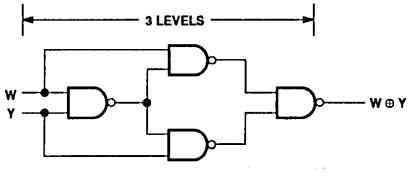


Figure 7–1(B).

Figure 7–1. Complementary Input Levels

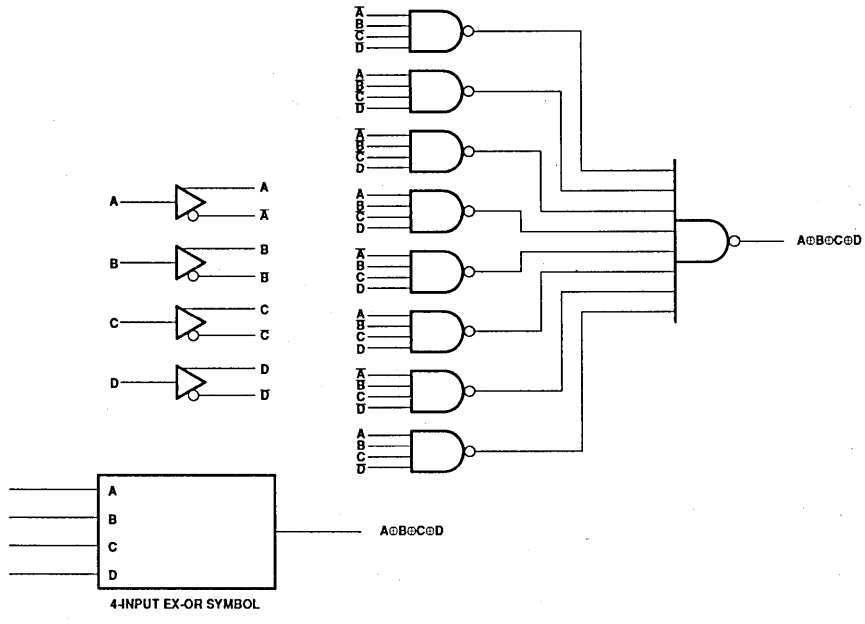


Figure 7–2. Four Variable Ex-ORs

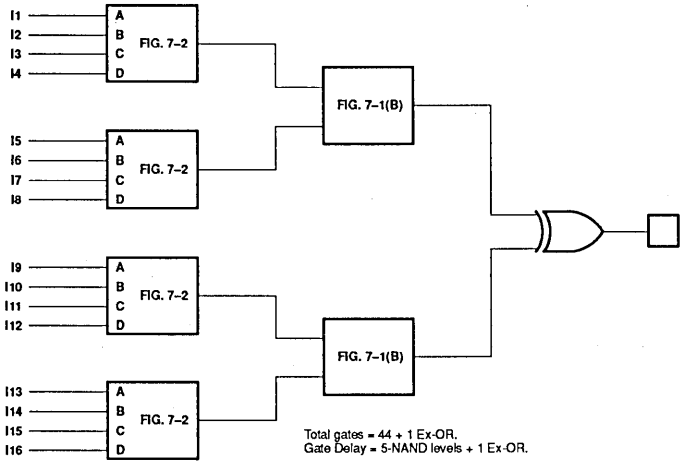


Figure 7–3. 16 Input Even Parity Generation

Two examples follow which were supplied by one of our European Sales Engineers, Nils Lindgren. The first, called "paritet", calculates even and odd parity for 24 input literals. Several output options are available and the design uses a cascade with a different partitioning than just previously discussed.

The second example "compare" implements, a 16-bit comparator over 32 input bits. The design generates outputs for conditions representing the classic "EQUAL", "AGTB" (A>B) and BGTA (B>A). The long, triangularized equation for T42 suggests that Nils found a clever editing approach to accurately enter a relatively long design equation into Signetics AMAZE.

```
File Name : PARITET
Date : 5/31/1988
Time : 10:26:22

###################### P I N   L I S T ######################

        Left                                        Right
   LABEL     ** FNC **PIN          PIN** FNC **    LABEL
VCC          ** +5V **  8-|       |-46 ** +5V **VCC
A            ** I   **  9-|       |-45 ** I   **K
B            ** I   ** 10-|       |-44 ** I   **J
C            ** I   ** 11-|   P   |-43 ** I   **I
D            ** I   ** 12-|   L   |-42 ** I   **H
E            ** I   ** 13-|   H   |-41 ** I   **G
F            ** I   ** 14-|   S   |-40 ** /O  **N/C
N/C          ** B   ** 15-|   5   |-39 ** /O  **N/C
N/C          ** B   ** 16-|   0   |-38 ** /O  **N/C
N/C          ** B   ** 17-|   1   |-37 ** I   **OEN
N/C          ** B   ** 18-|       |-36 ** O   **N/C
N/C          ** O   ** 19-|       |-35 ** O   **N/C
GND          ** 0V  ** 20-|       |-34 ** 0V  **GND

        Bottom                                      Top
   LABEL     ** FNC **PIN          PIN** FNC **    LABEL
N/C          ** O   ** 21-|       |- 7 ** I   **Y
N/C          ** O   ** 22-|       |- 6 ** I   **X
ODD_OC       ** O   ** 23-|       |- 5 ** I   **V
ODD          ** /O  ** 24-|   P   |- 4 ** I   **U
EVEN         ** /O  ** 25-|   L   |- 3 ** I   **T
EVEN_OC      ** /O  ** 26-|   H   |- 2 ** I   **S
N/C          ** /O  ** 27-|   S   |- 1 ** I   **R
N/C          ** O   ** 28-|   5   |-52 ** I   **Q
N/C          ** O   ** 29-|   0   |-51 ** I   **P
N/C          ** O   ** 30-|   1   |-50 ** I   **O
N/C          ** O   ** 31-|       |-49 ** I   **N
N/C          ** O   ** 32-|       |-48 ** I   **M
N/C          ** O   ** 33-|       |-47 ** I   **L
```

Figure 7–4. PARITET PLHS501 Pinlist

```
File Name : PARITET
@DEVICE TYPE
   PLHS501
@DRAWING
@REVISION
@DATE
   1988
@SYMBOL
@COMPANY
   Philips
@NAME
   Nils Lindgren
@DESCRIPTION
   24 bit parity circuit
@INTERNAL NODE
   J0 J1 J2 J3 J4 J5 J6 J7 J8 J9 T0 T1 T2 T3
@COMMON PRODUCT TERM
@I/O DIRECTION
   OE1=T2*T3*/OEN;
   OE2=/OEN;
   OE3=T0*T1*/OEN;
@I/O STEERING
@LOGIC EQUATION
     "FIRST LEVEL: 'EVEN' FROM GROUPS OF THREE INPUTS"
   J0=/A*/B*/C + /A*B*C + A*/B*C + A*B*/C;
   J1=/D*/E*/F + /D*E*F + D*/E*F + D*E*/F;
   J2=/G*/H*/I + /G*H*I + G*/H*I + G*H*/I;
   J3=/J*/K*/L + /J*K*L + J*/K*L + J*K*/L;
   J4=/M*/N*/O + /M*N*O + M*/N*O + M*N*/O;
   J5=/P*/Q*/R + /P*Q*R + P*/Q*R + P*Q*/R;
   J6=/S*/T*/U + /S*T*U + S*/T*U + S*T*/U;
   J7=/V*/X*/Y + /V*X*Y + V*/X*Y + V*X*/Y;
     "SECOND LEVEL: 'EVEN' FROM FOUR GROUPS AT A TIME"
   J8=/J0*/J1*/J2*/J3 + /J0*/J1*J2*J3 + J0*J1*/J1*/J3 + /J0*J1*J2*J3
      + J0*/J1*/J2*J3 + /J0*J1*/J2*J3 + J0*/J1*J2*/J3 + J0*J1*J2*J3;
   J9=/J4*/J5*/J6*/J7 + /J4*/J5*J6*J7 + J4*J5*/J6*/J7 + /J4*J5*J6*J7
      + J4*/J5*/J6*J7 + /J4*J5*/J6*J7 + J4*/J5*J6*/J7 + J4*J5*J6*J7;
   T0=/(J8*J9);
   T1=/(/J8*/J9);
   T2=/(J8*/J9);
   T3=/(/J8*J9);
   ODD=/(T2*T3);
   EVEN=/(T0*T1);
   ODD_OC=0;
   EVEN_OC=/(1);
```

Figure 7–5. PARITET PLHS501 .BEE File

```
File Name : compare
Date : 5/31/1988
Time : 10:25:29

###################### P I N   L I S T ######################

        Left                                          Right
   LABEL     ** FNC **PIN           PIN** FNC **     LABEL
VCC          ** +5V **  8-|       |-46 ** +5V **VCC
A0           ** I   **  9-|       |-45 ** I   **B2
A1           ** I   ** 10-|       |-44 ** I   **B1
A2           ** I   ** 11-|   P   |-43 ** I   **B0
A3           ** I   ** 12-|   L   |-42 ** I   **AF
A4           ** I   ** 13-|   H   |-41 ** I   **AE
A5           ** I   ** 14-|   S   |-40 ** I   **AD
A6           ** I   ** 15-|   5   |-39 ** I   **AC
A7           ** I   ** 16-|   0   |-38 ** I   **AB
A8           ** I   ** 17-|   1   |-37 ** I   **AA
A9           ** I   ** 18-|       |-36 ** O   **N/C
N/C          ** O   ** 19-|       |-35 ** O   **N/C
GND          ** 0V  ** 20-|       |-34 ** 0V  **GND

        Bottom                                        Top
   LABEL     ** FNC **PIN           PIN** FNC **     LABEL
EQUAL        ** O   ** 21-|       |- 7 ** I   **BF
AGTB         ** O   ** 22-|       |- 6 ** I   **BE
BGTA         ** O   ** 23-|       |- 5 ** I   **BD
N/C          ** /O  ** 24-|   P   |- 4 ** I   **BC
N/C          ** /O  ** 25-|   L   |- 3 ** I   **BB
N/C          ** /O  ** 26-|   H   |- 2 ** I   **BA
N/C          ** /O  ** 27-|   S   |- 1 ** I   **B9
N/C          ** O   ** 28-|   5   |-52 ** I   **B8
N/C          ** O   ** 29-|   0   |-51 ** I   **B7
N/C          ** O   ** 30-|   1   |-50 ** I   **B6
N/C          ** O   ** 31-|       |-49 ** I   **B5
N/C          ** O   ** 32-|       |-48 ** I   **B4
N/C          ** O   ** 33-|       |-47 ** I   **B3
```

Figure 7–6. PLHS501 Pinlist for 16–Bit Comparator

```
File Name : compare
Date : 5/31/1988
Time : 10:25:43

@DEVICE TYPE
   PLHS501
@DRAWING
@REVISION
@DATE
@SYMBOL
@COMPANY
   PHILIPS
@NAME
   NILS LINDGREN
@DESCRIPTION
   16 BIT COMPARATOR WITH THREE OUTPUTS:
     EQUAL,AGTB (A>B), AND BGTA (B>A)
@INTERNAL NODE
   T1  T2  T3  T4  T5  T6  T7  T8
   T9  T10 T11 T12 T13 T14 T15 T16
   T17 T18 T19 T20 T21 T22 T23 T24
   T25 T26 T27 T28 T29 T30 T31 T32
   T41 T42
@COMMON PRODUCT TERM
@I/O DIRECTION
@I/O STEERING
@LOGIC EQUATION
   T1=/(AF*/BF);    T2=/(/AF*BF);
   T3=/(AE*/BE);    T4=/(/AE*BE);
   T5=/(AD*/BD);    T6=/(/AD*BD);
   T7=/(AC*/BC);    T8=/(/AC*BC);
   T9=/(AB*/BB);    T10=/(/AB*BB);
   T11=/(AA*/BA);   T12=/(/AA*BA);
   T13=/(A9*/B9);   T14=/(/A9*B9);
   T15=/(A8*/B8);   T16=/(/A8*B8);
   T17=/(A7*/B7);   T18=/(/A7*B7);
   T19=/(A6*/B6);   T20=/(/A6*B6);
   T21=/(A5*/B5);   T22=/(/A5*B5);
   T23=/(A4*/B4);   T24=/(/A4*B4);
   T25=/(A3*/B3);   T26=/(/A3*B3);
   T27=/(A2*/B2);   T28=/(/A2*B2);
   T29=/(A1*/B2);   T30=/(/A1*B1);
   T31=/(A0*/B0);   T32=/(/A0*B0);

   T41=T1*T2*T3*T4*T5*T6*T7*T8*T9*T10*T11*T12*T13*T14*T15*T16*T17*
       T18*T19*T20*T21*T22*T23*T24*T25*T26*T27*T28*T29*T30*T31*T32;
   T42=                                                          /T1+
                                                               /T3*T2+
                                                            /T5*T4*T2+
                                                         /T7*T6*T4*T2+
                                                      /T9*T8*T6*T4*T2+
                                                   /T11*T10*T8*T6*T4*T2+
                                               /T13*T12*T10*T8*T6*T4*T2+
                                           /T15*T14*T12*T10*T8*T6*T4*T2+
                                       /T17*T16*T14*T12*T10*T8*T6*T4*T2+
                                   /T19*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                               /T21*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                           /T23*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                       /T25*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                   /T27*T26*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
               /T29*T28*T26*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
           /T31*T30*T28*T26*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2;

EQUAL=T41;
AGTB=T42;
BGTA=/(T41+T42);
```

Figure 7–7. Compare PLHS501 .BEE File

Data Bus Operations

The following is basically an academic example, posed for the sake of illustration. Suppose some special data bus operations are desirable. For the purpose of illustration, let's label the microprocessor bus output side as ODAT0–ODAT15 and the output of our PLHS501 as D0–D15. Basically, the microprocessor will output straight data and the PLHS501 will alter it according to some plan.

We will replicate multiple identical cells, but they need not be identical in practice. Table 7–2 shows the operations to be done (just about any could be chosen, provided they meet the gate budget).

Table 7–2. Data Operations

I_2	I_1	D_{OUT}
0	0	ODATI (pass)
0	1	ODATI (complement)
1	0	SWITCH
1	1	DOUBLE SHIFT

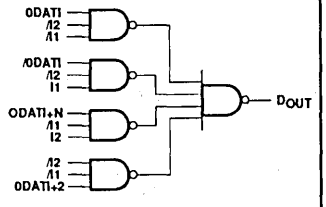


Figure 7–8. Basic Cell Structure

It may be observed that in one mode, the data passes directly, it complements in another, switches bits in another and rotates right in the last. Four input gates per bit are required to map the bits, and one output gate. Clearly, the straight PLHS501 NAND outputs can be judiciously used, but care must be taken when using other output functions. A 16-bit data bus requires 16 cell configuration where each cell is essentially identical to Figure 7–8, but its internal structure may be altered to account for the particular output pins logic function.

PLHS502
Application Notes
Vol. 1

Programmable Logic Devices

INTRODUCTION

Certain design techniques are used repeatedly by nearly all digital systems designers. If these useful building blocks occur with enough volume production, they become special purpose contenders for silicon manufacturers to justify rendering as standard products. Some building blocks, however, are never viewed as likely candidates because the performance requirements may be too high, the volumes not high enough, or it never occurred to marketeers that these subsystems would be valuable. System designers could fashion solutions to these building blocks from glue logic or PLDs and sometimes small gate arrays. Several typical building blocks will be illustrated here — including a 4-byte datapipe, a small content addressable memory (CAM), a system resource scoreboard and a synchronous receiver/transmittor. The generation of each building block will be demonstrated with a Signetics PLHS502 (Figure 1–1). This device is rendered in the Programmable Macro Logic (PML) architecture which deftly bridges the gap between gate arrays and ordinary PLDs.

Designed with a high-speed bipolar process, the PLHS502 combines 64 extremely wide foldback NAND gates with two types of internal flip-flops. The PLHS502 provides 8 internal D flip-flops and 8 internal S-R flip-flops. Each flip-flop can toggle in excess of 50MHz. These flip-flops are called "hard macro" flip-flops. Unique among such programmable devices, each flip-flop has completely independent clocking. This allows either external clocking (from four different pins) or internally derived clocking events. Ripple and synchronous controllers may be freely mixed. It should be remembered that additional flip-flops and specially custom designed flip-flops may be configured from the NAND array. Clock independence is a requirement for generating distinct internal sequencers and controllers. Additionally, it should be noted that the flip-flop Q outputs cross the clock fusing array, but $\overline{Q}$ outputs do not. The PLHS502 is packaged in a 68-pin PLCC.

This application note consists of several sections. The next section briefly describes Signetics SNAP software package for implementing PML designs. Section 3 describes the basic process for estimating whether a design will fit into a PLHS502. In Section 4, some guidelines for designing synchronous state machines are given with focus on efficient counter and shifter design. Specific examples are included which may be easily mimiced for successful state machine design. Additional guidelines are then provided in Section 5, for optimizing a design before it is implemented with SNAP. These guidelines will help guarantee that SNAP implements the function precisely as needed. In Section 6, some larger examples are provided which illustrate some interesting and unique capabilities of the PLHS502. Section 7 details a procedure for merging logic functions into flip-flops for faster, more efficient structures.

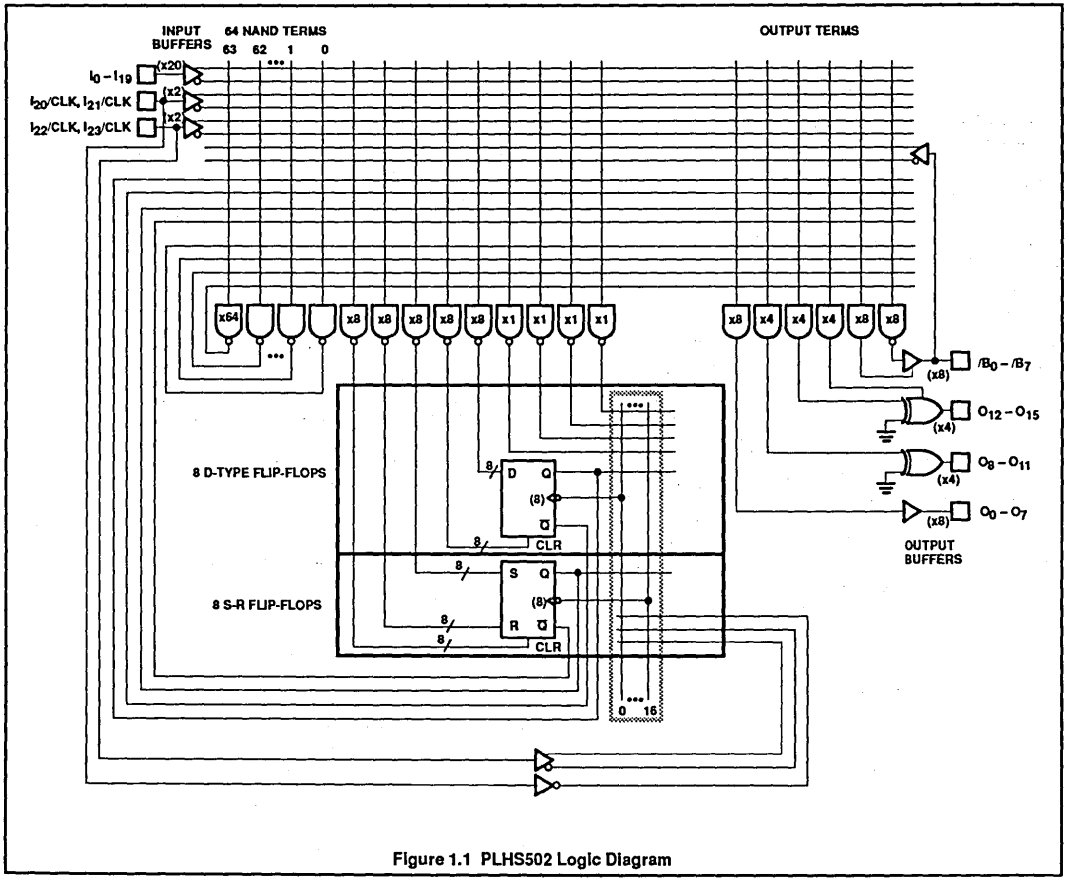


Figure 1.1 PLHS502 Logic Diagram

DEVELOPMENT SUPPORT

Because the architecture encourages deep functional nesting, a new support tool has been developed. Synthesis, Netlist, Analysis and Program (SNAP) software defines a gate array type development environment. SNAP permits several forms of design capture (schematic, Boolean equations, state equations, etc.), a gate array simulator with back annotation, waveform display and a complete fault analyzer and final fusemap compilation and model extraction. SNAP comes with a library of cells, and designs may be captured independently of the ultimate device that will implement the design. This permits the designer to migrate his design among a family of PML devices just as gate array designs can be moved to larger foundations when they do not route on smaller ones. Figure 2–1 shows the SNAP user interface "Shell" which dictates one sequence of operations to complete a design. Other sequences may be used.

The top portion of the shell depicts the paths available for design entry. Any design may be implemented in any one or a blend of all methods. For instance, a shift register might best be described schematically but a decoder by logic equations. These may be united with a multiplexor described by a text netlist as well. Ultimately, each form of input will be transformed to a function netlist and passed either to the simulation section or to the compiler section. Waveform entry is for simulation stimuli.

The simulator portion of SNAP is a 5-State gate array simulator with full timing information, setup and hold time checking, toggle and fault grade analysis and the ability to display in a wide range of formats, any set of nodes within the design. This permits a designer to zoom in with a synthetic logic state analyzer and view the behavior of any point in the design. Simulations can occur with unit delays, estimations or exact delays. The sequence of operations depicted in Figure 2–1 is entirely arbitrary, as many other paths exist.

It should be noted that the output of the "merger" block represents the composite design, but as yet is not associated to a PML device. This occurs in the compiler portion wherein association to the device occurs and a fusemap is compiled. This is analogous to placement and routing in a gate array environment. Because of the inter-connectibility of PML, this is not difficult. Once compiled, the exact assignment of pins, gates and flip-flops is known, so timing parameters may be associated and a new simulation model generated with exact detailed timing embedded. The design may be simulated very accurately at this point, and if correct, a part should be programmed.

To facilitate future migration to workstations, SNAP has been written largely in C. The internal design representation is EDIF (Electronic Design Interchange Format) compatible which permits straightforward porting to many commercially viable environments. SNAP currently utilizes OrCAD for schematic entry with eminent availability of FutureNet™ DASH.

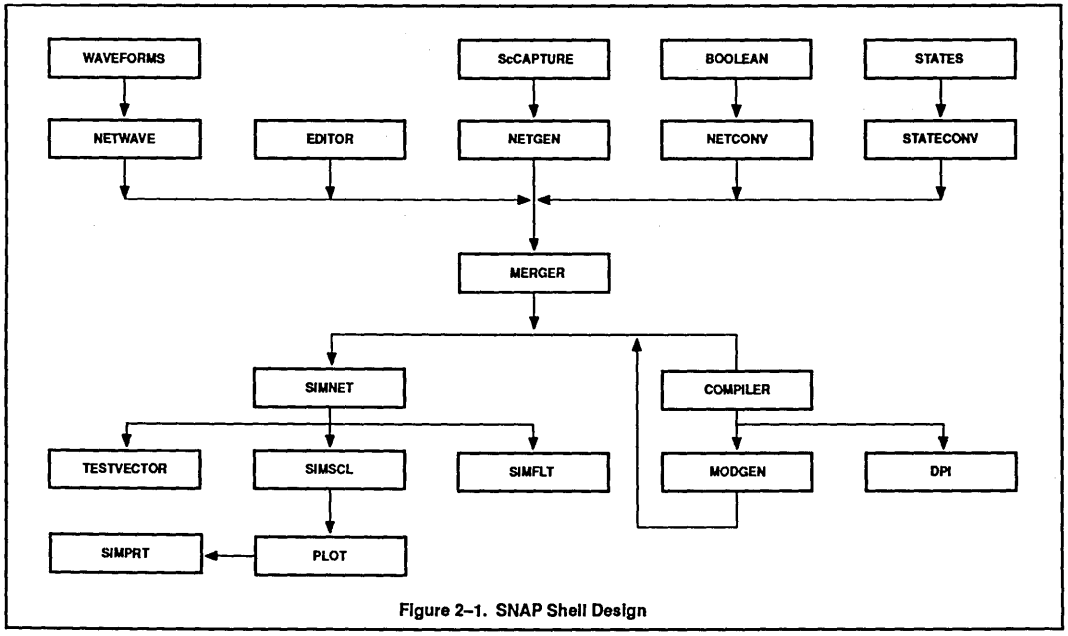


Figure 2–1. SNAP Shell Design

FutureNet and DASH are trademarks of DATA I/O

PLHS502
Application Notes
Vol. 1

Programmable Logic Devices

CAPACITY AND PARTITIONING CONSIDERATIONS

One of the dominant attributes of PML architecture is its complete interconnectibility. Any function – NAND, flip-flop, input and output structures, can be connected to any other. PML devices do not exhibit the restricted interconnect bottleneck like other programmable gate arrays. If there is capacity within the part for a function, it can be connected without the sad surprise ending of "nonroutability". Estimation for design fit is simply a matter of tallying function usage against a fixed set of resources using a table lookup. An elementary table of typical useful functions is provided in Table 3–1. Clearly Table 3–1 shows only a few of the typical functions achievable and their relative "expense" from the total function budget. As with gate arrays, the designer needs only to implement the portion of his chosen function that is to be actually used.

Fortunately, if the designer is using SNAP, all unused functions will be automatically eliminated. This is done by netlist analysis where SNAP observes an output within your circuit which is unconnected. It eliminates the unconnected gate and reanalyzes to see if there are more unconnected gates in the design. The procedure iterates until there are no more unconnected gates. When estimating whether a function will fit or not, the values in Table 3–1, if used without modification, should result in a high gate count. So, to more accurately assess fit, they should be derated to account for automatic netlist trimming. As well, the estimator should consider logic functions which can be obtained for "free" from input buffers and output functions. Guidelines, provided in Section 5, will illustrate this process.

Table 3–1. PLHS502 Gate Equivalent Table

FUNCTION	INTERNAL NAND EQUVALENT	COMMENTS
Gates		
NANDs	1	For 1 to 32-pin input variables
ANDs	1	Additional internal inputs can be used as needed
NORs	1	Additional internal inputs can be used as needed
ORs	1	Additional internal inputs can be used as needed
Macro Flip-Flops		
D-Type Flip-Flop	N/A	Total budget = 8
SR-Type Flip-Flop	N/A	Total budget = 8
Gate Implemented Flip-Flops		
D-Type Flip-Flop	6	With asynchronous S-R
T-Type Flip-Flop	6	With asynchronous S-R
J-K-Type Flip-Flop	10	With asynchronous S-R
Transparent-D Latch	4	With asynchronous S-R
S-R Latch	2	With asynchronous S-R
Decoders		
3-to-8	8	Inverted inputs available
4-to-16	16	Inverted inputs available
5-to-32	32	Inverted inputs available (24 chip outputs only)
Encoders		
8-to-3	15	Inverted inputs, 2 logic levels
16-to-4	32	Inverted inputs, 2 logic levels
32-to-5	41	Inverted inputs, 2 logic levels
Multiplexers		
4-to-1	5	Inverted inputs available
8-to-1	9	Inverted inputs available
16-to-1	17	Inverted inputs available
27-to-1	28	Can address only 27 external inputs - more if internal only. This disallows clock inputs to flip-flop.

PLHS502 Rough Resource Budget = 64 NANDs, 8 D, 8 SR, 24 inputs, 16 outputs, 8 bidirectionals.

PLHS502
Application Notes
Vol. 1

Programmable Logic Devices

STATE MACHINE DESIGN

Synchronous state machines can be classified in roughly three practical categories — sequence generators, sequence detectors and controllers. These can also be subcategorized as Mealy, Moore, finite state, linear, etc. A very large application market is covered, by considering the basic design of counters and shifters because a counter (with possible decoding) can be viewed as a generalized sequence generator and a shifter (with decoding) can be viewed as a sequence detector. A couple of small examples should illustrate the basic principles of flip-flop selection, picking optimal solutions and trading off hard macro and soft macro functions. First, a few small counters will be discussed in detail from a logic viewpoint. Then, small shifters will be shown. These basic designs are extendible so that a designer can recognize the patterns to make the counters and shifters bigger if necessary. In the next section, some general guidelines for getting "smaller, tighter" designs will be given.

Counter Design Notes

A straightforward 3-bit up/down counter transition table is depicted in Figure 4–1(a). The state variables are designated A, B, C and the direction control is U. Up counting occurs when U = 1 and down counting occurs when U = 0. Figure 4–1(b) shows four variable maps with the next state transitions at the current state and input intersections. These transitions will be useful in reference to Figures 4–2, 4–3 and 4–4 where the design is cast onto D, S-R and J-K flip-flop solutions.

We will not initially be interested in the full design details but rather, only on the number of product terms and sum terms for each solution. Figure 4–2(b) shows the K-map loops for a D flip-flop solution. Figure 4–2(a) shows the required transitions used to generate the three maps in Figure 4–2(b). There are 9 loops for the A and B variables requiring 9 product terms and 2 sum terms for driving the A and B flip-flops. The C flip-flop requires no additional logic. Note the "SUM" terms are free on PLHS502 D flip-flops because of the embedded NANDs.

P.S.			N.S. U = 0			N.S. U = 1		
A	B	C	A	B	C	A	B	C
0	0	0	1	1	1	0	0	1
0	0	1	0	0	0	0	1	0
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	1
1	0	1	1	0	0	1	1	0
1	1	0	1	0	1	1	1	1
1	1	1	1	1	0	0	0	0

COUNTER STATE TABLE

(a)

A TRANSITIONS

CU \ AB	00	01	11	10
00	0→1	0→0	1→1	1→0
01	0→0	0→0	1→1	1→1
11	0→0	0→1	1→0	1→1
10	0→0	0→0	1→1	1→1

B TRANSITIONS

CU \ AB	00	01	11	10
00	0→1	1→0	1→0	0→1
01	0→0	1→1	1→1	0→0
11	0→1	1→0	1→0	0→1
10	0→0	1→1	1→1	0→0

C TRANSITIONS

CU \ AB	00	01	11	10
00	0→1	0→1	0→1	0→1
01	0→1	0→1	0→1	0→1
11	1→0	1→0	1→0	1→0
10	1→0	1→0	1→0	1→0

(b)

Figure 4–1. 3–Bit U/D Counter Definition

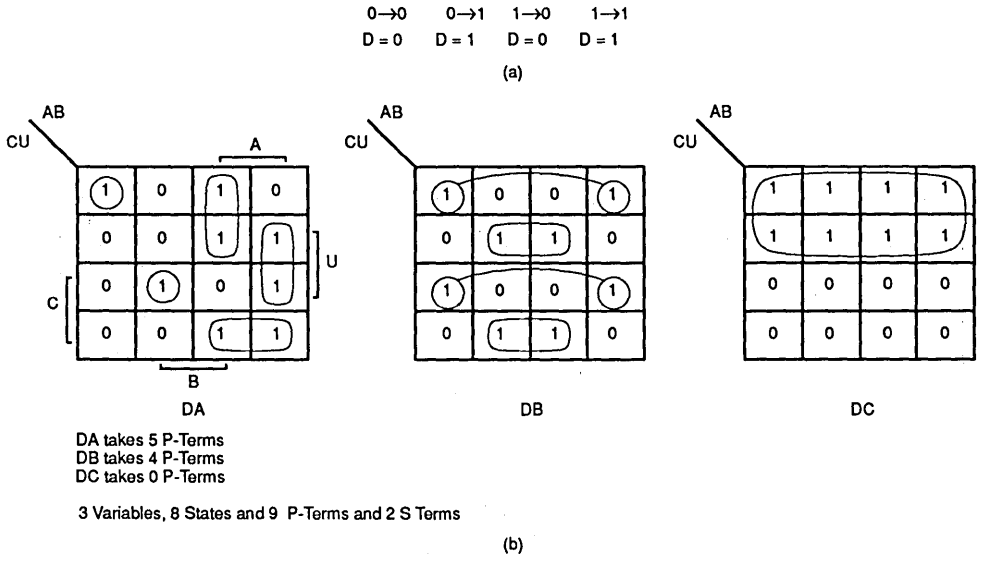


Figure 4–2. K-Map Loops Using D-FFs

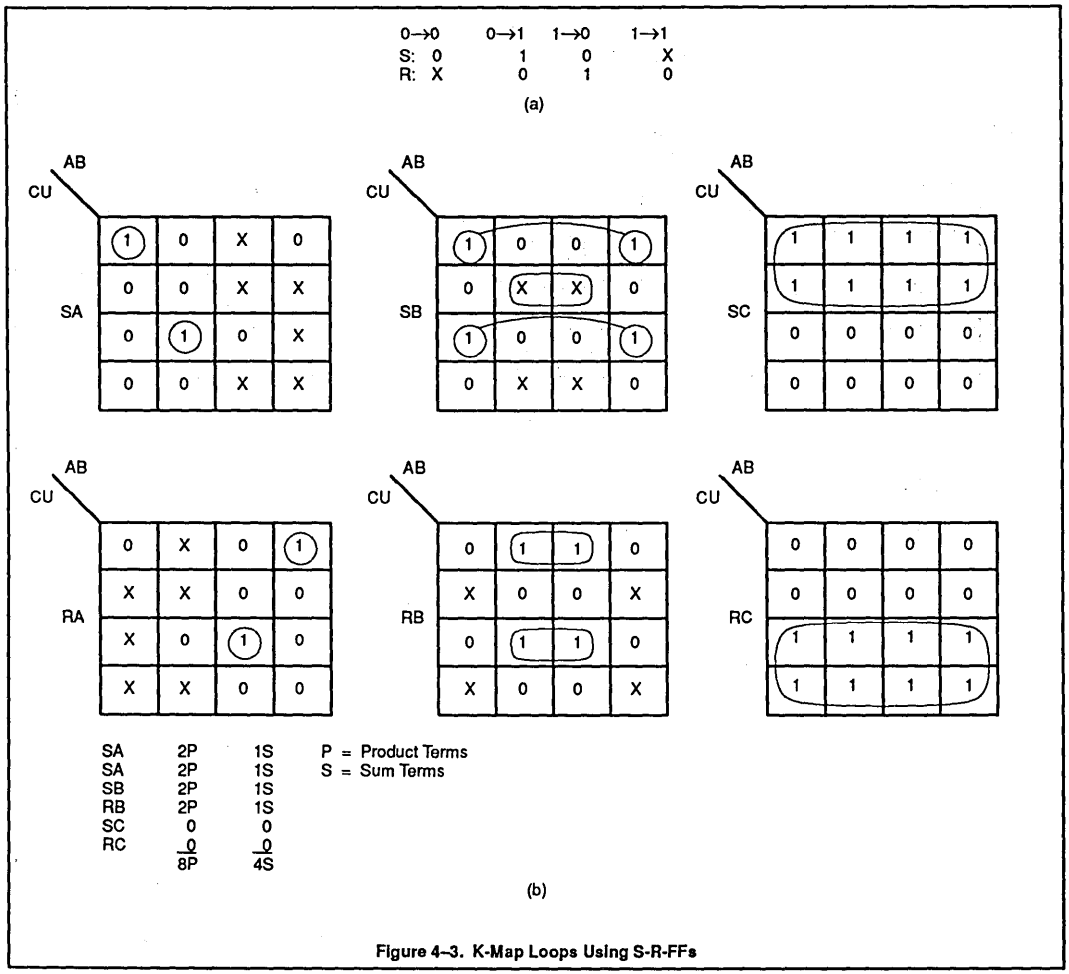


Figure 4–3. K-Map Loops Using S-R-FFs

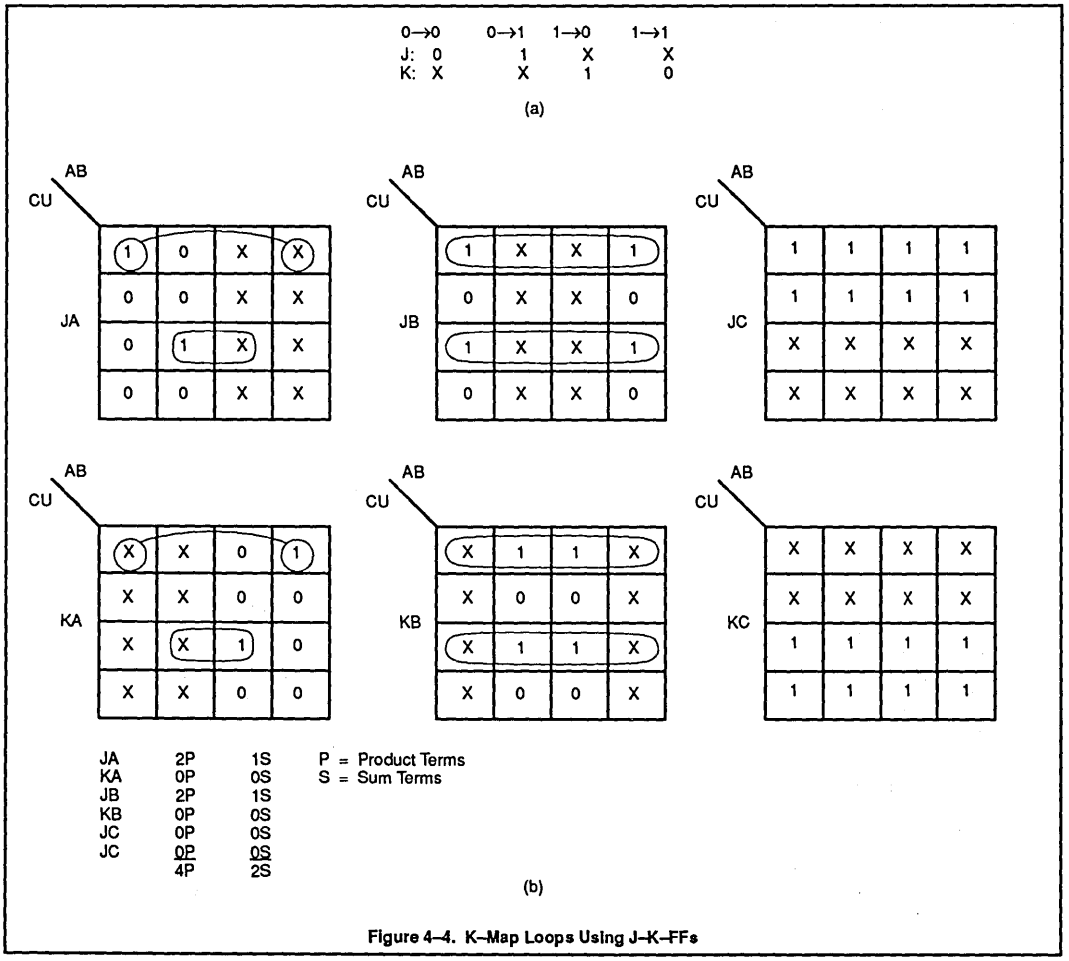


Figure 4–4. K–Map Loops Using J–K–FFs

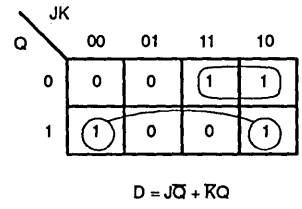
Figure 4–3 shows the same design implemented with S-R flip-flops. Figure 4–3 (a) shows the required S-R transitions and Figure 4–3(b) maps them onto the corresponding state variable maps. Tallying the loops, we find a total of 8 product terms and 4 sum terms. Again, the sum terms are free.

Figure 4–4 shows the design again on J-K flip-flops. Figure 4–4(a) shows the transitions and Figure 4–4(b) the K-map loops. Again tallying yields 4 product terms and 4 sum terms. The C variable is realized by J = K = 1 using no product terms, or sum terms.

From standard logic design we know that D flip-flops will increase product terms (no Don't Care transitions), S-R flip-flops are less dramatic and J-Ks increase product terms the least (i.e., maximum Don't Cares).

However, the PLHS502 has no J-K flip-flops. For simple toggling, the D flip-flop requires no additional circuitry and is the smallest (usually fastest) implementation. So, Ds should always be assigned for the least significant bit.

J	K	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



$D = J\overline{Q} + \overline{K}Q$

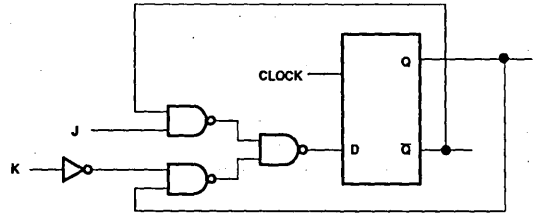


Figure 4–5. J-K Flip-Flop Derived from a D

If required, J-K flip-flops can be constructed from D flip-flops by utilizing the structure shown in Figure 4–5, as a substitution. The sum term is taken from the NAND physically connected to the D flip-flop and the K input may be derivable from either an input inverter or a $\overline{Q}$ from a flip-flop. There will be a penalty for using this structure from a speed point, but it may save gates if used judiciously.

This example assumes the designer is implementing the counter with the internal hard macro flip-flops. If the design is being generated from the NAND array only, it should be noted that the payoff will be interestingly different. The D flip-flop requires 6 NANDs and the J-K flip-flop will require 10 NANDs. In this version, the sum terms cost an additional NAND gate each. So, the 3-bit up/down counter will require:

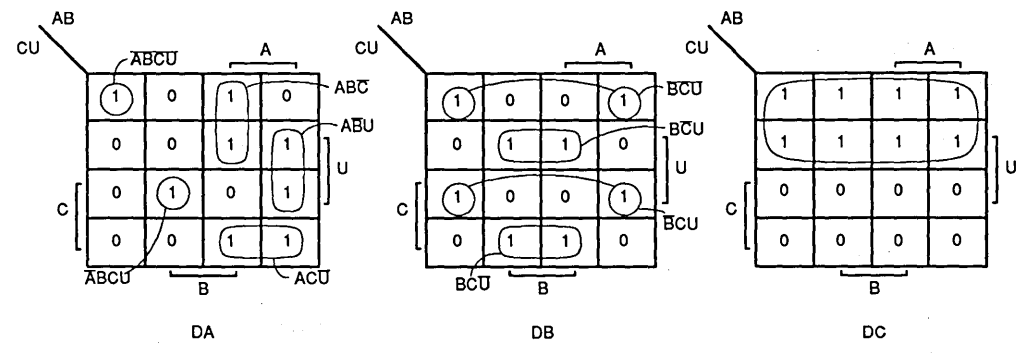
Flip-flops :	3 x 6 =	18	
Sum terms	:	2	
Prod. terms	:	9	
	Total =	29	gates

The same design built from J-Ks configured from NANDs will require:

Flip-flops :	3 x 10 =	30	
Sum terms	:	3	
Prod. terms	:	4	
	Total =	37	gates

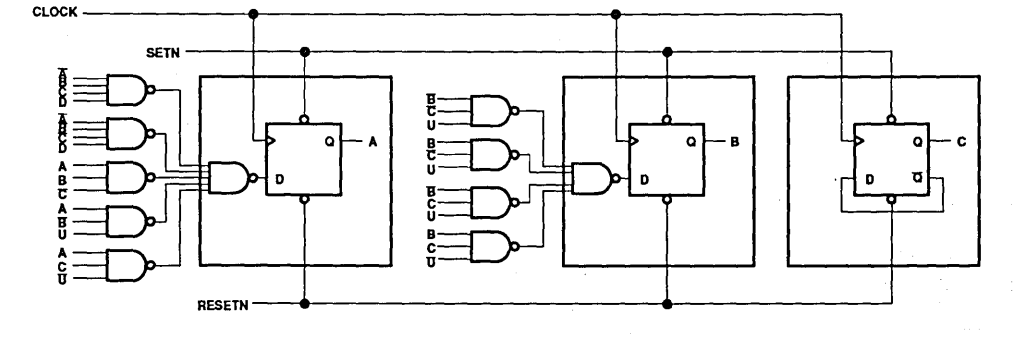
The all D version looks pretty good at this point from a total gate count view. Let's take a closer look at the D flip-flop solution:

Figure 4–6 shows the D flip-flop solution with all prime implicants looped and the corresponding transition equations are below. This is implemented in Figure 4–7 using conventional D flip-flops in the PLHS502. An alternate solution would be to substitute the 6 NAND D flip-flop for each D-box in Figure 4–7, but a better (only 25 gates!) solution can be achieved by merging the logic gates on the input of the flip-flops right into the NAND flip-flop structure as shown in Figure 4–8. This technique was described in PLHS501 Applications Notes Volume 2.



DA = $\overline{A}BCU + \overline{A}B\overline{C}U + AB\overline{C} + ABU + AC\overline{U}$
DB = $\overline{B}CU + B\overline{C}U + \overline{B}CU + BC\overline{U}$
DC = $\overline{C}$

Figure 4–6. D Flip-Flop Equations for 3-Bit Counter



TOTAL COST = 3 D-FFS
+ 9 GATES

Figure 4–7. 3-Bit Solution using PLHS502 D Flip-Flops

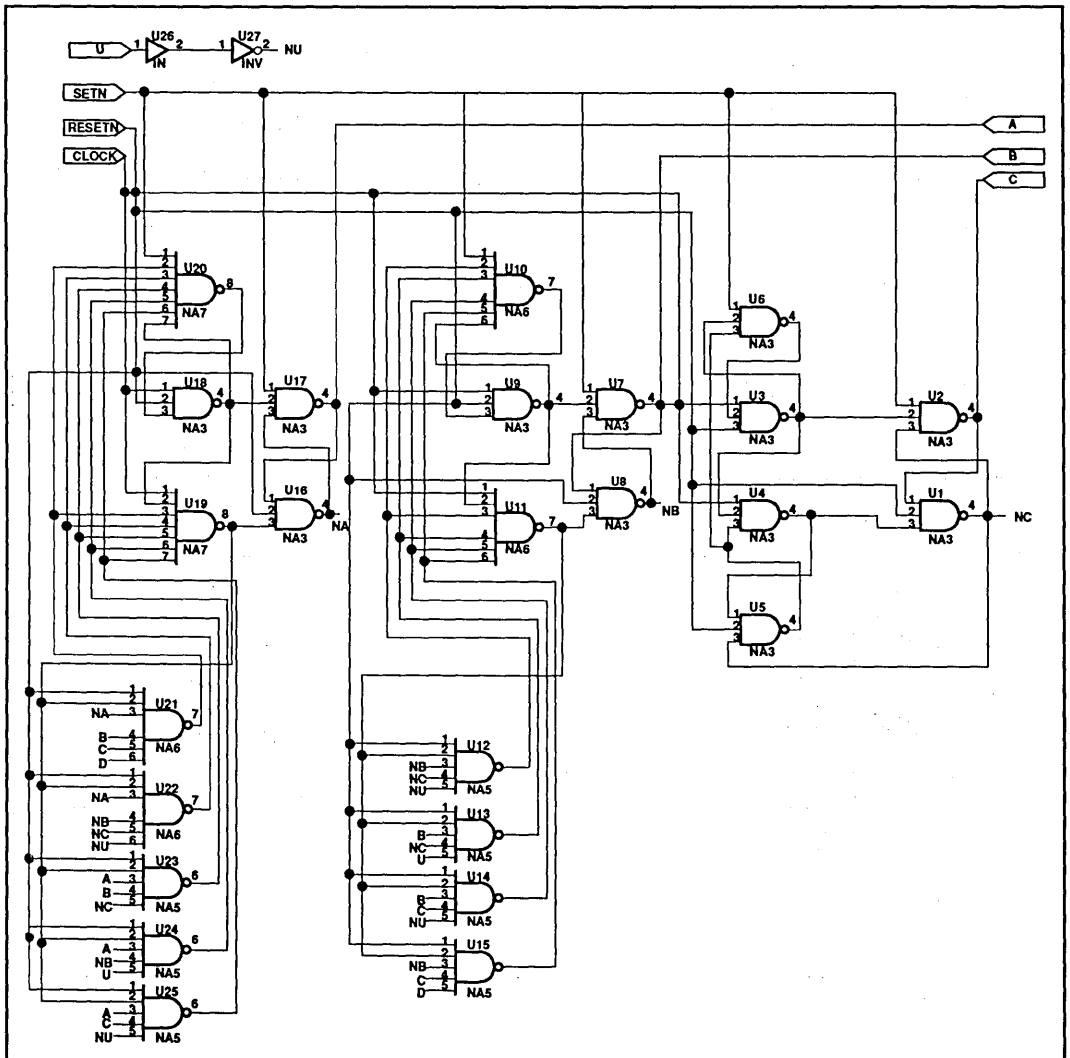
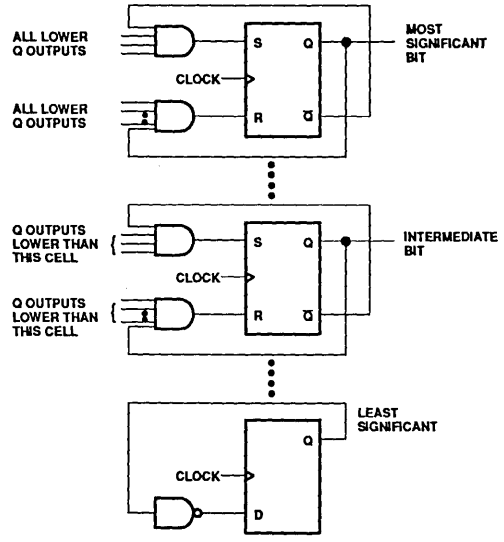


Figure 4–8. 3-Bit Solution With Merged NAND Flip-Flop

A couple of final notes on counter design. The basic structure for a simple up counter is shown in Figure 4–9. This one uses a D flip-flop least significant bit and a generic format for high order bits, as shown. Note that only 2 product terms per bit are needed.

Down counters can be treated similarly. Fairly general sequence generators can be configured from counters by simply adding a combinational decoder, as needed. One final counter example is shown in Figure 4–10 wherein a 10-bit counter is defined using the SNAP Boolean equation format. Note that the logic equations follow the format described in Figure 4–9. All resets are shared as well as all clocks. This design would require eight hard macro S-R flip-flops, one hard macro D flip-flop and one additional S-R flip-flop which SNAP would automatically configure from a hard macro D flip-flop.



NOTE: Asynchronous reset is not shown.

Figure 4–9. Up Counter Structure

```
@PINLIST
CLOCK I; RESET I;      A   0;   B   0;   C   0;   D   0;
                       E   0;   F   0;   G   0;   H   0;
                       I   0;   L   0;
@LOGIC EQUATIONS
A.D   =  /A;
B.S   =  /B*A;
B.R   =  B*A;
C.S   =  /C*A*B;
C.R   =  C*A*B;
D.S   =  /D*A*B*C;
D.R   =  D*A*B*C;
E.S   =  /E*A*B*C*D;
E.R   =  E*A*B*C*D;
F.S   =  /F*A*B*C*D*E;
F.R   =  F*A*B*C*D*E;
G.S   =  /G*A*B*C*D*E*F;
G.R   =  G*A*B*C*D*E*F;
H.S   =  /H*A*B*C*D*E*F*G;
H.R   =  H*A*B*C*D*E*F*G;
I.S   =  /I*A*B*C*D*E*F*G*H;
I.R   =  I*A*B*C*D*E*F*G*H;
L.S   =  /L*A*B*C*D*E*F*G*H*I;
L.R   =  L*A*B*C*D*E*F*G*H*I;
A.RST =  /RESET;
B.RST =  /RESET;
C.RST =  /RESET;
D.RST =  /RESET;
E.RST =  /RESET;
F.RST =  /RESET;
G.RST =  /RESET;
H.RST =  /RESET;
I.RST =  /RESET;
L.RST =  /RESET;
A.CLK =  CLOCK;
B.CLK =  CLOCK;
C.CLK =  CLOCK;
D.CLK =  CLOCK;
E.CLK =  CLOCK;
F.CLK =  CLOCK;
G.CLK =  CLOCK;
H.CLK =  CLOCK;
I.CLK =  CLOCK;
L.CLK =  CLOCK;
```

Figure 4–10. 10-Bit Up Counter — SNAP Style

Shifter Design Notes

Efficient shifter design is critical to achieve the fastest, most economical PLHS502 sequence recognizers. For ideal shifters, no additional gates should be required if the designer correctly exploits the hard macro flip-flops for the part. Normally, one views a shifter as an input to D, Q to D, Q to D, etc., like circus elephants walking nose to tail. But, the PLHS502 D flip-flop has an embedded NAND gate which, for this type of design, appears to "get in the way". By recognizing the availability of the $\overline{Q}$ output, if it is used instead, the design procedure remains "nose to tail" substituting $\overline{Q}$ for Q into the NAND (using only one input). Should a fancier shifter be required (see the Synchronous Receiver/Transmitter design at the end), the designer may choose to even implement soft macro "merged" shifter flip-flops.

The maximum internal "all shifter" capacity of the PLHS502 is 26 bits, assuming conventional structure flip-flops. If one were to design a schematic with 26 D flip-flop cells in cascade, SNAP would configure the first 8 from the hard macro Ds, the next 8 from the hard macro S-Rs and the last 10 from the NAND array.

An example illustrating a nonschematic captured 3-bit shifter follows. Figure 4–11 illustrates its' state diagram, Figure 4–12 shows the state equation solution and Figure 4–13 shows the very compact Boolean equation solution.

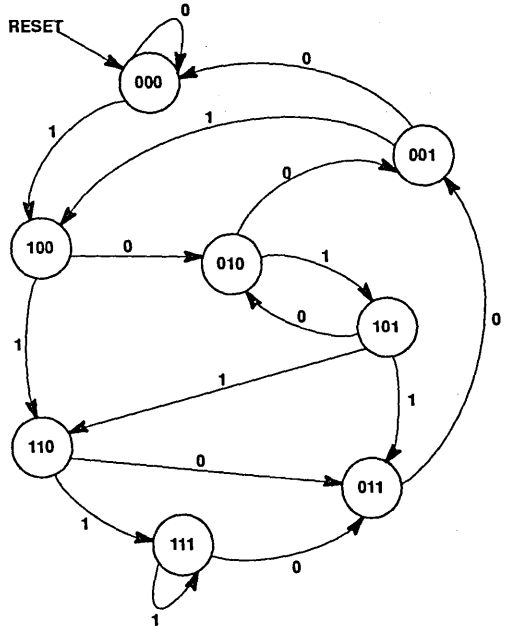


Figure 4–11. 3-Bit Shift Register State Diagram

```
@PINLIST
CLOCK I;   QA    0;
RESET I;   QB    0;
DATA       I;    QC   0;
@INPUT VECTORS
I1    =    DATA;
I0    =    /DATA;
@OUTPUT VECTORS
@STATE VECTORS
           [QA, QB, QC];
S0    =    000   B;
S1    =    001   B;
S2    =    010   B;
S3    =    011   B;
S4    =    100   B;
S5    =    101   B;
S6    =    110   B;
S7    =    111   B;
@TRANSITIONS
WHILE [ ]
IF RESET   THEN [S0]
WHILE [S0]
IF    I1   THEN [S4]
IF    I0   THEN [S0]
WHILE      [S1]
IF    I1   THEN [S5]
IF    I0   THEN [S0]
WHILE [S2]
IF    I1   THEN [S5]
IF    I0   THEN [S1]
WHILE [S3]
IF    I1   THEN [S5]
IF    I0   THEN [S1]
WHILE [S4]
IF    I1   THEN [S6]
IF    I0   THEN [S2]
WHILE [S5]
IF    I1   THEN [S6]
IF    I0   THEN [S2]
WHILE [S6]
IF    I1   THEN [S7]
IF    I0   THEN [S3]
WHILE [S7]
IF    I1   THEN [S7]
IF    I0   THEN [S3]
```

Figure 4–12. 3-Bit Shifter — State Equations

```
@PINLIST
CLOCK I;
DATA    I;
RESET I;
QA      O;
QB      O;
QC      O;
@LOGIC EQUATIONS
;
QA.D   = DATA;
QB.D    =  QA;
QC.D    =  QB;
QA.RST  =  RESET;
QB.RST  =  RESET;
QC.RST  =  RESET;
QA.CLK  =  CLOCK;
QB/CLK  =  CLOCK;
QC.CLK  =  CLOCK;
```

Figure 4–13. 3-Bit Shifter — Boolean Solution

PLHS502
Application Notes
Vol. 1

Programmable Logic Devices

ADDITIONAL DESIGN GUIDELINES

The following guideline summary is by no means complete. Rather, it is a list of straightforward substitutions which the designer can make to help guarantee that the design fits. The basic approach is to build the design using the basic building blocks of the architecture. For the PLHS502 this means using NAND gates, D flip-flops and S-R flip-flops. To make this clear, we will enumerate and illustrate good basic design substitutions.

1. Use NAND gates whenever possible.
2. Use S-R flip-flops for counters over 4 bits long.
3. Use D flip-flops (if possible) for the least significant counter bit.
4. If possible eliminate NOR functions by converting to AND with complemented input (use $\overline{Q}$ on flip-flops and available input complements).
5. AND gates which feed NAND gates can be replaced by wider NAND gates (exploit NAND width).
6. Eliminate all extra inversions by exploiting input complements and flip-flop /Q outputs.
7. If you exceed your flip-flop budget of hard flip-flops, put the most complex flip-flop configurations into soft macros (see Section 6). Fold the gating function into the flip-flop.
8. Careful output pin assignment can result in a free logic function as the signal leaves the device. So, assign complemented, buffered outputs accordingly. Exclusive-OR/parity controlled outputs are slower, so assign them accordingly.
9. Build toggle chains out of D flip-flops, then S-R flip-flops, to conserve NAND gates.
10. The NAND-feeding D-FF structure may be thought of as an AND-feeding D-FF with Q and $\overline{Q}$ reversed.
11. For very large counters, converting D-FFs to J-K FFs may be appropriate. If necessary, then do so.
12. Efficient methods for implementing Exclusive-OR functions are described in PLHS501 Applications Notes Volume 2.

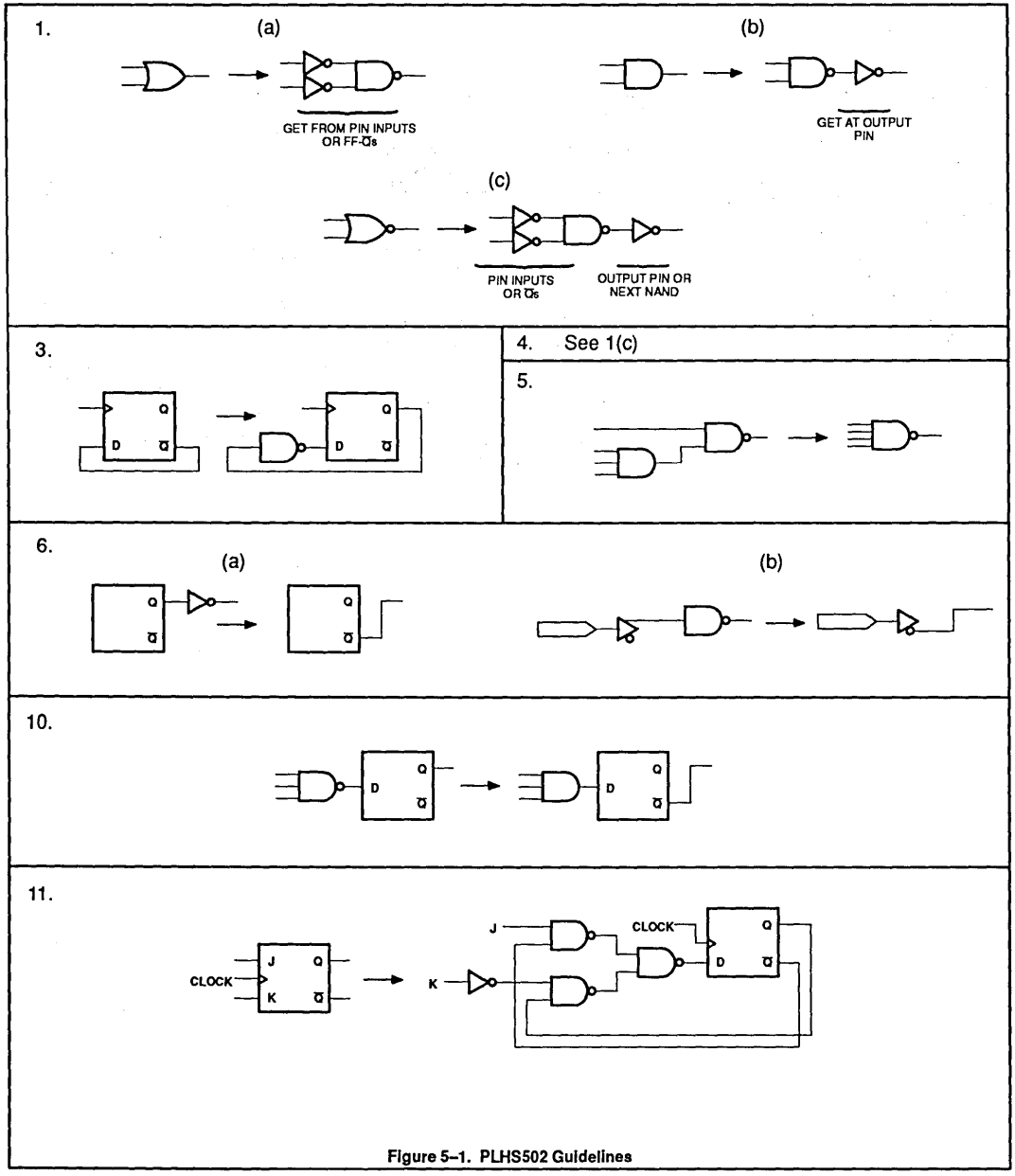


Figure 5–1. PLHS502 Guidelines

PLHS502
Application Notes
Vol. 1

Programmable Logic Devices

ADDITIONAL PLHS502 APPLICATIONS

These examples illustrate various applications the PLHS502 is capable of, which are quite interesting.

Byte Data Pipe

A common system building block is the byte data pipe illustrated in Figure 6–1. This elementary structure illustrates a parallel cascade of octal registers where each tier is independently clocked. Data arrives at the input pins and is clocked in by clock A. After settling, this is clocked into the second tier by clock B, the third tier by clock C and the output tier by clock D. The two center registers are generated from NAND gate transparent latches comprised of three gates. This classic three-gate latch has a static hazard in the ones, but careful timing and masking the input and output logical image to the outside world, with edge triggered registers, essentially eliminates this evidence. This consumes 48 of the 64 gates budget in the PLHS502. The remaining gates may now be used as needed to configure this data path for a more specific applications such as:

a. A distributed decoder as in a RISC pipeline. Each remaining gates can detect 32-bit internal state combinations (expanded to 40 bits if simultaneous examination of the input pins is desired).

b. A 4-byte queue for interprocessor and processor to bus communications and synchronization.

c. A 3-, 4-, or 5-byte sequence detector for byte oriented protocols.

Clearly, one of the internal register tiers could be freed up or the arrangement altered to have a three position data pipe and a group of S-R registers to implement a bus handshaker or internal counters for a queue pointer, etc.

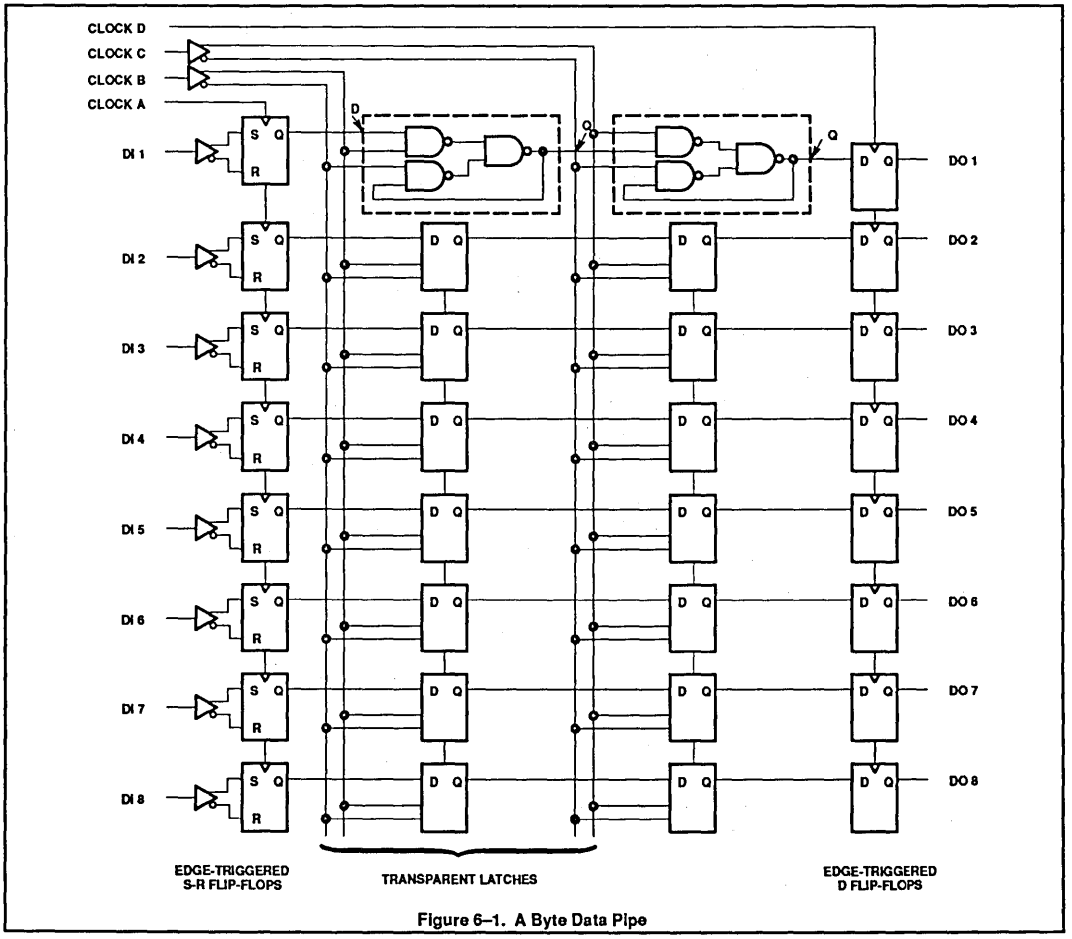


Figure 6–1. A Byte Data Pipe

Custom CAMS

Another common block used in memory and I/O systems today is the content addressable memory (CAM), illustrated in Figure 6–2. These are used for associative searching and often implement the TAG structure or VALID bit structure for the translation lookaside buffer in a cache system. CAMs more closely resemble a small register array where each cell has an independent data compare operation with a global polling mechanism. This example illustrates the PLHS502's ability to realize a small, relatively fast 4X4 CAM. In this example, the 16 register cells are made with the D flip-flop and the S-R flip-flops configured as Ds. Each flip-flop output is compared to a corresponding data input with a coincidence function which is generated from three NANDs. Groups of four comparators are then ANDed together to generate the HIT signals indicating the presence of a 4-bit item. If speed and gate economy are required, the composite compare function can be generated, exploiting wide gates rather than deep cascades.

The CAM operation is simple. The CAM must first be loaded, with four bits in each tier. Loading was chosen to be 4-bit parallel with independent clocking. This allows "one transaction" replacement and is the most flexible approach for implementing arbitrary updating policies. Once it is loaded, the 4-bit data is applied on the same lines (this could easily be changed to four different lines).

When a value is applied, a "HIT" is generated if the current value matches one of the stored 4-bit items. How the "HIT" is used by the outside system is system dependent. If a value is applied and there is no "HIT," there is no response. Again, there are a number of remaining internal NAND gates as well as most of the gates tied to the I/O pins to perform additional tasks and adapt the behavior of this building block.

This is a very efficient structure to implement a direct mapped cache, where four 1-megaword regions could create the logical image of a 16-megaword region. The time from valid address to valid "HIT" is about 20 nanoseconds — max.

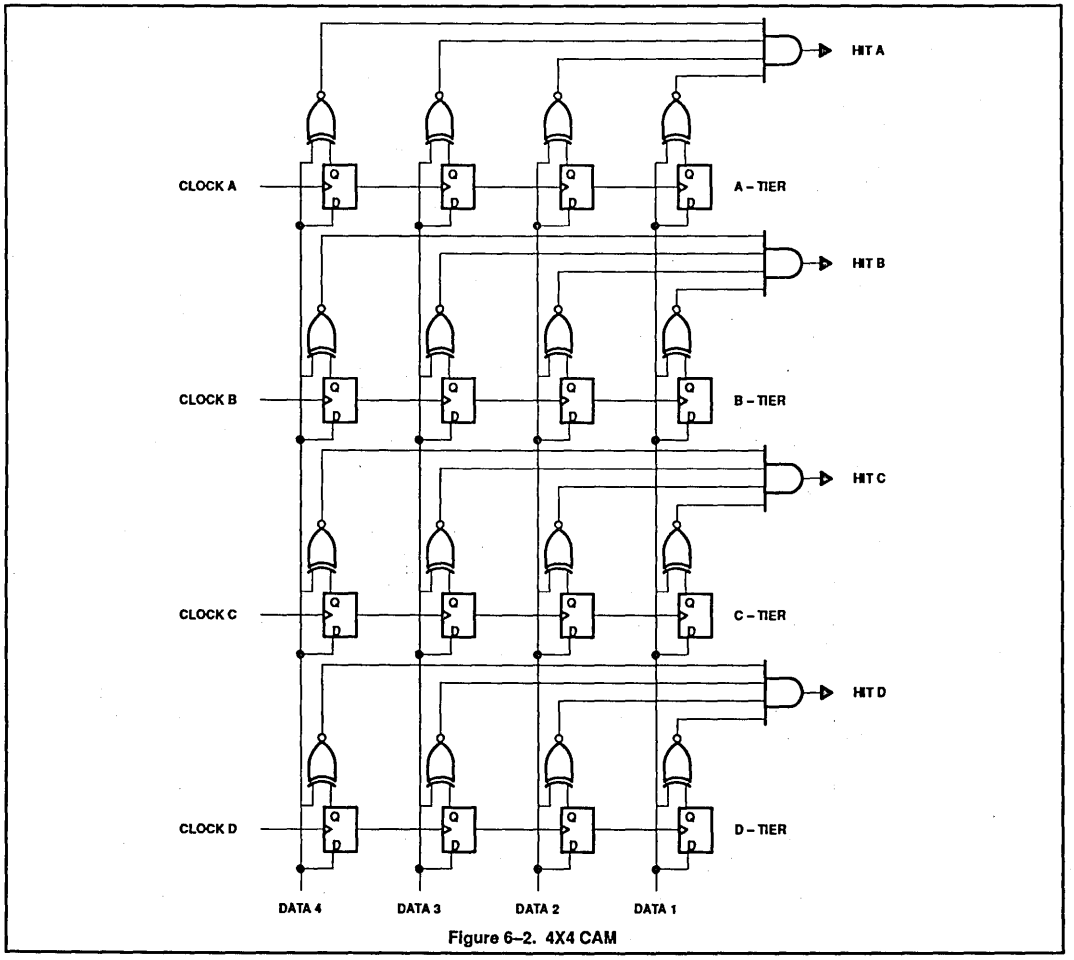


Figure 6–2. 4X4 CAM

System Scoreboard

High digital system throughput may be achieved by allowing high-speed processors to run independently at their own rate. Hence, operation speed is only limited when data or function dependencies occur. This requires a special sort of synchronization mechanism to allow independent processes to realign when necessary. A classic solution to this is the Scoreboard concept, illustrated in Figure 6–3. In essence, this is an overall system register with independent status bits assigned to specific functional units and data registers. The idea is an old one, but has recently resurfaced since modern RISC designers have begun resurrecting highly parallel CISC architectured artifacts. The concept is valid for a computer system where resources are all viewed as independent processors.

Basically, one bit in the Scoreboard is assigned to each distinct functional unit or independent data register. When a request (from anywhere) arrives for that unit, it is assigned unless the unit is busy. This locks out future assignment until that resource is released. For an elementary Scoreboard, the basic requirement is independent registers cells (i.e., must be able to request, assign and release each one, essentially asynchronously). Figure 6–5 shows an implementation of a 9-element Scoreboard. This requires 27 inputs and 9 outputs (using cell version 1) and leaves 5 bi-directional lines, 7 outputs and most of the foldback NANDs. A global reset may be added by logically ORing each release signal with a global reset through the NAND function tied to each flip-flop. This design uses the 8 D flip-flops and one S-R flip-flop per status sentinel. This leaves 7 flip-flops to implement additional functions as well.

Input pins can be freed up if an external clock is assigned to all of the allocate signals and each request is gated with the flip-flops current status (cell version 2). The rendition using cell version 2 still permits independent release of each cell.

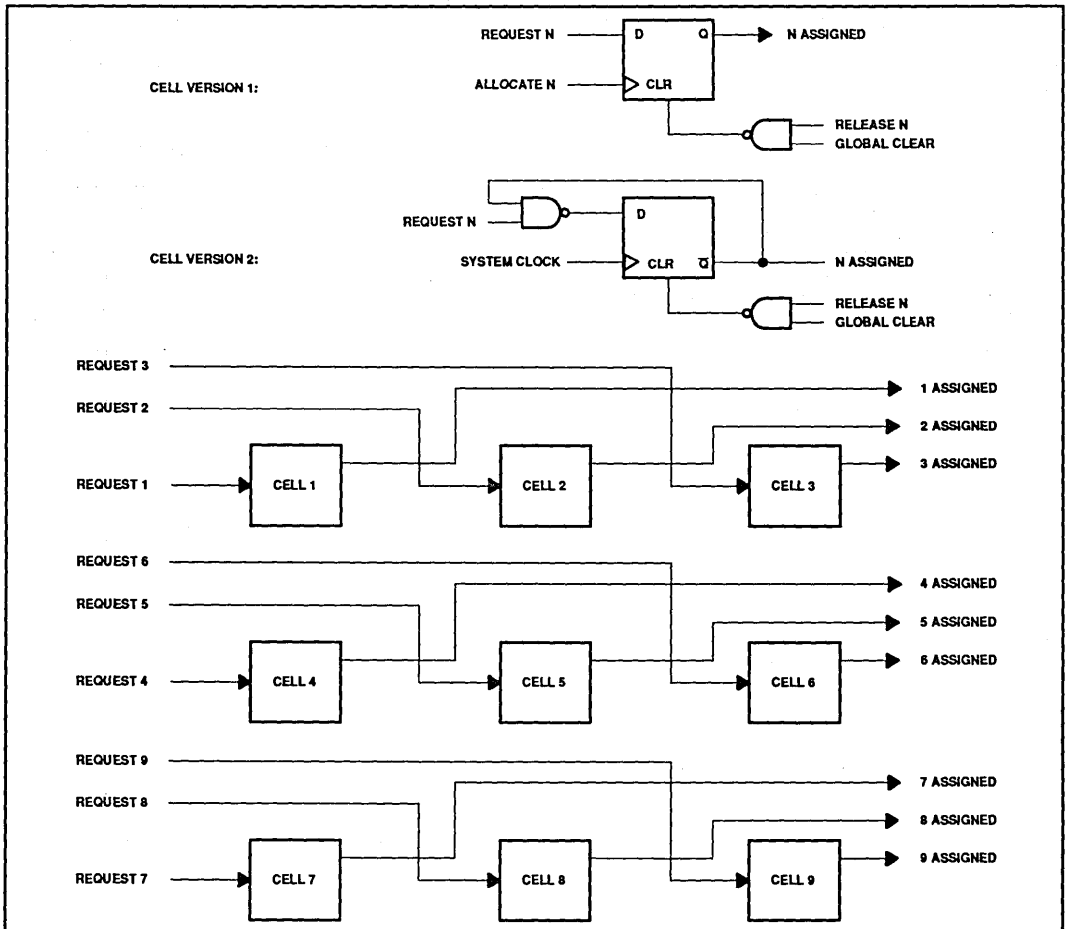


Figure 6–3. System Scoreboard

Synchronous Receiver/ Transmitter

This system building block is an elementary synchronous receiver and transmitter. This example will illustrate multiple independent state machine designs as well as some standard design techniques to utilize every available function component out of the PLHS502. Figure 6–4 shows the elementary structure of the synchronous serial communication device. Basically, it is two independent machines where each half is comprised of a counter and shifter. Figure 6–5 depicts the high-level schematic of Figure 6–4. The transmitter must load a byte in parallel and ship the contents out serially. The receiver must receive serial data and indicate the presence of a correctly framed byte. From partitioning considerations, the counters will be be generated from the internal S-R flip-flops. The shifters can be made from the internal D flip-flops but the requirement is for at least 16 D flip-flops — which means at least 8 must be constructed from NANDs. Intuitively, it would be best to exploit the free NAND on the hard macro D flip-flops to construct a simple MUX for each D-cell as shown in Figure 6–6a. This would use 16 NANDs. Then, the receive register could be constructed from an octal cascade of flip-flops constructed from six NANDs each (Figure 6–6b). This would use 48 NAND gates and expend the rest of foldback array. This would create a problem to complete the design of the byte counters because they each require four gates to construct the drive terms for the S-R flip-flops. An old gate array trick is very appropriate here. Figure 6–6c shows a 2-input multiplexor driving a 6-gate D flip-flop. By logical merging and exploiting the width and nesting depth of PML, one can get the entire function realized by the final cell version of Figure 6–6d, which takes only seven gates to do the entire cell. Hence, making the transmitter from 56 NANDs (8 copies of the 7-gate cell) gives 8 left over foldback NANDs for the generation of two 3-bit counters. It should be noted that this is a faster solution because the D signal passes through as many gates as without the multiplexor. Two levels of time delay are achieved instantaneously, as shown in Figure 6–6, T1 = T3 which is less than T2. The receive register is simply constructed from the hard macro D flip-flops in direct cascade. This leaves many remaining input and output pins which can be used elsewhere (i.e., fast I/O decode and such) for other system needs. One obvious addition would be to implement decode of specific characters from the receive buffer using the many remaining output pins.

Figure 6–7 shows the schematic of one of the 3-bit counter modules which permit byte framing in the receiver and transmitter. Figure 6–8 through 6–10 detail the exact schematics used to complete the receiver/transmitter ("RECTRAN") design.

Figure 6–11 shows the SNAP simulation of the 7-gate composite cell alternately loading and shifting. In Figure 6–11, the text shown at the top is the Simulation Control File which constitutes the stimuli vectors for the simulation. The waveform below depicts the stimuli and corresponding response vectors. Incidentally, the two remaining S-R flip-flops can be used for anything, but one can signal a data available condition to the outside world for the receiver and the other can indicate Busy or Done for the transmitter to reflect some status to the outside world.

A complete series of simulation is presented in Figures 6–12 through 6–15. The simulation control file depicted in Figure 6–16 performs the operations whose result waveforms are shown in Figures 6–13 through 6–15. Figure 6–17 shows the SNAP macro netlist which binds the smaller pieces of the overall design together. Figure 6–18 provides the netlist which constitutes the receiver piece of RECTRAN while Figure 6–19 gives the same for the transmitter. Expanding the hierarchy, Figure 6–20 displays the exact netlist for the "flopcell" and Figure 6–21 shows how the netlist for one of the three bit counters appears. The entire RECTRAN design was captured schematically using OrCAD/SDT.

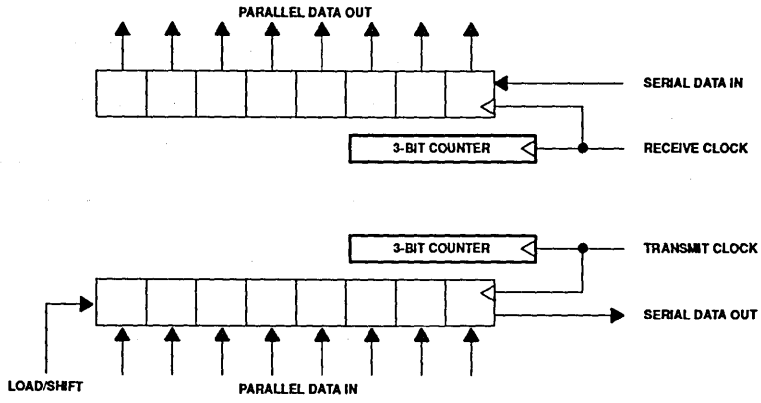


Figure 6–4. Synchronous Receiver/Transmitter — RECTRAN

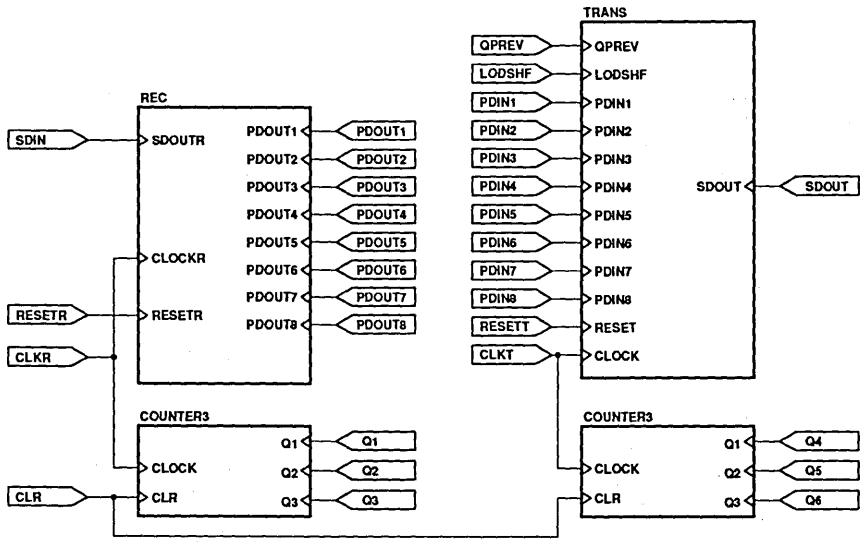


Figure 6–5. High Level RECTRAN Schematic

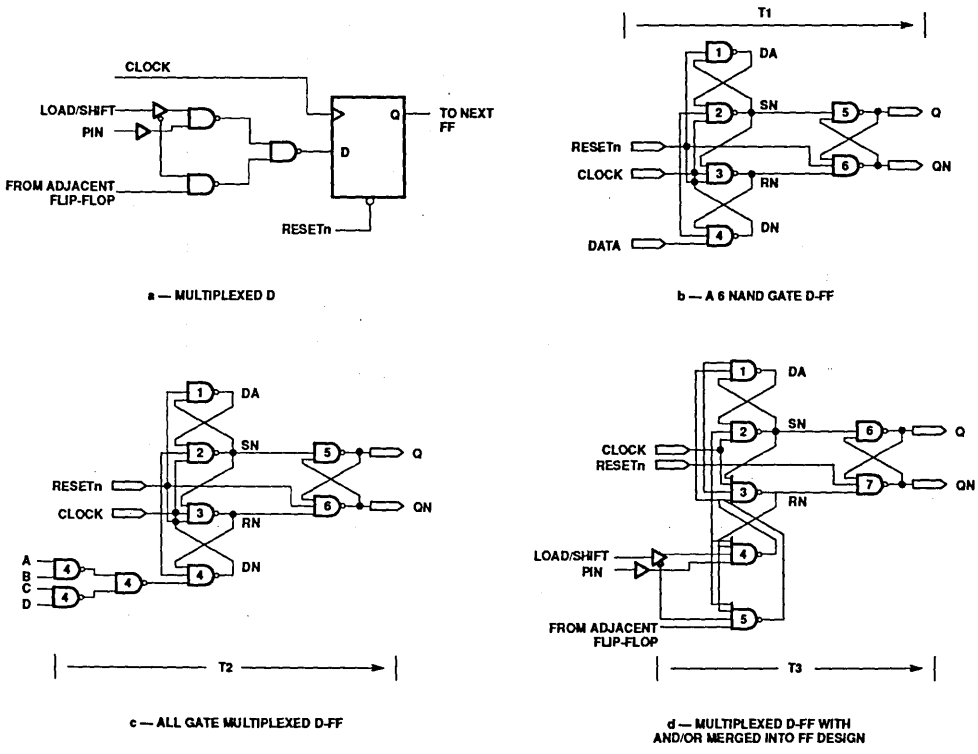


Figure 6–6. Custom Flip–Flop Metamorphosis

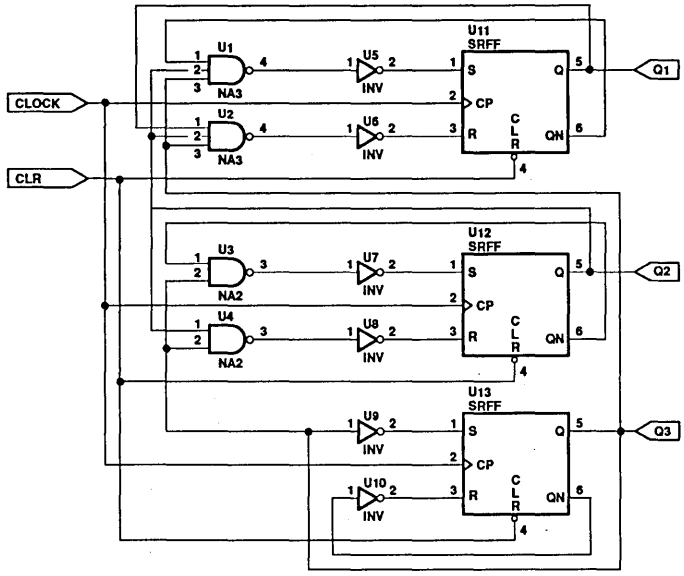


Figure 6–7. 3-Bit Counter Schematic

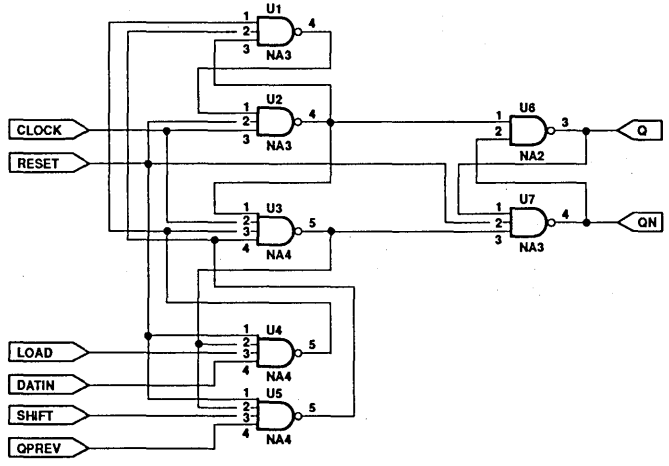


Figure 6–8. Final Merged D Flip-Flop Schematic

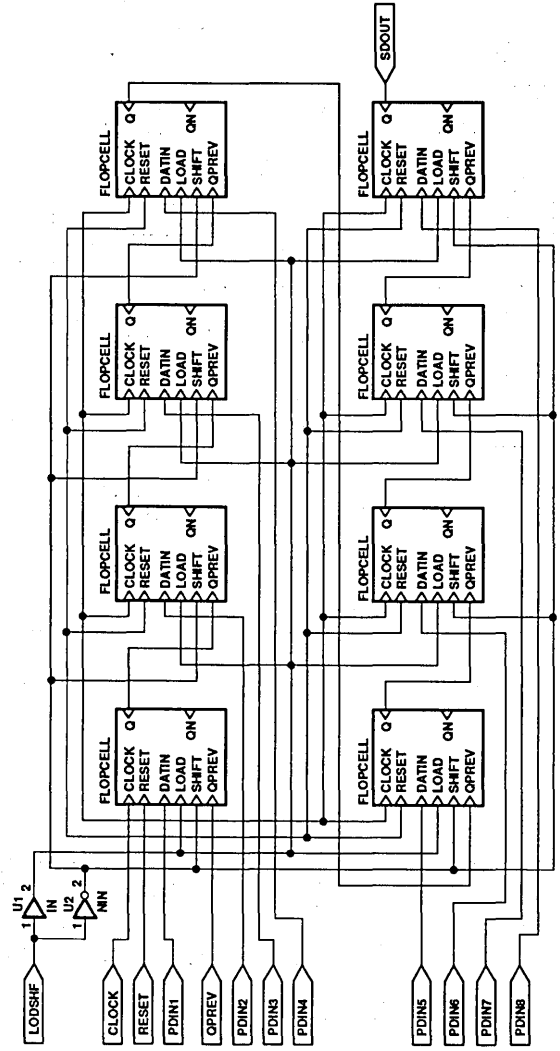


Figure 6–9. Transmitter Schematic

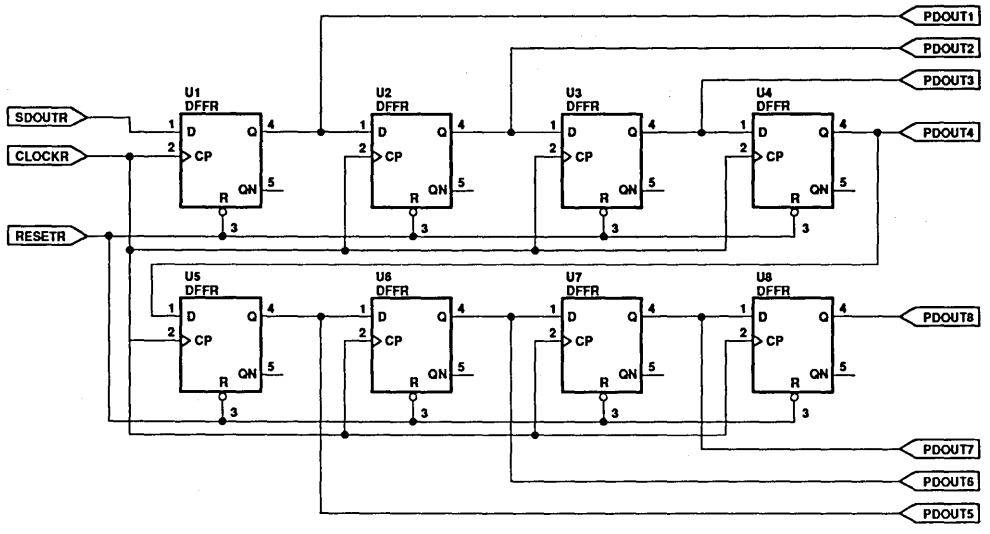


Figure 6–10. Receiver Schematic

```
*  Simulation Control File for Flopcell
*  Strategy   1.  Reset issued initially
*             2.  Define a repetitive clock @ 10MHz
*             3.  Load a logical one
*             4.  Switch mode to shift, and shift in a logical zero
*             5.  Load a logical zero
*             6.  Switch mode to shift, and shift in a logical one
*             7.  Issue another reset
*
   S 1(20,40,450,470)reset
   S 0(50,100,etc.)clock
   S 0(45,70,245,270)lod shf
   S 0(45,90)datin
   S 0(345,370)q prev
   P reset,clock,lod_shift,datin,q_prev,q
   SU Time=500
   F
```

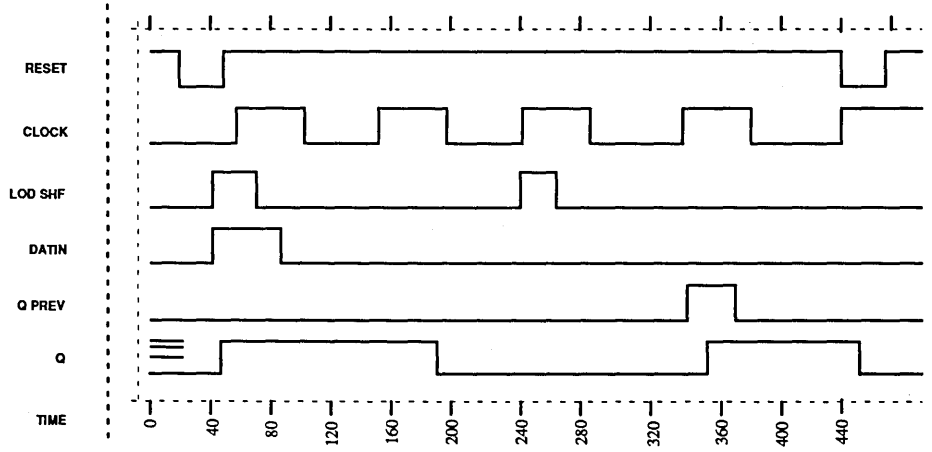


Figure 6–11. SNAP Simulation of the Merged Flip-Flop

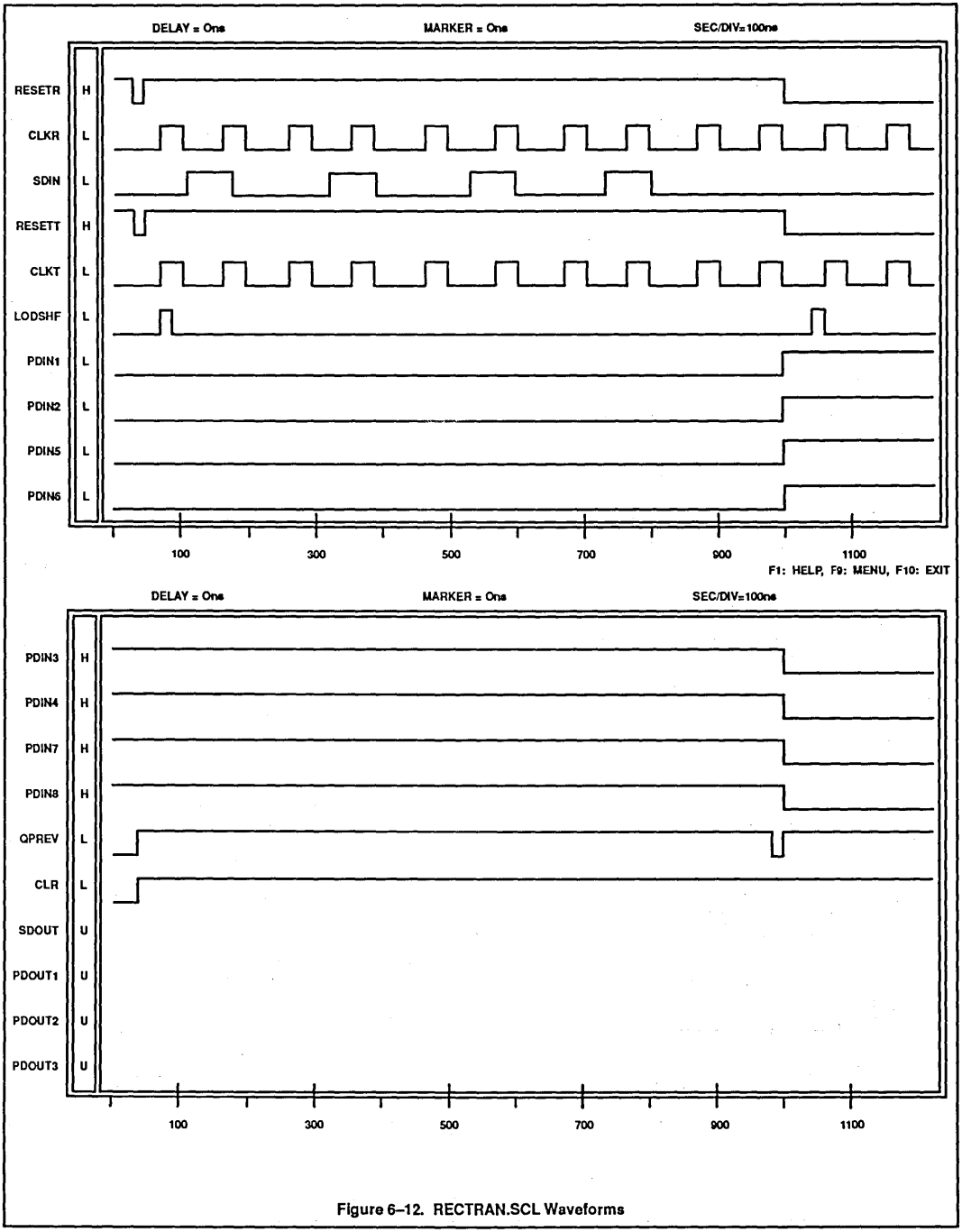


Figure 6–12. RECTRAN.SCL Waveforms

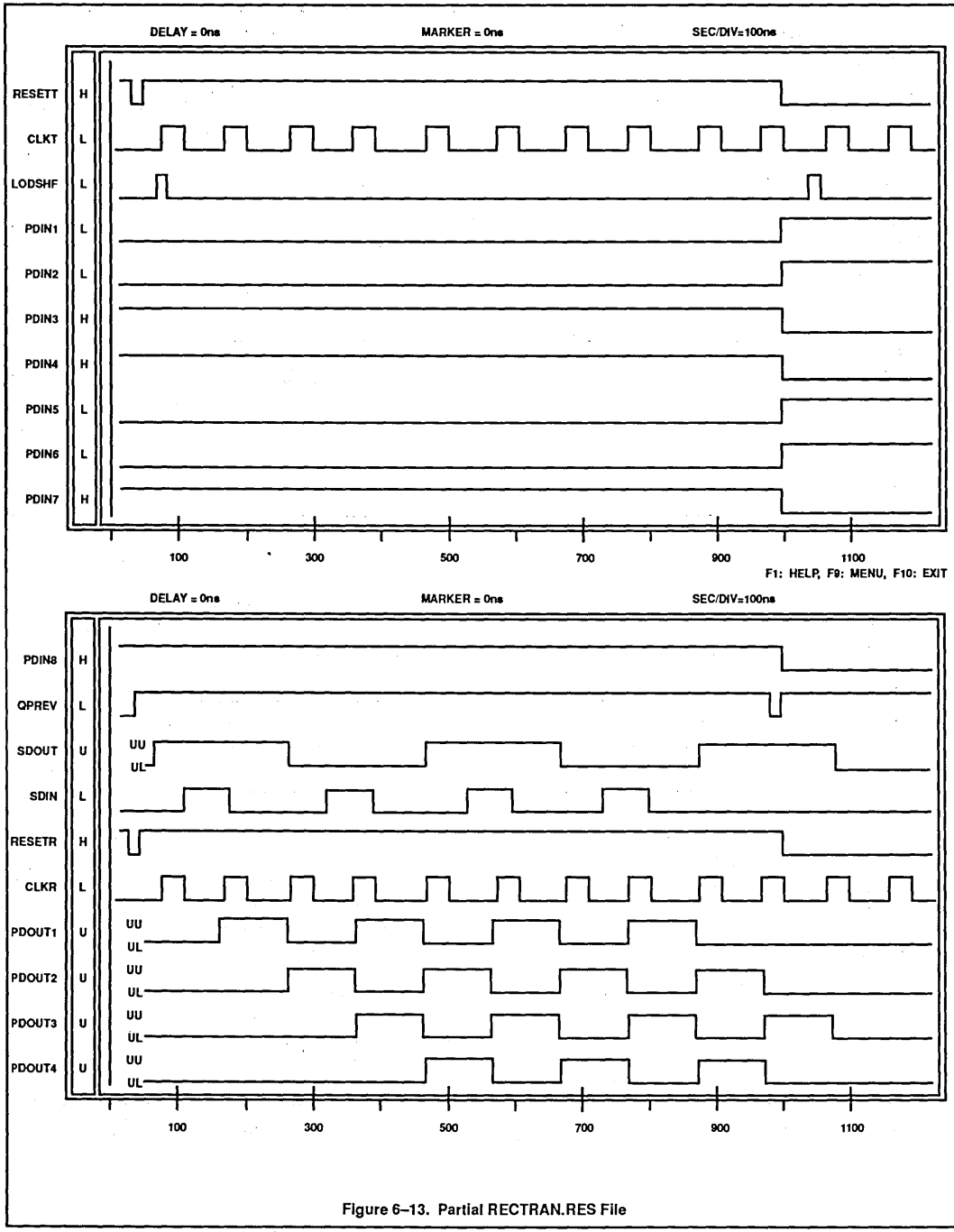


Figure 6–13. Partial RECTRAN.RES File

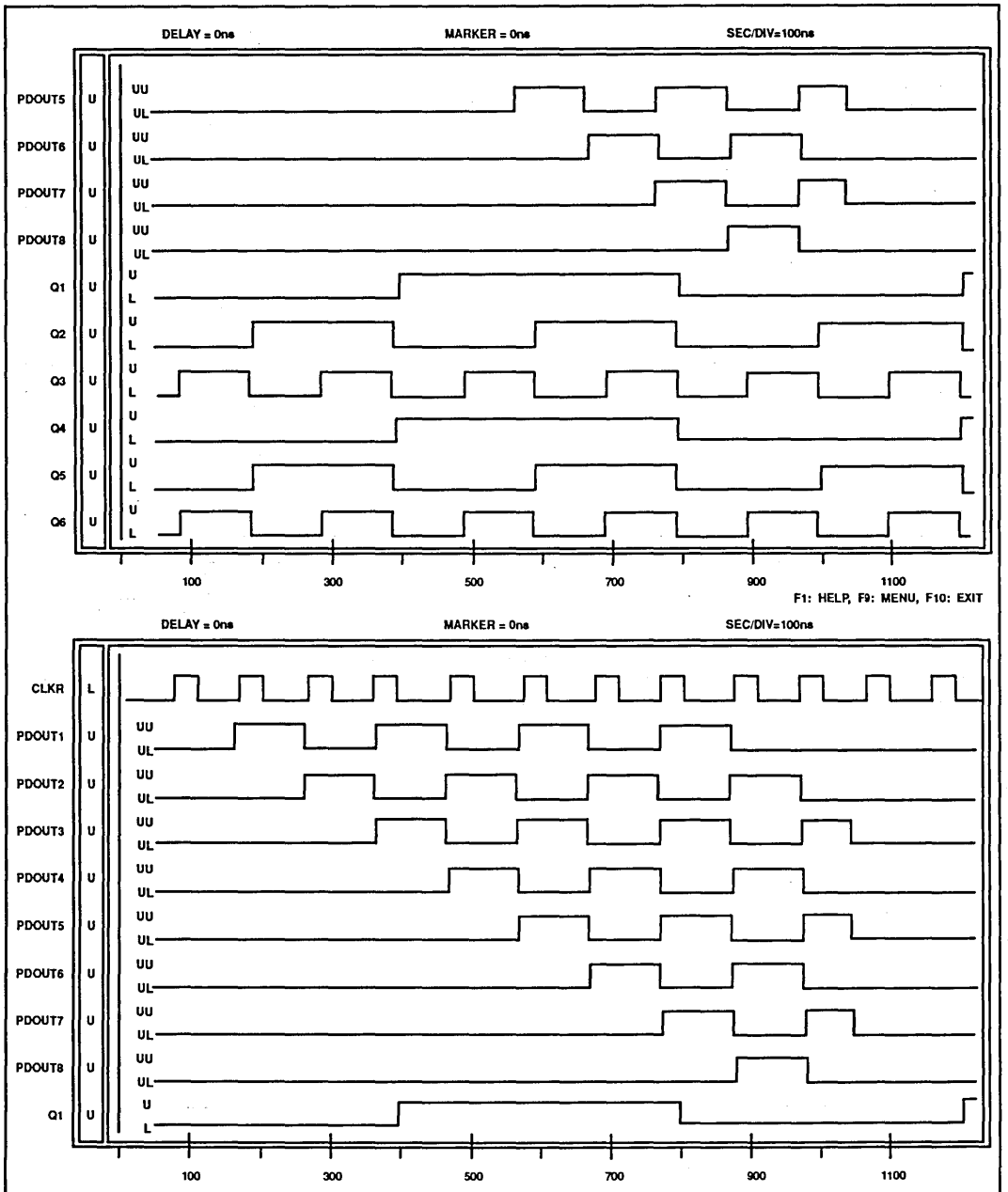


Figure 6–14. Additional RECTRAN.RES File

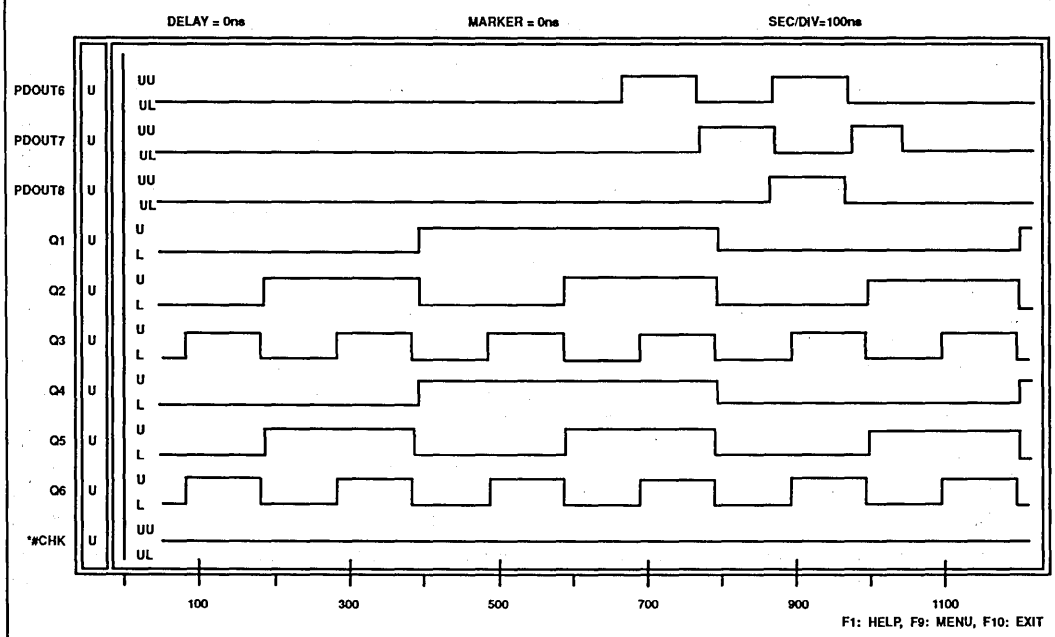


Figure 6–15. Final RECTRAN.RES File

```
S   1(20,40,1000)RESETR
S   0(50,100,ETC)CLKR
S   0(100,150,300,350,500,550,700,750)SDIN
S   1(20,40,1000)resetT
S   0(50,100,etc)clkT
S   0(45,70,1025,1050)lodshf
S   0(1000)pdin1,pdin2,pdin5,pdin6
S   1(100)pdin3,pdin4,pdin7,pdin8
S   0(20)CLR
* S 1(2000)VCC
 p resetT,clKT,lodshf,pdin1,pdin2,pdin3,pdin4,pfin5,pdin6
# ,pdin7,pdin8,qprev,sdout,SDIN,RESETR,CLKR,PDOUT1,
# PDOUT2,PDOUT3,PDOUT4,PDOUT5,PDOUT6,PDOUT7,PDOUT8,
# Q1,Q2,Q3,Q4,Q5,Q6
 PCO
 sutime=2000
 f
```

Figure 6–16. RECTRAN.SCL Text

```
************************************************************
*           Output of Netgene              Version 1.0     *
* Date:  5/ 3/1989                         Time:  18:36:25   *
************************************************************
* Input File Name        :  RECTRAN.EDF                     *
* Netlist File           :  RECTRAN.MAC                     *
************************************************************
*
 MACRO
*
Z RECTRAN_SCH I(CLKR,CLKT,CLR,LODSHF,PDIN1,PDIN2,PDIN3,PDIN4,
# PDIN5,PDIN6,PDIN7,PDIN8,QPREV,RESETR,RESETT,SDIN) O(PDOUT1,
# PDOUT2,PDOUT3,PDOUT4,PDOUT5,PDOUT6,PDOUT7,PDOUT8,Q1,Q2,
# Q3,Q4,Q5,Q6,SDOUT)
*
CELL_REC_0_0 REC I(CLKR,RESETR,SDIN) O(PDOUT1,PDOUT2,PDOUT3,
# PDOUT4,PDOUT5,PDOUT6,PDOUT7,PDOUT8)
BLK01 TRANS I(CLKT,LODSHF,PDIN1,PDIN2,PDIN3,PDIN4,PDIN5,
# PDIN6,PDIN7,PDIN8,QPREV,RESETT) O(SDOUT)
BLK02 COUNTER3 I(CLKR,CLR) O(Q1,Q2,Q3)
BLK03 COUTNER3 I(CLKT,CLR) O(Q4,Q5,Q6)
*
 MEND
*
```

Figure 6–17. High-Level RECTRAN Macro File

```
*********************************************
*      Output of Netgene       Version 1.0    *
* Date:  5/ 2/1989            Time:  13:35:45 *
*********************************************
* Input File Name      :    REC.EDF           *
* Netlist File         :    REC.MAC           *
*********************************************
*
 MACRO
*
Z REC_SCH I(CLOCKR,RESETR,SDOUTR) O(PDOUT1,PDOUT2,PDOUT3,
# PDOUT4,PDOUT5,PDOUT6,PDOUT7,PDOUT8)
*
U1 DFFR I(SDOUTR,CLOCKR,RESETR) O(PDOUT1,DMO1)
U2 DFFR I(PDOUT1,CLOCKR,RESETR) O(PDOUT2,DMO2)
U3 DFFR I(PDOUT2,CLOCKR,RESETR) O(PDOUT3,DMO3)
U4 DFFR I(PDOUT3,CLOCKR,RESETR) O(PDOUT4,DMO4)
U5 DFFR I(PDOUT7,CLOCKR,RESETR) O(PDOUT8,DMO5)
U6 DFFR I(PDOUT6,CLOCKR,RESETR) O(PDOUT7,DMO6)
U7 DFFR I(PDOUT5,CLOCKR,RESETR) O(PDOUT6,DMO7)
U8 DFFR I(PDOUT4,CLOCKR,RESETR) O(PDOUT5,DMO8)
*
 MEND
*
```

Figure 6–18. Receiver Macro File

```
*************************************************
*          Output of Netgene          Version 1.0       *
* Date:  1/ 5/1989                      Time:  13:37:13 *
*************************************************
*                                                       *
* Input File Name    :    TRANS.EDF                     *
* Netlist File       :    TRANS.MAC                     *
*                                                       *
*************************************************
*
 MACRO
*
Z TRANS_SCH I(CLOCK,LODSHF,PDIN1,PDIN2,PDIN3,PDIN4,PDIN5,
# PDIN6,PDIN7,PDIN8,QPREV,RESET) O(SDOUT)
*
U1 IN I(LODSHF) O(SN01)
U2 NIN I(LODSHF) O(SN09)
BLK01 FLOPCELL I(CLOCK,PDIN1,SN01,QPREV,RESET,SN09) O(SNO2,
# DM01)
BLK02 FLOPCELL I(CLOCK,PDIN2,SN01,SN02,RESET,SN09) O(SN03,
# DM02)
BLK03 FLOPCELL I(CLOCK,PDIN3,SN01,SN03,RESET,SN09) O(SN04,
# DMO3)
BLK04 FLOPCELL I(CLOCK,PDIN4,SN01,SN04,RESET,SNO9) O(SN05,
# DMO4)
BLK05 FLOPCELL I(CLOCK,PDIN8,SN01,SN08,RESET,SN09) O(SDOUT,
# DM05)
BLK06 FLOPCELL I(CLOCK,PDIN7,SN01,SN07,RESET,SN09) O(SN08,
# DM06)
BLK07 FLOPCELL I(CLOCK,PDIN6,SN01,SN06,RESET,SN09) O(SN07,
# DM07)
BLK08 FLOPCELL I(CLOCK,PDIN5,SN01,SN05,RESET,SN09) O(SN06,
# DM08)
*
 MEND
*
```

Figure 6–19. Transmitter Macro File

```
*************************************************
*          Output of Netgene          Version 1.0       *
* Date:  1/ 5/1989                      Time:  13:34:36 *
*************************************************
*                                                       *
* Input File Name    :   FLOPCELL.EDF                   *
* Netlist File       :   FLOPCELL.MAC                   *
*                                                       *
*************************************************
*
 MACRO
*
Z FLOPCELL_SCH I(CLOCK,DATIN,LOAD,QPREV,RESET,SHIFT) O(Q,
# QN)
*
U1 NA3 I(SN05,SN01,SN03) O(SN02)
U2 NA3 I(SN02,RESET,CLOCK) O(SN03)
U3 NA4 I(SN03,CLOCK,SN05,SN01) O(SN04)
U4 NA4 I(RESET,SN04,LOAD,DATIN) O(SN05)
U5 NA4 I(RESET,SN04,SHIFT,QPREV) O(SN01)
U6 NA2 I(SN03,QN) O(Q)
U7 NA3 I(Q,RESET, SN04) O(QN)
*
 MEND
*
```

Figure 6–20. Flopcell Macro File

```
*************************************************
*          Output of Netgene          Version 1.0       *
* Date:  1/ 5/1989                      Time:  13:42:14 *
*************************************************
*                                                       *
* Input File Name    :   COUNTER3.EDF                   *
* Netlist File       :   COUNTER3.MAC                   *
*                                                       *
*************************************************

*
 MACRO
*
Z COUNTER3_SCH I(CLOCK,CLR) O(Q1,Q2,Q3)
*
U1 NA3 I(SN03,Q2,Q3) O(SN01)
U2 NA3 I(Q1,Q2,Q3) O(SN04)
U3 NA2 I(SN09,Q3) O(SN06)
U4 NA2 I(Q2,Q3) O(SN10)
U5 INV I(SN01) O(SNO2)
U6 INV I(SN04) O(SNO5)
U7 INV I(SNO6) O(SNO7)
U8 INV I(SN10) O(SN08)
U9 INV I(Q3) O(SN11)
U10 INV I(SN12) O(SN13)
U11 SRFF I(SNO2,SNO5,CLOCK,CLR) O(Q1,SN03)
U12 SRFF I(SN07,SN08,CLOCK,CLR) O(Q2,SNO9)
U13 SRFF I(SN11,SN13,CLOCK,CLR) O(Q3,SN12)
*
 MEND
*
```

Figure 6–21. 3–Bit Counter Macro File

PLHS502
Application Notes
Vol. 1

Programmable Logic Devices

ADVANCED FLIP-FLOP MERGING

Folding logic functions directly into the workings of a flip-flop has a unique pay-off with PML. (As illustrated in Sections 4 and 6). Since the basic approach may be extended to other flip-flop structures, this technique may be used more "generally". The figures provided in this section illustrate examples of simple and complicated structures.

Consider the D-latch in Figure 7–1:

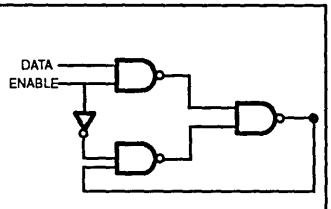


Figure 7–1. D-Latch

If the data input point is driven from a two-level logic function as shown in Figure 7–2(a), the gate count is six with one inverter. The input signals pass through two logic layers before entering the latch, then through two more to reach the Q output.

From Figure 7–2(a), note that the data input to gate G4 is driven High from the G3 output when either G1 or G2 (or both) is driven Low. Also, note that when Data is High, and Enable is High, the output of G4 will be Low. Under these conditions, the outputs of G1 or G2 are driven in a direction similar to the output of G3 and G4. By eliminating G3 and G4 and substituting G1 and G2 where G4 was, some gate and speed saving is gained as shown in Figure 7–2(b). Note that the Enable signal has to be reinserted carefully. This permits a faster, cheaper, and more efficient merged latch. This is the simplest structure and the reader is encouraged to prove the operation by simulation or construction. The general approach is a simple expansion of these steps.

1. Isolate the positive asserted flip-flop input.
2. Isolate the two-level AND/OR driving function.
3. Eliminate the NAND gate in step 1 but preserve inputs and outputs.
4. Eliminate the second-level NAND from step 2, but preserve inputs in the step 2 AND/OR structure.
5. Place the corresponding intermediate outputs to replace the eliminated flip-flop input gates output.
6. Place any inputs which fed the flip-flop input gate onto the new input points from step 5 (i.e., the input points of the step 2 AND/OR structure).

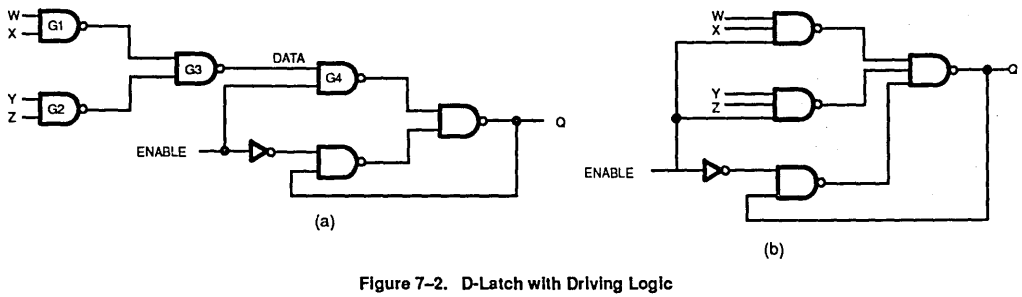


Figure 7–2. D-Latch with Driving Logic

Let's illustrate the procedure by applying it to a more complicated flip-flop — the dreaded J-K! Figure 7–3 depicts a J-K with a two-level logic function tied to J and a different function tied to K. J gets the sum of three product terms and K gets the sum of two.

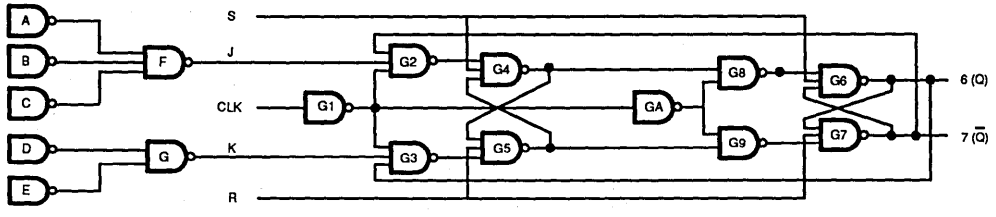


Figure 7–3. J-K with Gating

Because the inputs to gates A, B, C, D, E will be maintained, we don't care what they are. From the recipe:

Step 1. Isolate the positive asserted flip-flop input.
For J this is gate G2.
For K this is gate G3.

Step 2. Isolate the two-level AND/OR driving function.
(see function 1 and 2)

Step 3. Eliminate the NAND inputs in step 1 but preserve inputs and outputs.
(Figure 7–4(a))

Step 4. Eliminate the second-level NAND from step 2.
(Figure 7–4(a))

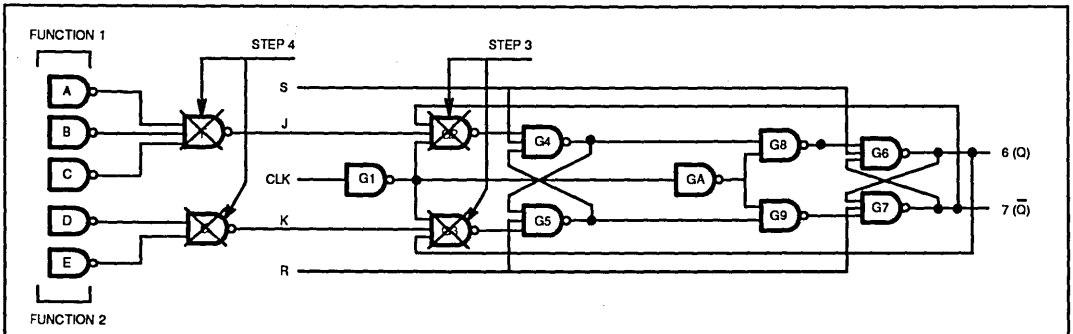


Figure 7–4(a). A Merged J-K Flip-Flop

It is hoped that the reader can, in general, avoid the use of the J – K structures built from gates as shown due to their inefficiency of gate usage. There is also a potential timing liability in that the clock path G1 to GA is faster than the data input paths and creates a possible race. However, this example serves to illustrate that a merging process can be applied systematically, with success, to even relatively complicated structures.

Application Note	
Date of Issue	October 1990
Rev. Date	
Programmable Logic Devices	

PML2552
Application Notes, Vol. 1
A simple logic analyzer module using the PML2552 and the PLHS601

SUMMARY

A simple logic analyzer module can be implemented with a PML2552, a PLHS601, and two SRAM devices, with some resources left for other purposes. The module monitors the data flow on the data bus and stores the data in the SRAMs. Upon detection of a predetermined data pattern, it captures an additional 255 bytes of data before generating an Interrupt Request to the processor. The processor can read on the data bus the SRAM address where the last data byte is stored. That address gives the processor the starting point to go back and retrieve all 4K-bytes of SRAM data both before and after the triggering data pattern. After acknowledge by the processor, the module is cleared to resume operation. This design can readily be modified to suit many data acquisition applications.

The design is implemented entirely by schematics and compiled with SNAP. It has several purposes:

- To show a consistent design methodology for multiple-PLD designs: SNAP easily handles a hierarchical bottom-up design.
- To show a multiple-PLD design: A complete system is implemented with only two higher-density PLDs tied together in a straightforward manner. The SNAP software then elegantly handles all aspect of the design process just as it were for a single-PLD design.
- To show where the PML2552 and PLHS601 fit into your system: The PML2552 in this example incorporates two counters, two latched decoders and random logic. The PLHS601 implements fast, wide-input multiplexers.
- To show a few design hints to take advantage of the PML2552 flexible architecture, such as input address latch, efficient counter implementation, and wide-input multi-level logic.

OPERATION

The two 4KX4 SRAM chips store the 8-bit data stream present on the data bus, in both of the processor's read and write modes. ALE is the Address Latch Enable signal asserted low by the processor to signal a valid address. /PSEN goes high to indicate an access to external memory. Those signals are specific to the 8051-type microcontroller which is used for this example. They should be generated by additional logic for other processors or controllers.

For the 8051-type microcontroller, the lower address byte and the data byte are multiplexed on the same A0–A7/D0–D7 lines. The "Address Latch and Decoder" block (ADDLTCHB) separates the lower byte of the address from the multiplexed address/data bus. The first decoder output is the memory decode (/RAMCE, for RAM Chip Enable) for the 4K-bytes of static RAM at addresses 8XXX when the RAMs are in the read mode. The second output is /INTREC (for Interrupt Received) decoded from a special address – 4000(H) in this example – that the processor writes to, in order to acknowledge receipt of an interrupt from this module. Outputs C4001 and C4002 are used to send the last SRAM address to the data bus. More on this later.

The "Decoder/Sequence Detector" block (DECSEQ) monitors the 8-bit data stream on the data bus. If the predefined data code (F3 in this example) is detected, BREAK goes high to enable clocking of the 8-bit counter which then starts counting up with every external memory access. In the mean time, data on the data bus is still captured and stored in the SRAMs. Clocking of the 8-bit counter is provided by CNTCK1 generated from every external memory access. As long as the 8-bit counter has not reached 255, CNT is low, as determined by the "255 Decode and Latch" (CNT255) block.

With the CNT line still low, the SRAMs are write-only and enabled by every external memory access to store the captured data from the data bus. The SRAMs addresses are provided by the 12-bit counter outputs CT0–CT11 via the 12-bit MUX MUX12. Clocking of the 12-bit counter is also from CNTCK1 and enabled as long as CNT is low.

At the count of 255 by the 8-bit counter, the "255 Decode and Latch" brings /CNT low to stop the 8-bit counter and CNT high to inhibit further data storage by the SRAMs. INT (for Interrupt) also goes high with CNT to interrupt the processor. The SRAMs then become read-only and from now on can be accessed at addresses 8XXX. The 12-bit counter also stops to latch the last SRAM address accessed. With CNT high, the 12 address inputs of the SRAM are now connected to the processor's A0–A11 address lines by MUX12. The SRAM chip enable is decoded by the "Address Decode and Latch" block (ADDLTCHB) so the RAMs can be accessed at addresses 8XXX.

The SRAM address which contains the last data byte is the output (12 bits) of the 12-bit counter when CNT goes high. The processor can read the lower address byte on the data bus D0–D7 at memory address 4001(H) and the higher address byte at memory address 4002(H). The ADDLTCHB block decodes the address lines to direct the 8-bit multiplexer MUX8 to send either the lower or the higher address byte on the data bus. The processor can use the last SRAM address as a starting point to retrieve all 4K-bytes of data stored in the SRAM before and after the riggering data pattern (F3).

The two multiplexers are implemented in the PLHS601.

The processor acknowledges the interrupt by writing a special data byte F4(H) to address 4000(H). The "Address Latch and Decoder" block (ADDLTCHB) decodes the 4000(H) address and brings /INTREC (for Interrupt Received) low to enable the F4(H) data decode. The "Decoder/Sequence Detector" decodes the data byte (F4), asserts /INTACK low (for Interrupt Acknowledge) to clear all the latches and counters (/LCLR low). Normal operation can then resume.

A simple logic analyzer module using the PLM2552 and the PLHS601

DISCUSSION

Overall schematic

Figure 1 shows the block diagram of the complete design. The several blocks shown are implemented either in just two PMLs: a PML2552 and a PLHS601.

The top-level schematic of the complete module is shown in Figure 2. It depicts the two PMLs as sub-blocks of the overall design. The block labeled PML2552B represents the PML2552 as seen from the top level. The block labeled HS601 represents the PLHS601. PML2552B is actually a complete SNAP design with several sub-blocks connected together and compiled. Similarly, HS601 is a complete SNAP design. Notice that the connections between the two PMLs are done in the top-level schematic. In case the design was entered by other means than a schematic, the connections at the top-level can be done by editing the overall netlist. Section 12.6 of Signetics SNAP manual discusses this approach in more detail.

PML2552B

As mentioned earlier, this block in the top-level represents the PML2552. Figure 3 shows its detailed schematic with the several sub-blocks. They are the sequence detector DECSEQ (Figure 4), the address-latch block ADDLTCHB (Figure 5), the count-255 decoder CNT255 (Figure 6), the 8-bit counter CNT8BIT (Figure 7), the twelve-bit counter CNT12B (Figure 8), and random logic gates. Each sub-block is a separate SNAP design up to the NETGEN step where SNAP generates the individual .MAC file. SNAP then merges those .MAC files into the overall PML2552B.MAC file to proceed with the compilation. The above discussion shows how SNAP elegantly handles a hierarchical design, independently of the target PLD or PML device. Now let's look at the individual sub-blocks.

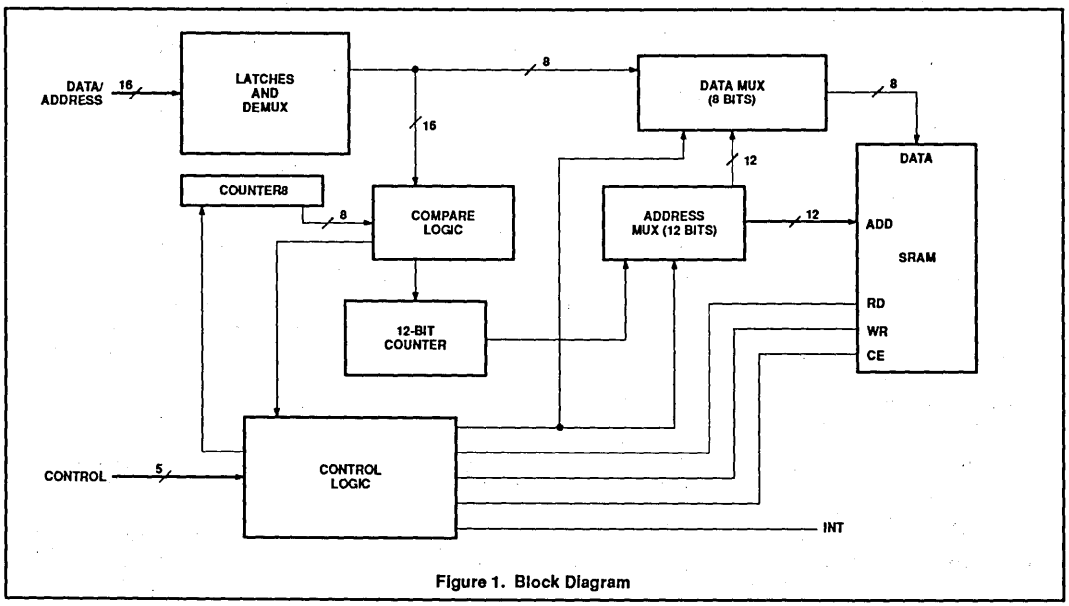


Figure 1. Block Diagram

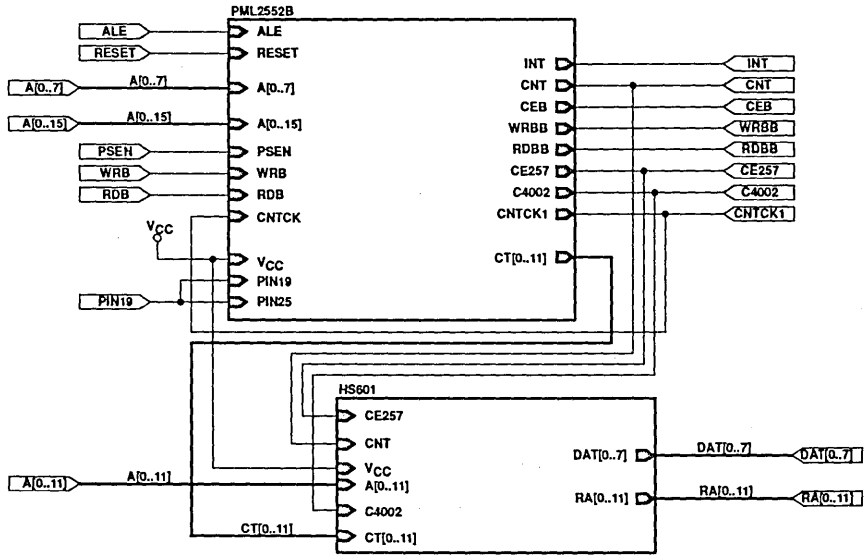


Figure 2. Top-level Schematic of the Logic Analyzer Module, without the SRAMs

A simple logic analyzer module using the PLM2552 and the PLHS601

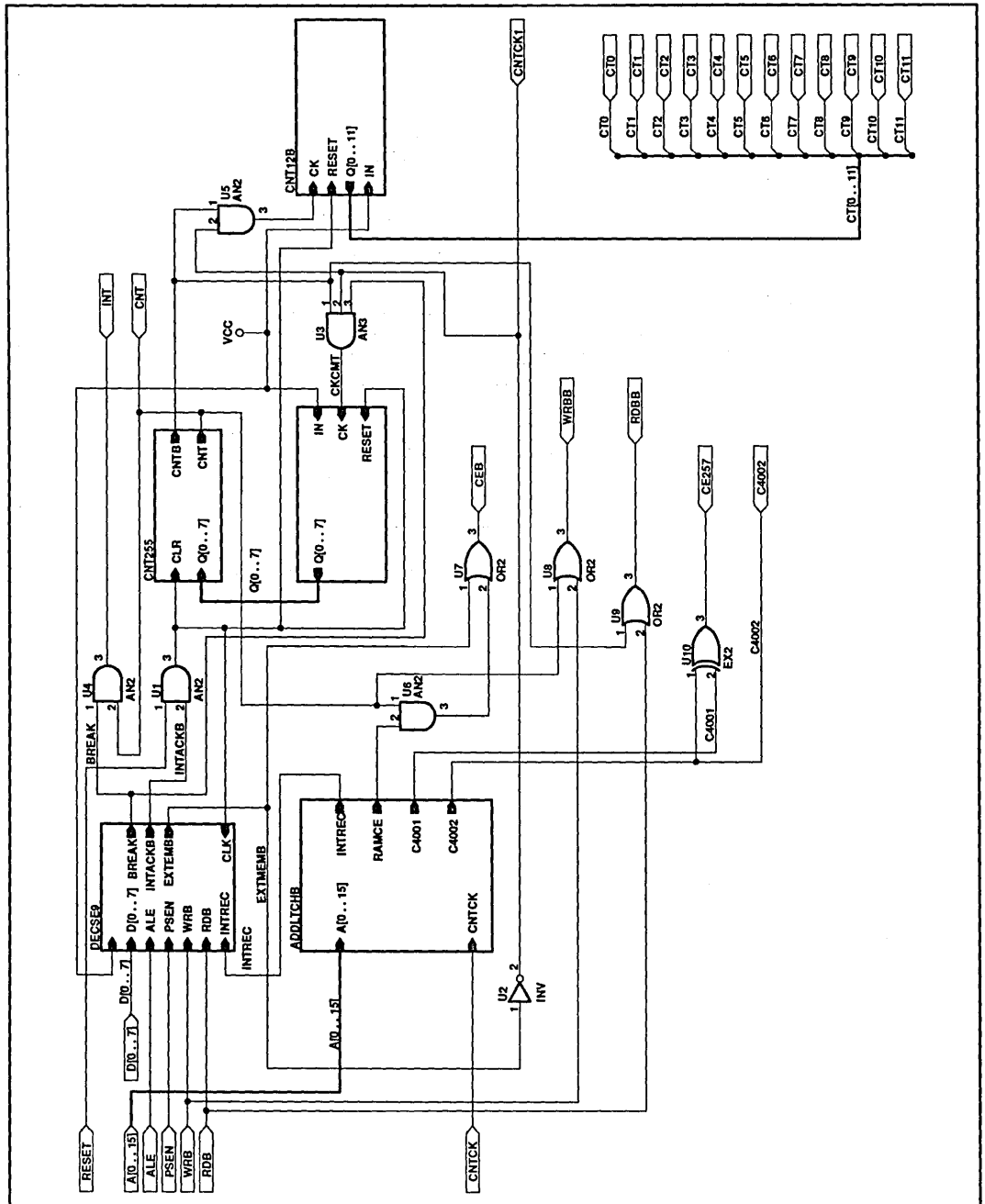


Figure 3. Schematic of PML2552B Block

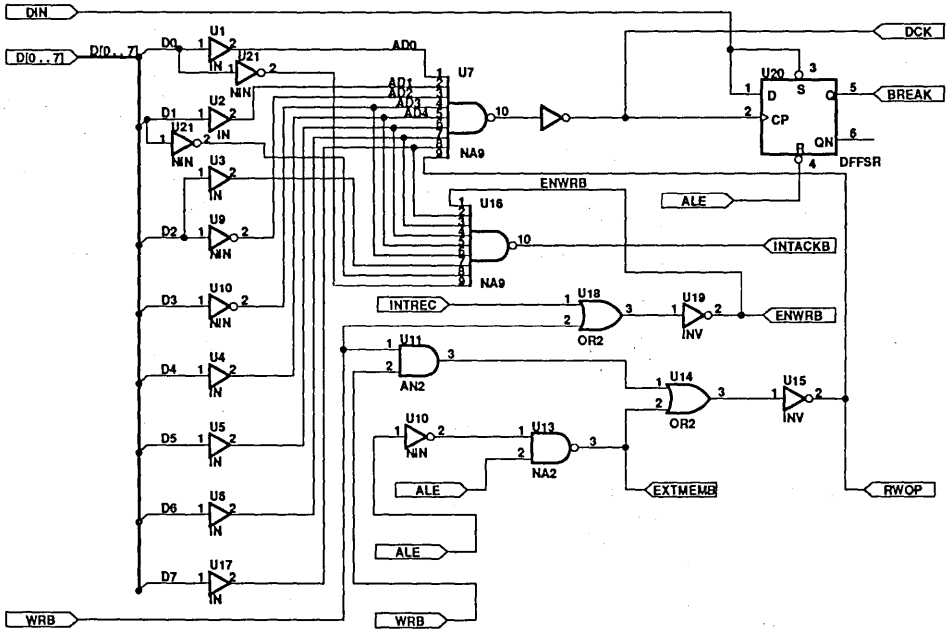


Figure 4. The Sequence Detector DECSEQ

DECSEQ (Figure 4)

The sequence detector is basically a decoder. A close look at its schematic reveals two important points of interest: I/O buffers and wide-input NAND gates.

In the schematic, one will notice that symbols IN and NIN are tied directly to the inputs D[0..7]. In the SNAP symbol library, they represent the non-inverting and inverting input buffers of the actual PML2552 device. They are additional circuitries necessary to maintain appropriate levels of voltage and current. If the above symbols were not entered in the schematic, SNAP would still incorporate them in the FuseMapc operation. Since the first-pass logic minimization takes place before the device-mapping process, a pure schematic diagram without the I/O buffers to distinguish between the peripheral and internal logic could create erroneous results. It is advisable then, to make a practice of placing I/O buffers right before or after the intended I/O pins. Some examples of these buffers are:

IN	direct input
NIN	inverted input
OUT	direct output
NOUT	inverted output
TOUT	3-State output

U7 and U16 in the schematic are 9-input NAND gates. They are easily implemented in Signetics PML devices with a single foldback NAND gate. In general, thanks to its wide-input NAND gates, a PML device can implement virtually any logic function with no more than two gate levels, allowing shorter propagation delays.

A simple logic analyzer module using the PLM2552 and the PLHS601

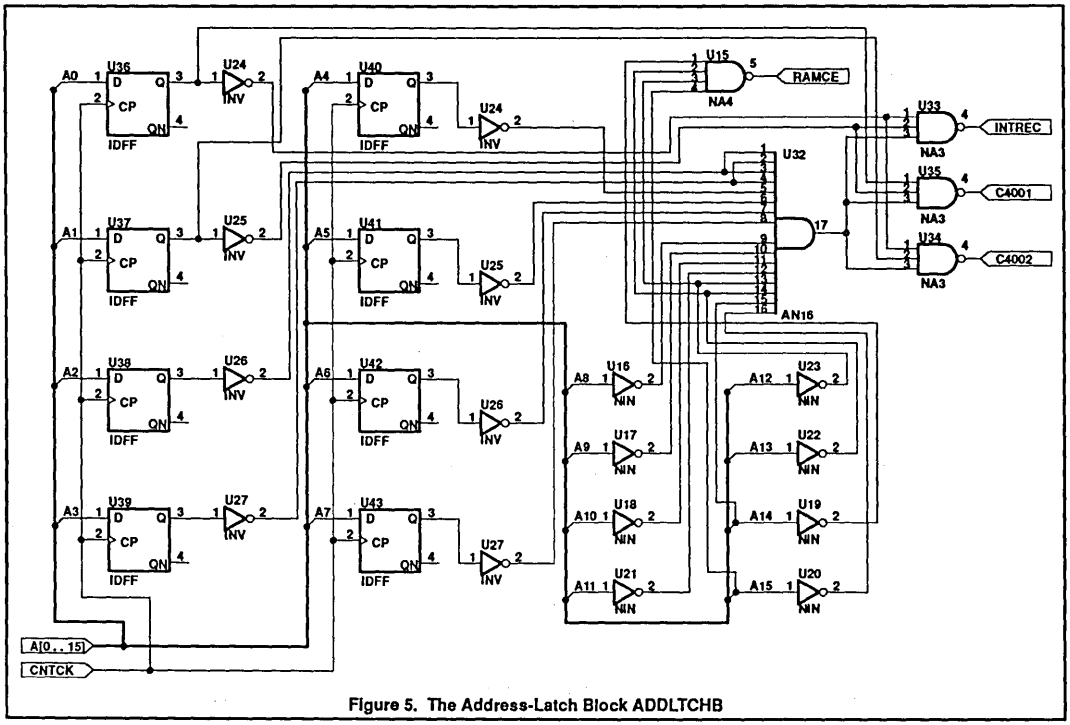


Figure 5. The Address-Latch Block ADDLTCHB

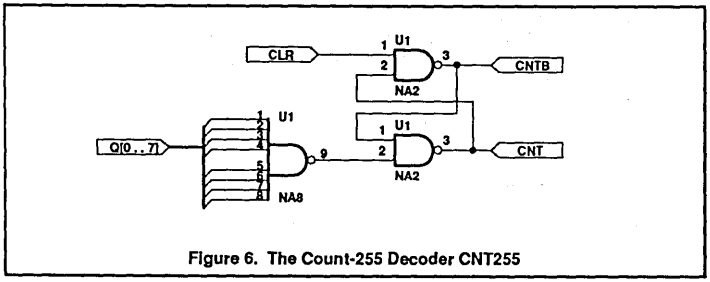


Figure 6. The Count-255 Decoder CNT255

ADDLTCHB (Figure 5)

This block reveals an extremely useful feature of the PML2552 architecture which is the (by-passable) input register. In cases where address and data are multiplexed on the same bus, such as for the 8051-type microcontroller, such input registers are naturally suitable for the demultiplexing task. If one wishes to use the corresponding pins as straight inputs, the registers can be by-passed. Again, U32 is a wide-input NAND gate and input buffer symbols are used to denote intended device input pins.

CNT255 (Figure 6)

This block contains a logic gate tied to a simple latch implemented with two foldback NAND gates. This shows how easily one can make up additional storage elements with the PML architecture. Thanks to the total connectivity of the NAND foldback array, all types of flip-flops and latches can be constructed from the NAND gates. SNAP automatically handles the construction for the user with its extensive library of flip-flop symbols.

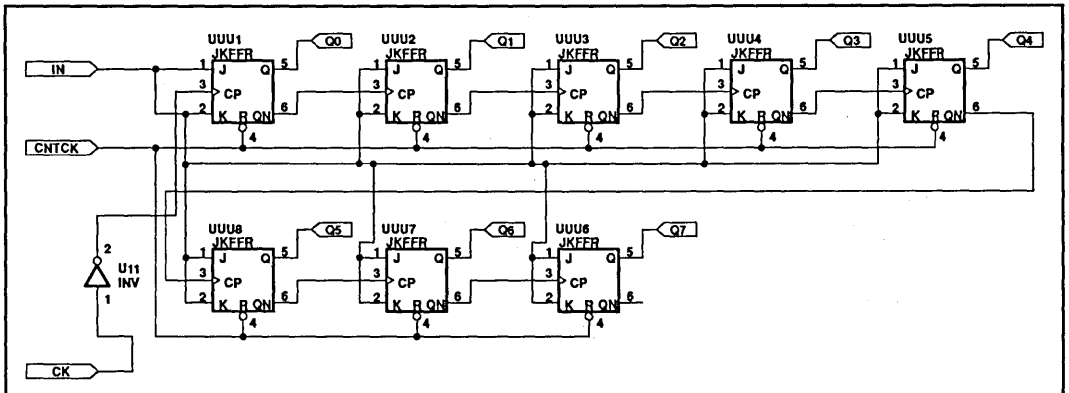


Figure 7. Eight-Bit Counter CNT8BIT

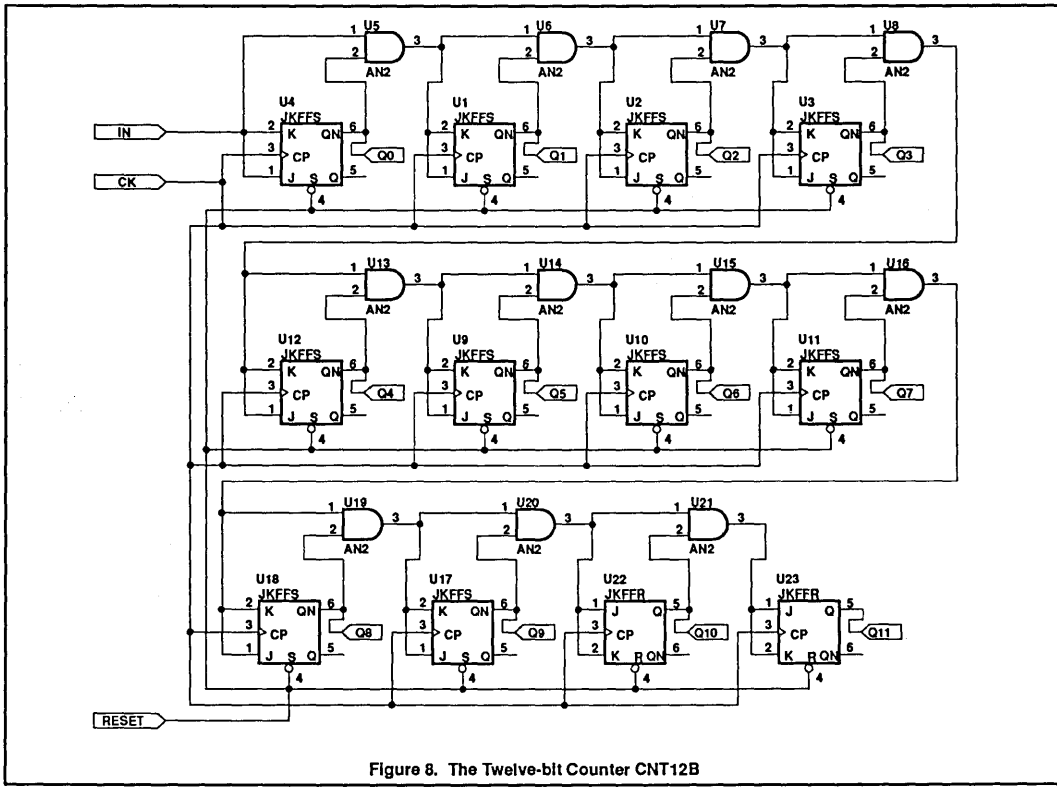


Figure 8. The Twelve-bit Counter CNT12B

A simple logic analyzer module using the PLM2552 and the PLHS601

CNT8BIT (Figure 7)

This block contains an 8-bit ripple counter implemented with 8 internal JK flip-flops. The symbol is pulled from the SNAP library.

The important note relates here to the architecture of the PML2552. The 10 internal JK flip-flops with Reset are individually clockable. Since they are all positive edge-triggered, each flip-flop clock input must be the Q-bar output of the previous stage to implement a ripple counter.

CNT12B (Figure 8)

This block shows a 12-bit synchronous counter implemented with 12 internal JK flip-flops of the PML2552. There are two important reasons why the 12-bit counter is implemented as shown.

First, the JK flip-flop with Preset (JKFFS) all have a common clock. Also, the total connectivity of foldback NAND gates make synchronous counters naturally fit into the PML2552 architecture.

Secondly, a close look Figure 8 shows that ten of the flip-flops in the counter are JKFFS (JK flip-flop with Preset or Set) and two of them are JKFFR (JK flip-flops with Clear or Reset). One would notice that the last two stages of the counter are implemented the conventional way. However, the first ten flip-flops have J-K, Q-QN reversed and Set tied to the common Reset line. To understand, one just have to remember the duality between J and K, Q and QB, Set and Reset. By swapping between J and K, Q and QB, Set and Reset, one can use the JKFFP just like a JKFFR.

HS601 (Figure 9)

This block contains two fast wide-input multiplexers. In this example, the PLHS601 handles the steering of two sets of 12-bit data (CT[0..11] and A[0..11]) into output sets RA[0..11] and DAT[0..7]. This is the type of applications where the PLHS601 is well suited for, i.e. handling wide data busses. This exapmle also illustrates how to implement a hierarchical, bottom-up design with SNAP.

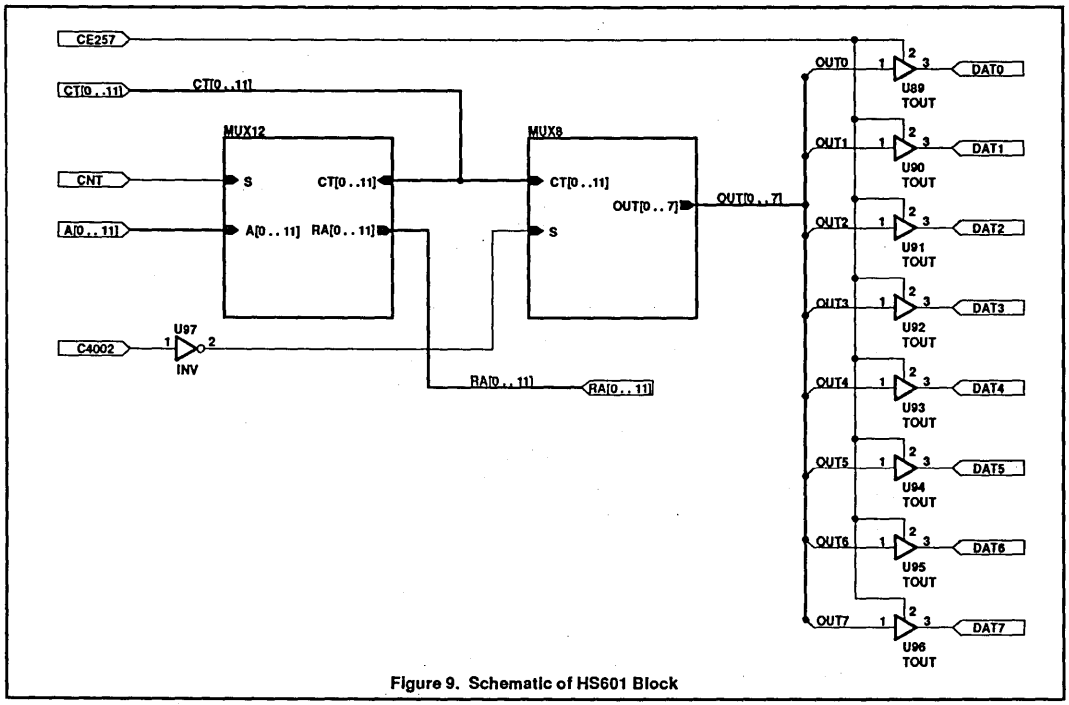


Figure 9. Schematic of HS601 Block

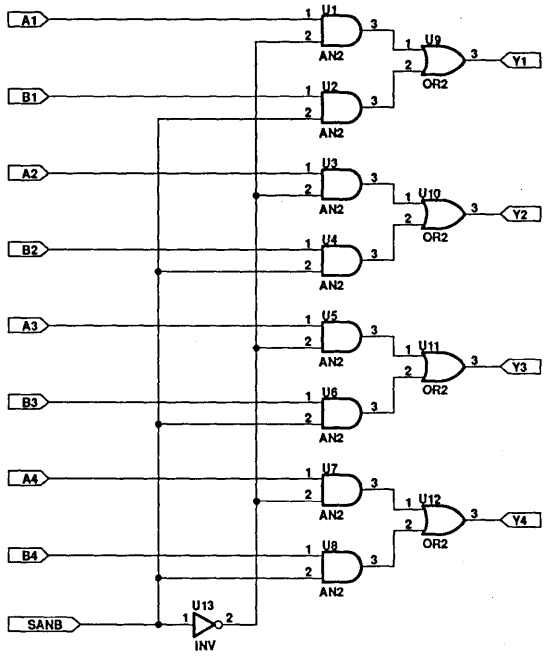


Figure 10. Schematic of 4-bit Multiplexer

MUX4 (Figure 10)

The schematic for the simple four-bit multiplexer is shown. Its nine-input logic is easily implemented with the wide-input foldback NAND gates of the PLHS601. SNAP transforms the AND-OR logic chain shown into a NAND-NAND chain in the compilation process.

MUX8 (Figure 11) and MUX12 (Figure 12)

The eight-bit and twelve-bit multiplexers are implemented from MUX4 sub-blocks. This shows that SNAP can readily deal with hierarchical designs. SNAP takes the MUX4.MAC file, merges two or three of them according to the connections in Figures 11 and 12 to create MUX8.MAC and MUX12.MAC.

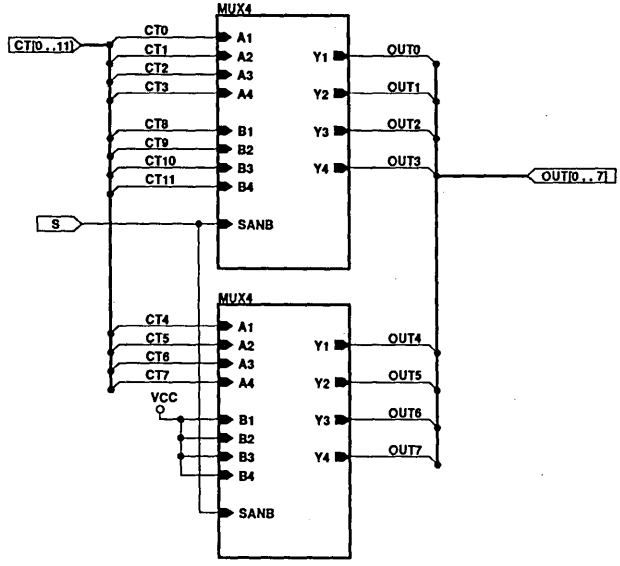


Figure 11. Schematic of 8-bit Multiplexer Built from Two 4-bit Multiplexers

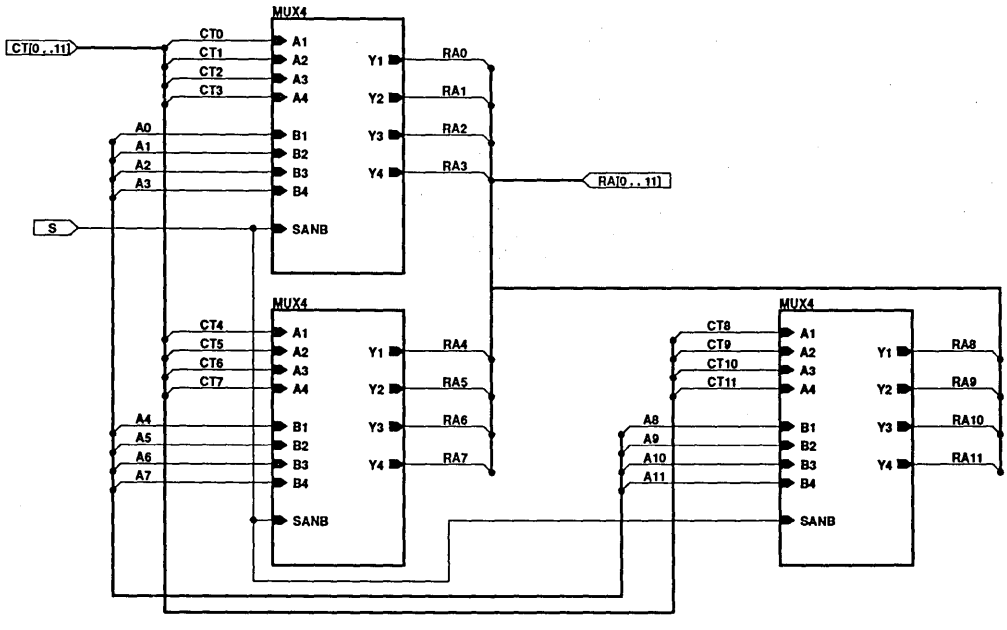


Figure 12. Schematic of 12-bit Multiplexer

A simple logic analyzer module using the PLM2552 and the PLHS601

SIMULATION

The simulation control file (.SCL) for the complete design is shown in Figure 13. Since we do not intend to cover the syntax of the SCL file here, the interested reader should consult the SNAP manual for more details. The SCL file is described below.

Structure of the simulation control file

This SCL file is written mainly as a set of subroutines, each of them performing a determined task, such as reading in data, writing out data, adding a delay, etc. That way, repetitive tasks are handled concisely. Adding or deleting tasks can be done just by commenting out the appropriate subroutine calls.

Description of the SCL file

The simulation file is set-up to duplicate an actual situation, except that timing is looser in order to concentrate on the logical functionality of the design. The simulation starts out with a system reset, then goes on with the detection of F3, the triggering pattern on the data bus. The next 255 read/write operations are performed, then the address of the last byte is sent out on the data bus in two consecutive bytes. Finally, an interrupt acknowledge from the system controller is generated.

Handling Pins 19 and 25 for the PML2552, and V_{CC} in Multiple-PLD Designs

For the PML2552, pin 18 (SCM), pin 19 (SCI) are used to handle the scan mode. Pin 25 (PD) enables or disables to power-down mode. In a typical application, if neither scan mode nor power-down is desired, pins 19 and 25 are tied to ground. Since SNAP 1.6 does not automatically handle the grounding of a signal, pins 19 and 25 are brought out to a separate input (see Figure 2). Actual grounding of that input is done in the SCL file. Similarly, in the case of multiple-PLD design, V_{CC} is not automatically transferred from the lower level of the single PLD to the top level. It is therefore brought out as another dummy input (Figure 2) and tied to the power node. It is also set to high permanently in the SCL file.

```
************************************************
*     Output of Waveform Version 1.40          *
* Date: 05/14/90                 Time: 14:31:41 *
************************************************
*                                              *
* Input File Name   : PML2552B.SCL             *
* Rule File Name    : Scl Rule                 *
* Output File Name  : PML2552B.SCL             *
*                                              *
************************************************
 P A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15,
#   ALE, RESET, RDB, WRB, PSEN, CNTCK1,
#   INT, CEB, CE257, CNT, C4002,
#   WRBB, RDBB, CT0, CT1, CT2, CT3, CT4, CT5, CT6, CT7, CT8, CT9, CT10,
#   CT11, RA0, RA1, RA2, RA3, RA4, RA5, RA6, RA7, RA8, RA9, RA10, RA11,
#   DAT0, DAT1, DAT2, DAT3, DAT4, DAT5, DAT6, DAT7
 PCO
*
* INITIAL SET UP
*
 ST 0 (PIN19)
 ST 1 (VCC)
*
* RESET ROUTINE
*
 SUB RST
 PC 'RESET'
 S 1 (20,60) RESET
* SU TIME=*+100
 END
*
* NORMAL WRITE ROUTINE
*
 SUB NWRITE
 PC 'NORMAL WRITE'
 S 0 (180, 260, 280, 840) A0, A1, A3, A10
 S 1 (180, 260) A4, A5
 S 0 (180, 840) A2, A6, A7, A15
 ST 0 (A8, A9, A12, A13, A14)
 ST 1 (A11)
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (300, 800) WRB
 ST 1 (RDB)
 SU TIME=*+1000
 END
*
* NORMAL READ ROUTINE
*
 SUB NREAD
 PC 'NORMAL READ ROUTINE'
 S 1 (180, 420, 620, 840) A0, A1, A2, A3
 S 0 (180, 840) A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) RDB
 ST 1 (WRB)
 SU TIME=*+1000
 END
*
```

Figure 13. Simulation Control File

```
* READ ROUTINE WHEN DATA = F3 DETECTED
*
 SUB F3READ
 PC 'F3 READ ROUTINE'
 S 0 (180, 420, 720, 840) A0, A1, A4, A5, A6, A7
 S 0 (180, 840) A8, A9, A10, A11
 S 1 (180, 840) A2, A3, A12, A13, A14, A15
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) RDB
 ST 1 (WRB)
 SU TIME=*+1000
 END
*
* SUBROUTINE TO ACKNOWLEDGE INTERRUPT
*
 SUB INTACK
 PC 'INTERRUPT ACKNOWLEDGE'
 S 1 (180, 260, 280, 840) A0, A1, A3
 S 0 (280, 840) A5, A6, A7
 S 1 (180, 260, 840) A2, A4
 S 1 (180, 840) A8, A9, A10, A11, A12, A13, A15
 S 0 (180, 840) A14
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) WRB
 ST 1 (RDB)
 SU TIME=*+1000
 END
*
* SUBROUTINE TO EXECUTE NORMAL READ AND WRITE 128 TIMES
*
 SUB NRW128
 PC 'READ AND WRITE 128 TIMES'
 SETV COUNT, 1
AGAIN
 CALL NREAD
 CALL NWRITE
 INCR COUNT, 1
 IFV COUNT<128 GT AGAIN
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) WRB
 ST 1 (RDB)
 END
*
* SUBROUTINE TO CREATE A DELAY
*
 SUB DELAY
 PC 'SUB FOR 180NS DELAY'
 SU TIME=*+180
 END
*
```

Figure 13. Simulation Control File (Continued)

```
* SUBROUTINE TO USE THE SETV AND DECV COMMANDS
* TO SEND A KNOWN DATA PATTERN ON A BUS
*
 SUB NBUS
 PC 'SUB TO HANDLE BUS'
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) WRB
 ST 1 (RDB)
 CALL DELAY
 SETV VAR1, 243
 DECV VAR1 (A7, A6, A5, A4, A3, A2, A1, A0)
 SETV VAR3, 1
 DECV VAR3 (A15, A14, A13, A12, A11, A10, A9, A8)
 SU TIME=*+1000
 END
*
* SUBROUTINE TO READ THE LOWER BYTE OF THE LAST SRAM
* ADDRESS. DAT0-7 SHOULD REFLECT THE LOWER BYTE OF THE
* 12-BIT COUNTER CT0-CT7
*
 SUB R4001
 PC 'SUB TO READ LOWER ADDRESS BYTE'
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) RDB
 ST 1 (WRB)
 CALL DELAY
 SETV VAR1, 1
 DECV VAR1 (A7, A6, A5, A4, A3, A2, A1, A0)
 SETV VAR3, 64
 DECV VAR3 (A15, A14, A13, A12, A11, A10, A9, A8)
 SU TIME=*+1000
 END
*
* SUBROUTINE TO READ THE HIGHER BYTE OF THE LAST SRAM
* ADDRESS. DAT0-7 SHOULD REFLECT THE HIGHER BYTE OF THE
* 12-BIT COUNTER CT8-CT11
*
 SUB R4002
 PC 'SUB TO READ LOWER ADDRESS BYTE'
 S 0 (60, 200, 860) ALE
 S 0 (60) PSEN
 S 1 (400, 800) RDB
 ST 1 (WRB)
 CALL DELAY
 SETV VAR1, 2
 DECV VAR1 (A7, A6, A5, A4, A3, A2, A1, A0)
 SETV VAR3, 64
 DECV VAR3 (A15, A14, A13, A12, A11, A10, A9, A8)
 SU TIME=*+1000
 END
*
```

Figure 13. Simulation Control File (Continued)

```
**************************
*
* START OF MAIN PROGRAM
*
**************************
* SYSTEM RESET
*
 CALL RST
*
* PERFORM A NORMAL WRITE
*
* CALL NWRITE
*
* PERFORM A NORMAL READ
*
 CALL NREAD
*
* PERFORM A BUS HANDLING
*
* CALL NBUS
*
* DELAY BY 180NS
*
 CALL DELAY
*
* PERFORM AN F3 READ
*
 CALL F3READ
*
* PERFORM ANOTHER 257 R/W OPERATIONS
*
 CALL NRW128
 CALL NREAD
 CALL NWRITE
 CALL R4001
 CALL R4002
*
* PERFORM INTERRUPT ACKNOWLEDGE
*
 CALL INTACK
*
 F
```

Figure 13. Simulation Control File (Continued)

```
Design from PML2552B.N2 - Created on Fri Jun 22, 1990  3:19PM
Device PML2552
   Cell name    used/total      %
========================================
     CKDIN552      0 /   4     0%
     CKNIN552      0 /   4     0%
       FBNAND     59 /  96    61%
         NAND     96 / 104    92%
       DIN552     11 /  25    44%
       NIN552     11 /  25    44%
      CDIN552      3 /   4    75%
      CNIN552      1 /   4    25%
        CK552      1 /   4    25%
      IDFF552      8 /  16    50%
      BDIN552      0 /  24     0%
      BNIN552      0 /  24     0%
      JKCL552     10 /  10   100%
      JKPR552     10 /  10   100%
      EXOR552      8 /   8   100%
      TOUT552     20 /  24    83%
      ODFF552     12 /  16    83%
```

Figure 14a. PML2552 Resources Summary

```
Device = PML2552
Pin1   = CT5
Pin2   = CT4
Pin3   = CT3
Pin5   = CT2
Pin6   = CT1
Pin7   = CT0
Pin8   = RESET
Pin9   = D7
Pin10  = D6
Pin11  = D5
Pin12  = D4
Pin13  = D3
Pin14  = D2
Pin16  = D1
Pin17  = D0
Pin20  = ALE
Pin22  = PSEN
Pin23  = WRB
Pin24  = RDB
Pin26  = A15
Pin28  = A14
Pin29  = A13
Pin30  = A12
Pin31  = A11
Pin33  = A10
Pin34  = A9
Pin35  = A8
Pin36  = CNTCK
Pin37  = A7
Pin39  = A6
Pin40  = A5
Pin41  = A4
Pin42  = A3
Pin43  = A2
Pin44  = A1
Pin45  = A0
Pin46  = CNTCK1
Pin47  = CEB
Pin48  = WRBB
Pin50  = RDBB
Pin51  = C4002
Pin52  = CE257
Pin53  = CNT
Pin54  = INT
Pin61  = CT11
Pin62  = CT10
Pin63  = CT9
Pin64  = CT8
Pin67  = CT7
Pin68  = CT6
```

Figure 14b. PML2552 Pin Assignment

```
Design from HS601.N2 - Created on Mon Aug 20, 1990  5:35PM
Device PLHS601
   Cell name    used/total      %
======================================
       DIN601     27 /  40    67%
       NIN601      2 /  40     5%
       FBNAND     56 /  78    71%
         NAND     40 /  64    62%
       EXO601      0 /   8     0%
       NOU601     20 /  24    83%
```

Figure 15a. PLHS601 Resources Summary

```
Device = PLHS601
Pin1   = A1
Pin2   = A2
Pin4   = A3
Pin5   = A4
Pin6   = A5
Pin7   = A6
Pin8   = A7
Pin10  = A8
Pin11  = A9
Pin13  = A10
Pin14  = A11
Pin15  = CNT
Pin17  = C4002
Pin18  = CE257
Pin21  = DAT0
Pin22  = DAT1
Pin23  = DAT2
Pin24  = DAT3
Pin25  = DAT4
Pin27  = DAT5
Pin28  = DAT6
Pin30  = DAT7
Pin36  = RA0
Pin38  = RA1
Pin39  = RA2
Pin40  = RA3
Pin41  = RA4
Pin42  = RA5
Pin44  = RA6
Pin45  = RA7
Pin47  = RA8
Pin48  = RA9
Pin49  = RA10
Pin51  = RA11
Pin52  = CT0
Pin53  = CT1
Pin55  = CT2
Pin56  = CT3
Pin57  = CT4
Pin58  = CT5
Pin59  = CT6
Pin61  = CT7
Pin62  = CT8
Pin64  = CT9
Pin65  = CT10
Pin66  = CT11
Pin68  = A0
```

Figure 15b. PLHS601 Pin Assignment

CONCLUSION

A simple 8-bit logic analyzer has been implemented with only two PML devices and two SRAMs. It performs all the required data handling, as well as the interface to the system controller. This examples shows that SNAP can easily handle hierarchical, multiple PLD designs. Several useful design hints relevant to the intended application areas of the PML2552 and PLHS601 and considerations related about their architectures are also presented. Figures 14 and 15 depict the resource usage summaries and pin assignments for those devices. There are plenty of logic and pins remaining to implement additional features. This design can readily be modified to become a data acquisition system.

Section 10
Package Outlines

INDEX

Package Outlines

PLCC

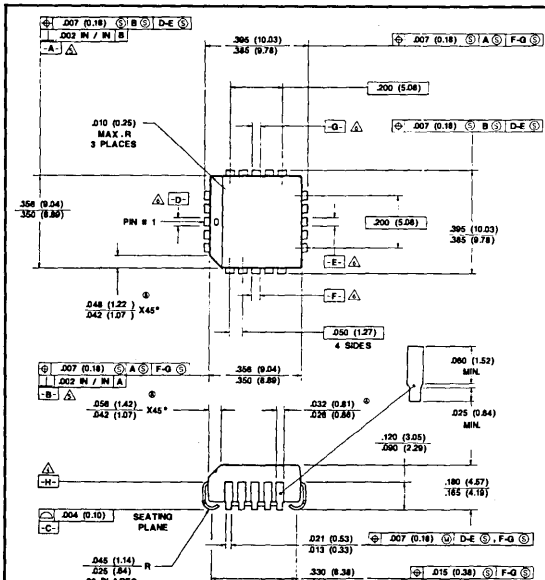
1. Package dimensions conform to JEDEC specifications for standard Plastic Leaded Chip Carrier outline (PLCC) package.
2. Controlling dimensions are given in inches with dimensions in millimeters contained in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M – 1982.
4. "D–E" and "F–G" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm (0.006") on any side.
5. Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
6. Lead material: Olin 194 (Copper Alloy) or equivalent, solder dipped.
7. Body material: Plastic (Epoxy).
8. Thermal resistance values are determined by Temperature Sensitive Parameter (TSP) method. This method uses the forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test conditions for these values are:

   Test Ambient—Still Air
   Test Fixture— θ_{JA}—Glass epoxy test board (2.24" × 2.24" × 0.062")
   θ_{JC}—Water cooled heat sink

PLASTIC LEADED CHIP CARRIER (PLCC)

NO. OF LEADS	PACKAGE CODE	DESCRIPTION	TYPICAL θ_{JA}/θ_{JC} VALUES (°C/W)	
			Average θ_{JA}	Average θ_{JC}
20	A	350mil-wide	70	30
28	A	450mil-wide	61	26
52	A	750mil-wide	42	14
68	A	950mil-wide	42	14

20-PIN PLASTIC LEADED CHIP CARRIER



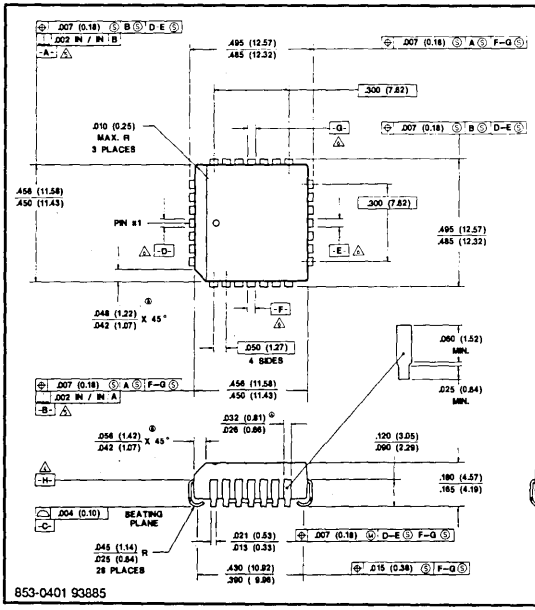
NOTES:

1. Package dimensions conform to JEDEC specification MO-047-AA for plastic leaded chip carrier 20 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "H" located at the top of mold parting line and coincident with top of lead. Where lead exists plastic body.
5. Location to datum "A" and "B" to be determined at plane "H".These datums do not include mold flash. Mold flash protrusion shall not exceed .006" (0.15mm) on any side.
6. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
7. Pin numbers continue counterclockwise to pin #20 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.

853-0400 93885 AL1

Package Outlines

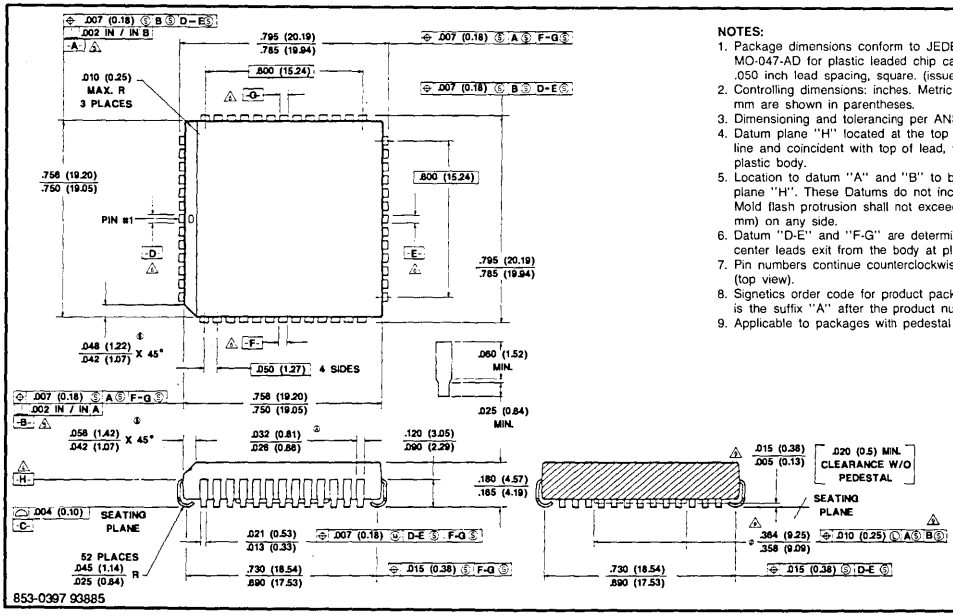
28-PIN PLASTIC LEADED CHIP CARRIER



NOTES:
1. Package dimensions conform to JEDEC specification MO-047-AB for plastic leaded chip carrier 28 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "H" located at the top of mold parting line and coincident with top of lead. Where lead exits plastic body.
5. Location to datum "A" and "B" to be determined at plane "H". These Datums do not include mold flash. Mold flash protrusion shall not exceed .006" (0.15 mm) on any side.
6. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
7. Pin numbers continue counterclockwise to pin #28 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.

AQ1

52-PIN PLASTIC LEADED CHIP CARRIER

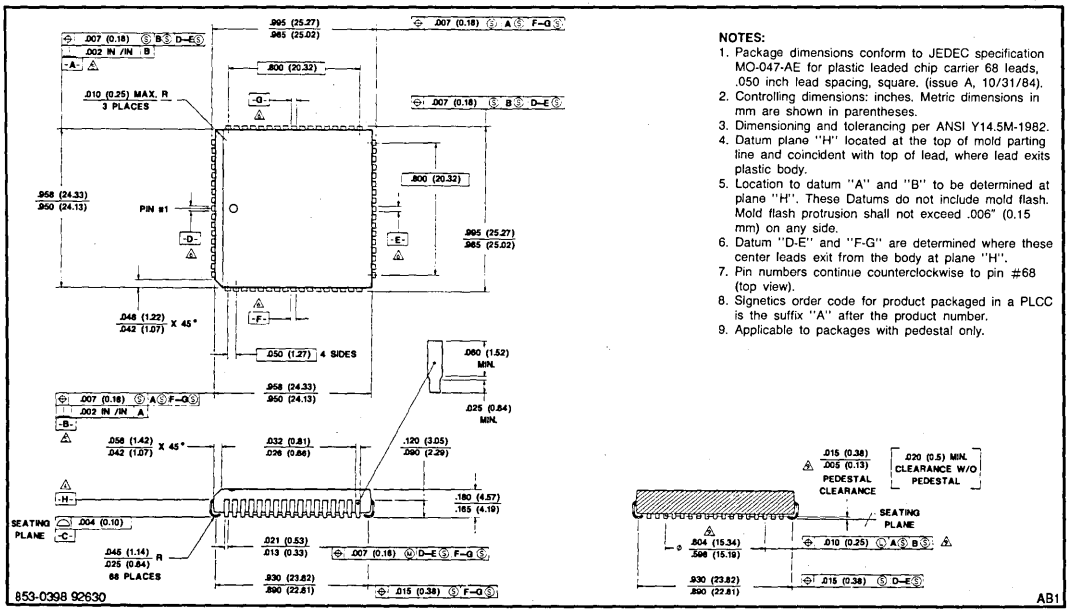


NOTES:
1. Package dimensions conform to JEDEC specification MO-047-AD for plastic leaded chip carrier 52 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "H" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
5. Location to datum "A" and "B" to be determined at plane "H". These Datums do not include mold flash. Mold flash protrusion shall not exceed .006" (0.15 mm) on any side.
6. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
7. Pin numbers continue counterclockwise to pin #52 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.

AA1

Package Outlines

68-PIN PLASTIC LEADED CHIP CARRIER



NOTES:

1. Package dimensions conform to JEDEC specification MO-047-AE for plastic leaded chip carrier 68 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "H" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
5. Location to datum "A" and "B" to be determined at plane "H". These Datums do not include mold flash. Mold flash protrusion shall not exceed .006" (0.15 mm) on any side.
6. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
7. Pin numbers continue counterclockwise to pin #68 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.

Package Outlines

HERMETIC CERDIP

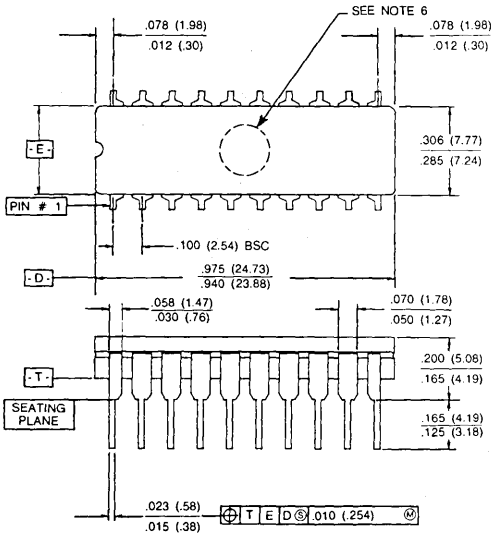
1. Package dimensions conform to JEDEC specifications for standard Ceramic Dual Inline (CERDIP) package.
2. Controlling dimensions are given in inches with dimensions in millimeters, mm, contained in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M – 1982.
4. Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Lead material: ASTM alloy F-30 (Alloy 42) or equivalent – tin plated or solder dipped.
7. Body material: Ceramic with glass seal at leads.
8. Thermal resistance values are determined by Temperature Sensitive Parameter (TSP) method. This method uses the forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test conditions for these values follow:

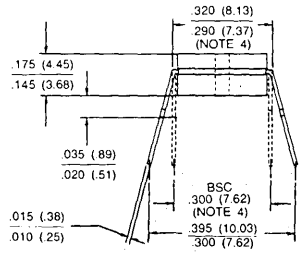
Test Ambient—Still Air
Test Fixture— θ_{JA}– Textool ZIF socket with 0.04" stand-off
θ_{JC}– Water cooled heat sink

CERAMIC DUAL IN-LINE PACKAGES

NO. OF LEADS	PACKAGE CODE	DESCRIPTION	TYPICAL θ_{JA}/θ_{JC} VALUES (°C/W)	
			Average θ_{JA}	Average θ_{JC}
20	F, FA	300mil-wide	72	8
24	F, FA	300mil-wide	62	7
28	F, FA	600mil-wide	45	6

20-PIN CERAMIC DUAL IN-LINE, F PACKAGE (WITH QUARTZ WINDOW, FA PACKAGE)



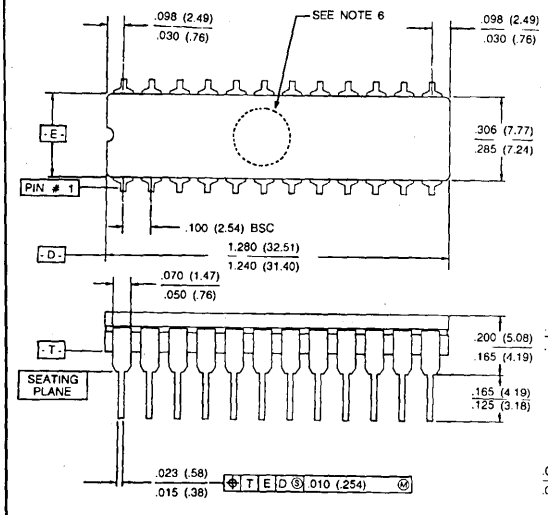


NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #20 when viewed from the top.
6. Denotes window location for EPROM products.

853-0584 88099 FL1

Package Outlines

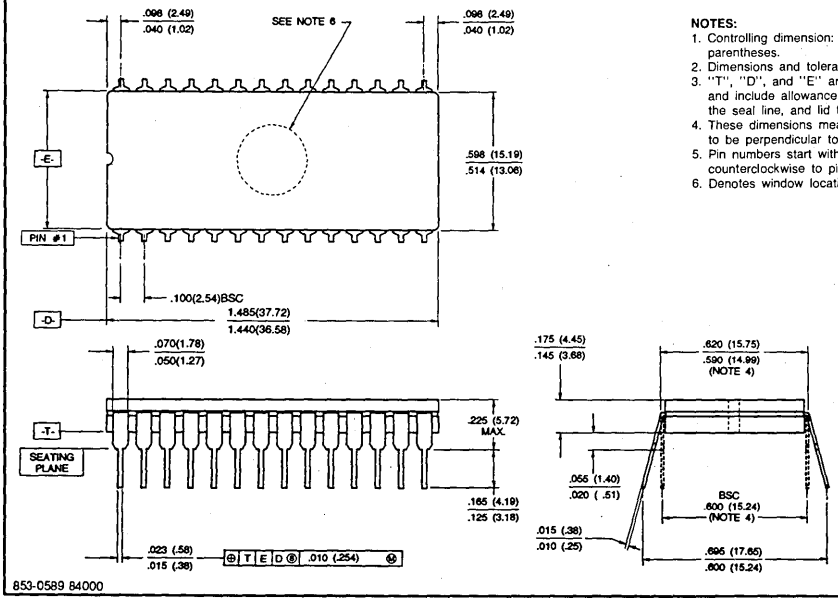
24-PIN CERAMIC DUAL IN-LINE, F PACKAGE



NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M - 1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from the top.
6. Denotes window location for EPROM products.

853-0586 84000 FN1

28-PIN CERAMIC DUAL IN-LINE, F PACKAGE (600mil-wide)

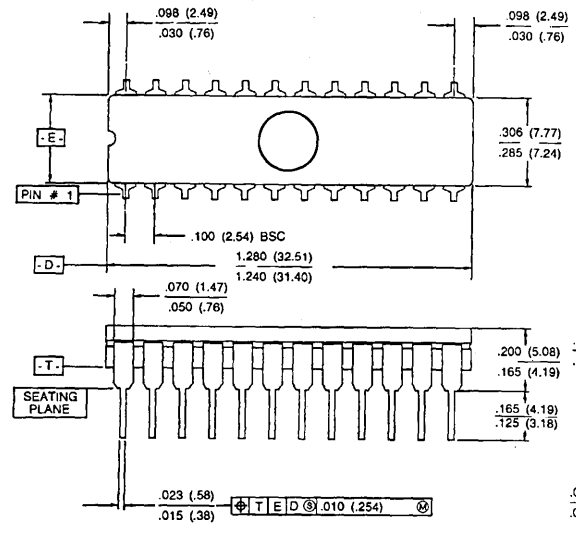


NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M - 1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #28 when viewed from the top.
6. Denotes window location for EPROM products.

853-0589 84000 FQ3

Package Outlines

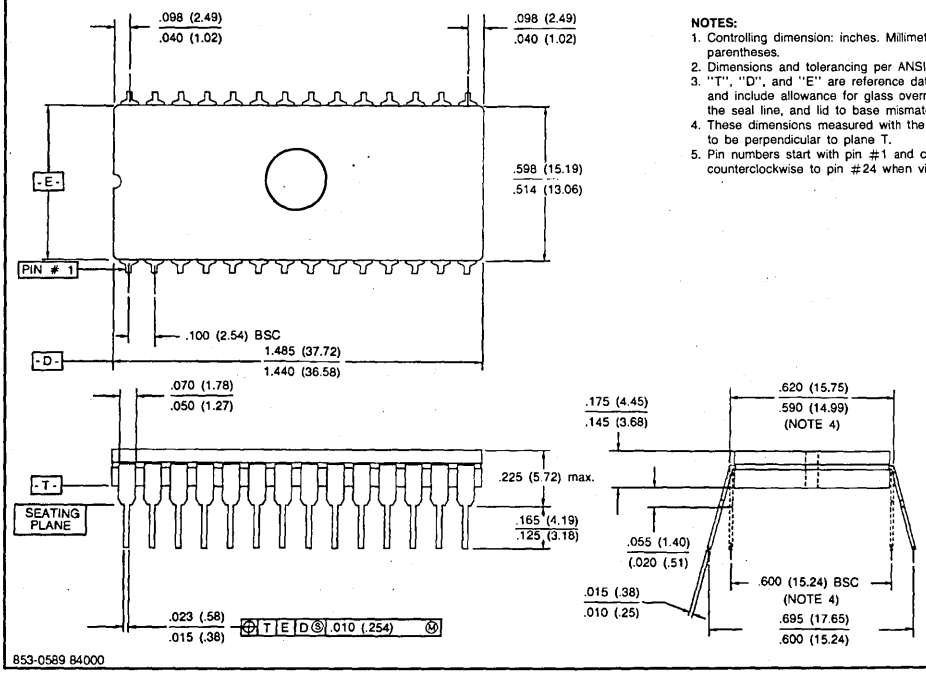
24-PIN CERAMIC DUAL IN-LINE WITH QUARTZ WINDOW, FA PACKAGE



NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M - 1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #20 when viewed from the top.

853-0586 84000 FN1

28-PIN CERAMIC DUAL IN-LINE WITH QUARTZ WINDOW, FA PACKAGE



NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M - 1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from the top.

853-0589 84000 FQ3

Package Outlines

J-LEADED CERQUAD WITH QUARTZ WINDOW

NO. OF LEADS	PACKAGE CODE	DESCRIPTION	TYPICAL θ_{JA}/θ_{JC} VALUES (°C/W)	
			Average θ_{JA}	Average θ_{JC}
68	KA	965mil-wide	44.5[1]	TBD

NOTE:
1. For die size of 55K mils^2, 1W power dissipation, soldered.

68-PIN J-LEADED CERQUAD WITH QUARTZ WINDOW

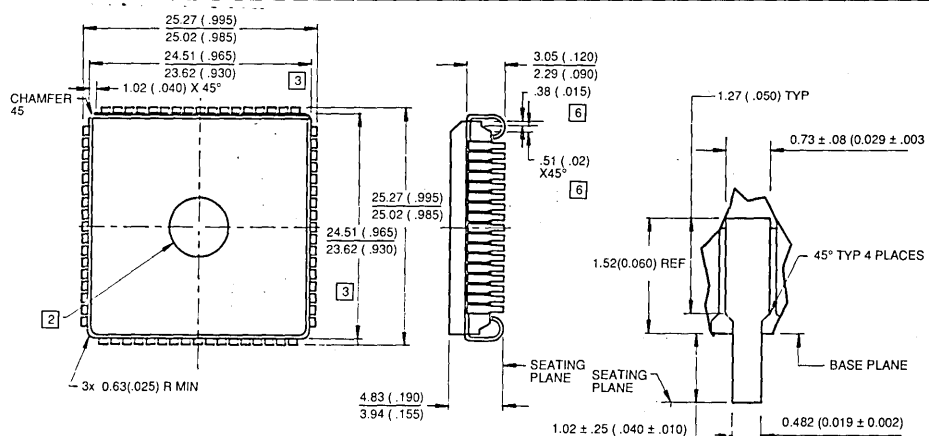


FIGURE 1. PRINCIPAL DIMENSIONS AND DATUMS mm/(inch)

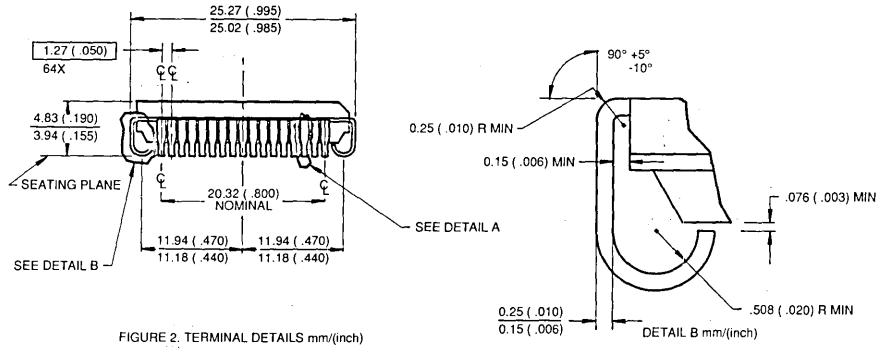


FIGURE 2. TERMINAL DETAILS mm/(inch)

1. ALL DIMENSIONS AND TOLERANCES TO CONFORM TO ANSI Y14.5-1982.
2. UV WINDOW IS OPTIONAL.
3. DIMENSIONS DO NOT INCLUDE GLASS PROTRUSION. GLASS PROTRUSION TO BE 0.005 INCHES MAX ON EACH SIDE.
4. CONTROLLING DIMENSION MILLIMETERS.
5. ALL DIMENSIONS AND TOLERANCES INCLUDE LEAD TRIM OFFSET AND LEAD PLATING FINISH.
6. BACKSIDE SOLDER RELIEF IS OPTIONAL AND DIMENSIONS ARE FOR REFERENCE ONLY.

853-1473 00441 KC1

Package Outlines

PLASTIC DIP

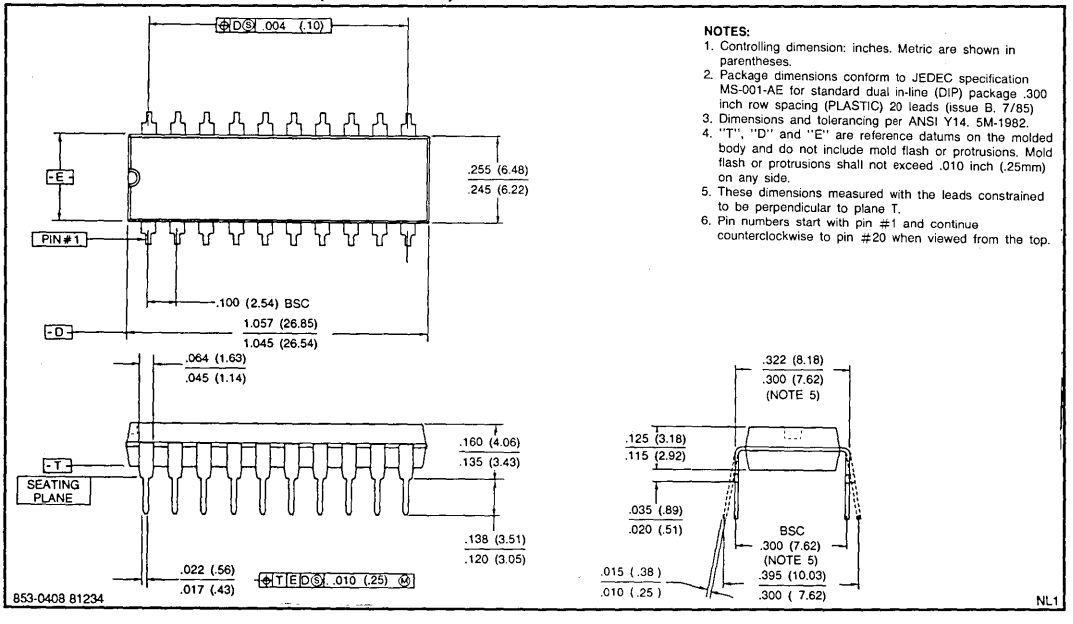
1. Package dimensions conform to JEDEC specification MS–001–AA for standard Plastic Dual Inline (DIP) package.
2. Controlling dimensions are given in inches with dimensions in millimeters, mm, contained in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M – 1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.01 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
7. Lead material: Olin 194 (Copper Alloy) or equivalent, solder dipped.
8. Body material: Plastic (Epoxy).
9. Thermal resistance values are determined by Temperature Sensitive Parameter (TSP) method. This method uses the forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test conditions for these values are:

   Test Ambient—Still Air
   Test Fixture— θ_{JA}– Textool ZIF socket with 0.04" stand-off
   θ_{JC}– Water cooled heat sink

PLASTIC DUAL IN-LINE PACKAGES

NO. OF LEADS	PACKAGE CODE	DESCRIPTION	TYPICAL θ_{JA}/θ_{JC} VALUES (°C/W)	
			Average θ_{JA}	Average θ_{JC}
20	N	Cu. Lead Frame 300mil-wide	63	27
24	N	Cu. Lead Frame 300mil-wide	56	26
28	N	Cu. Lead Frame 600mil-wide	46	18
28	N3	Cu. Lead Frame 300mil-wide	53	24

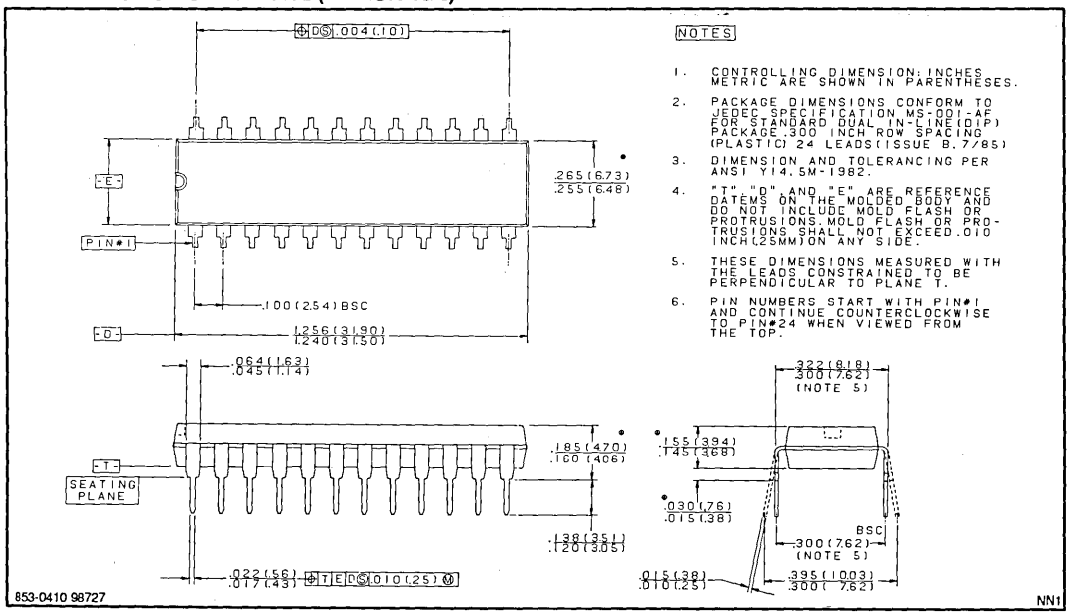
20-PIN PLASTIC DUAL IN-LINE (N PACKAGE)



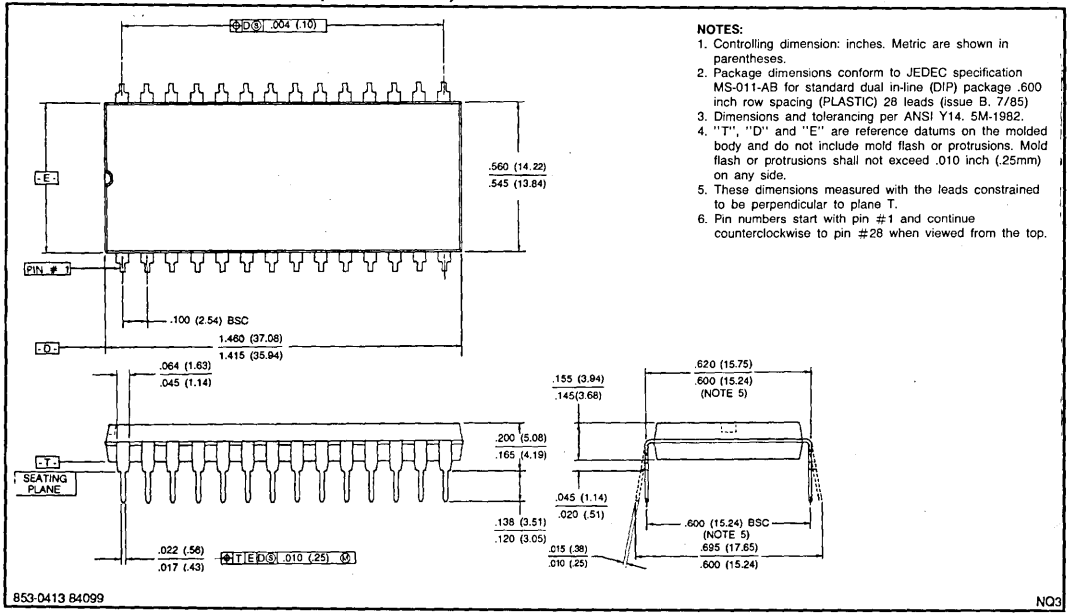
NOTES:
1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC specification MS-001-AE for standard dual in-line (DIP) package .300 inch row spacing (PLASTIC) 20 leads (issue B. 7/85)
3. Dimensions and tolerancing per ANSI Y14. 5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise to pin #20 when viewed from the top.

Package Outlines

24-PIN PLASTIC DUAL IN-LINE (N PACKAGE)

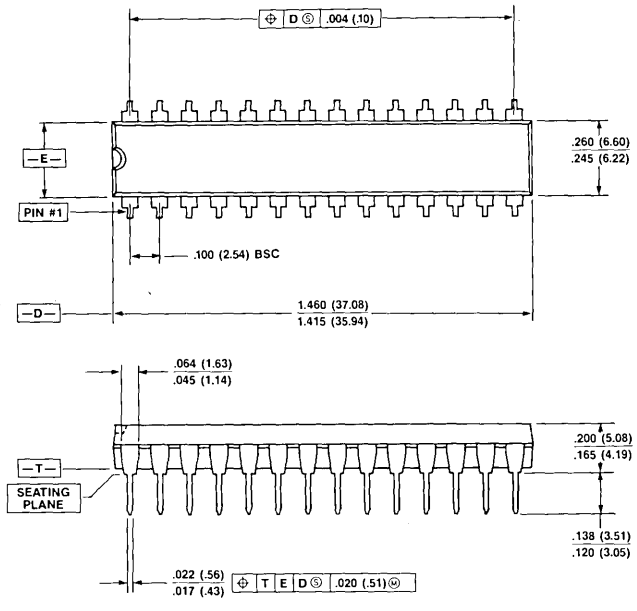


28-PIN PLASTIC DUAL IN-LINE (N PACKAGE)



Package Outlines

28-PIN PLASTIC DUAL IN-LINE (N3 PACKAGE)



NOTES:
1. Controlling dimensions: inches. Metric are shown in parentheses.
2. Dimensions are tolerancing per ANSI Y14.5M-1982.
3. "T", "D", and "E" are reference datums on the molded body and do not include mold-flash or protrusions which shall not exceed .010 inch (.25 mm) on any side.
4. These dimensions measured with the leads constrained to be perpendicular to plane "T".
5. Pin numbers start with #1 and continue counterclockwise to pin #28 when viewed from the top.

853-0864 89398

NQ1

Section 11
Sales Offices, Representatives & Distributors

Sales Offices, Representatives & Distributors

Programmable Logic Devices

SIGNETICS HEADQUARTERS
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, CA 94088-3409
Phone: (408) 991-2000

ALABAMA
Huntsville
Phone: (205) 830-4082

CALIFORNIA
Calabasas
Phone: (818) 880-6304

Irvine
Phone: (714) 833-8980
(714) 752-2780

San Diego
Phone: (619) 560-0242

Sunnyvale
Phone: (408) 991-3737

COLORADO
Englewood
Phone: (303) 792-9011

GEORGIA
Atlanta
Phone: (404) 594-1392

ILLINOIS
Itasca
Phone: (708) 250-0050

INDIANA
Kokomo
Phone: (317) 459-5355

MASSACHUSETTS
Westford
Phone: (508) 692-6211

MICHIGAN
Farmington Hills
Phone: (313) 553-6070

NEW JERSEY
Parsippany
Phone: (201) 334-4405

Toms River
Phone: (201) 505-1200

NEW YORK
Wappingers Falls
Phone: (914) 297-4074

OHIO
Columbus
Phone: (614) 888-7143

OREGON
Beaverton
Phone: (503) 627-0110

PENNSYLVANIA
Plymouth Meeting
Phone: (215) 825-4404

TENNESSEE
Greeneville
Phone: (615) 639-0251

TEXAS
Austin
Phone: (512) 339-9945

Richardson
Phone: (214) 644-1610

CANADA
SIGNETICS CANADA, LTD.
Etobicoke, Ontario
Phone: (416) 626-6676

Nepean, Ontario
Phone: (613) 225-5467

REPRESENTATIVES

ALABAMA
Huntsville
Elcom, Inc.
Phone: (205) 830-4001

ARIZONA
Scottsdale
Thom Luke Sales, Inc.
Phone: (602) 941-1901

CALIFORNIA
Folsom
Webster Associates
Phone: (916) 989-0843

COLORADO
Englewood
Thom Luke Sales, Inc.
Phone: (303) 649-9717

CONNECTICUT
Wallingford
JEBCO
Phone: (203) 265-1318

FLORIDA
Oviedo
Conley and Assoc., Inc.
Phone: (407) 365-3283

GEORGIA
Norcross
Elcom, Inc.
Phone: (404) 447-8200

ILLINOIS
Hoffman Estates
Micro-Tex, Inc.
Phone: (708) 382-3001

INDIANA
Indianapolis
Mohrfield Marketing, Inc.
Phone: (317) 546-6969

IOWA
Cedar Rapids
J.R. Sales
Phone: (319) 393-2232

MARYLAND
Columbia
Third Wave Solutions, Inc.
Phone: (301) 290-5990

MASSACHUSETTS
Chelmsford
JEBCO
Phone: (508) 256-5800

MICHIGAN
Brighton
AP Associates, Inc.
Phone: (313) 229–6550

MINNESOTA
Eden Prairie
High Technology Sales
Phone: (612) 944-7274

MISSOURI
Bridgeton
Centech, Inc.
Phone: (314) 291-4230

Raytown
Centech, Inc.
Phone: (816) 358-8100

NEW MEXICO
Albuquerque
F.P. Sales
Phone: (505) 345-5553

NEW YORK
Ithaca
Bob Dean, Inc.
Phone: (607) 257-1111

Rockville Centre
S-J Associates
Phone: (516) 536-4242

Wappingers Falls
Bob Dean, Inc.
Phone: (914) 297-6406

NORTH CAROLINA
Smithfield
ADI
Phone: (919) 934-8136

OHIO
Aurora
InterActive Technical Sales, Inc.
Phone: (216) 562-2050

Columbus
InterActive Technical Sales, Inc.
Phone: (614) 888-1256

Dayton
InterActive Technical Sales, Inc.
Phone: (513) 436-2230

OREGON
Beaverton
Western Technical Sales
Phone: (503) 644-8860

PENNSYLVANIA
Hatboro
Delta Technical Sales, Inc.
Phone: (215) 957-0600

TEXAS
Austin
Synergistic Sales, Inc.
Phone: (512) 346-2122

Houston
Synergistic Sales, Inc.
Phone: (713) 937-1990

Richardson
Synergistic Sales, Inc.
Phone: (214) 644-3500

UTAH
Salt Lake City
Electrodyne
Phone: (801) 264-8050

WASHINGTON
Bellevue
Western Technical Sales
Phone: (206) 641-3900

Spokane
Western Technical Sales
Phone: (509) 922-7600

WISCONSIN
Waukesha
Micro-Tex, Inc.
Phone: (414) 542-5352

CANADA
Calgary, Alberta
Tech-Trek, Ltd.
Phone: (403) 241-1719

Mississauga, Ontario
Tech-Trek, Ltd.
Phone: (416) 238-0366

Nepean, Ontario
Tech-Trek, Ltd.
Phone: (613) 225-5161

Richmond, B.C.
Tech-Trek, Ltd.
Phone: (604) 276-8735

Ville St. Laurent, Quebec
Tech-Trek, Ltd.
Phone: (514) 337-7540

PUERTO RICO
Santurce
Mectron Group
Phone: (809) 728-3280

DISTRIBUTORS

Contact one of our local distributors:
Anthem Electronics
Falcon Electronics, Inc.
Gerber Electronics
Hamilton/Avnet Electronics
Marshall Industries
Lex Electronics
Wyle/LEMG
Zentronics, Ltd.

11/9/90

DATA HANDBOOK SYSTEM

DATA HANDBOOK SYSTEM

Our Data Handbook System comprises more than 60 books with specifications on electronic components, subassemblies and materials. It is made up of six series of handbooks:

INTEGRATED CIRCUITS

DISCRETE SEMICONDUCTORS

DISPLAY COMPONENTS

PASSIVE COMPONENTS*

PROFESSIONAL COMPONENTS**

MATERIALS*

The contents of each series are listed on pages iii to viii.

The data handbooks contain all pertinent data available at the time of publication, and each is revised and reissued periodically.

When ratings or specifications differ from those published in the preceding edition they are indicated with arrows in the page margin. Where application is given it is advisory and does not form part of the product specification.

Condensed data on the preferred products of Philips Components is given in our Preferred Type Range catalogue (issued annually).

Information on current Data Handbooks and how to obtain a subscription for future issues is available from any of the Organizations listed on the back cover.
Product specialists are at your service and enquiries will be answered promptly.

* Will replace the Components and materials (green) series of handbooks.
** Will replace the Electron tubes (blue) series of handbooks.

INTEGRATED CIRCUITS

This series of handbooks comprises:

code	handbook title
IC01	**Radio, audio and associated systems** Bipolar, MOS
IC02a/b	**Video and associated systems** Bipolar, MOS
IC03	**ICs for Telecom** Bipolar, MOS Subscriber sets, Cordless Telephones
IC04	**HE4000B logic family** CMOS
IC05	**Advanced Low-power Schottky (ALS) Logic Series**
IC06	**High-speed CMOS; PC74HC/HCT/HCU** Logic family
IC07	**Advanced CMOS logic (ACL)**
IC08	**ECL 10K and 100K logic families**
IC09N	**TTL logic series**
IC10	**Memories** MOS, TTL, ECL
IC11	**Linear Products**
IC12	**I^2C-bus compatible ICs**
IC13	**Semi-custom** Programmable Logic Devices (PLD)
IC14	**Microcontrollers** NMOS, CMOS
IC15	**FAST TTL logic series**
Supplement to IC15	**FAST TTL logic series**
IC16	**CMOS integrated circuits for clocks and watches**
IC17	**ICs for Telecom** Bipolar, MOS Radio pagers Mobile telephones ISDN
IC18	**Microprocessors and peripherals**
IC19	**Data communication products**

DISCRETE SEMICONDUCTORS

This series of data handbooks comprises:

current code	new code	handbook title
S1	SC01	**Diodes** High-voltage tripler units
S2a	SC02*	**Power diodes**
S2b	SC03*	**Thyristors and triacs**
S3	SC04	**Small-signal transistors**
S4a	SC05	**Low-frequency power transistors and hybrid IC power modules**
S4b	SC06	**High-voltage and switching power transistors**
S5	SC07	**Small-signal field-effect transistors**
S6	SC08	**RF power transistors**
	SC09	**RF power modules**
S7	SC10	**Surface mounted semiconductors**
S8a	SC11*	**Light emitting diodes**
S8b	SC12	**Optocouplers**
S9	SC13*	**PowerMOS transistors**
S10	SC14	**Wideband transistors and wideband hybrid IC modules**
S11	SC15	**Microwave transistors**
S15**	SC16	**Laser diodes**
S13	SC17	**Semiconductor sensors**
S14	SC18*	**Liquid crystal displays and driver ICs for LCDs**

* Not yet issued with the new code in this series of handbooks.
** New handbook in this series; will be issued shortly.

DISPLAY COMPONENTS

This series of data handbooks comprises:

current code	new code	handbook title
T8	DC01	**Colour display components**
T16	DC02	**Monochrome monitor tubes and deflection units**
C2	DC03	**Television tuners, coaxial aerial input assemblies**
C3	DC04*	**Loudspeakers**
C20	DC05	**Flyback transformers, mains transformers and general-purpose FXC assemblies**

* These handbooks are currently issued in another series; they are not yet issued in the Display Components series of handbooks.

PASSIVE COMPONENTS

This series of data handbooks comprises:

current code	new code	handbook title
C14	**PA01**	**Electrolytic capacitors; solid and non-solid**
C11	**PA02**	**Varistors, thermistors and sensors**
C12	**PA03**	**Potentiometers and switches**
C7	**PA04**	**Variable capacitors**
C22	**PA05***	**Film capacitors**
C15	**PA06***	**Ceramic capacitors**
C9	**PA07***	**Piezoelectric quartz devices**
C13	**PA08**	**Fixed resistors**

* Not yet issued with the new code in this series of handbooks.

PROFESSIONAL COMPONENTS

This series of data handbooks comprises:

current code	new code	handbook title
T1	*	Power tubes for RF heating and communications
T2a	*	Transmitting tubes for communications, glass types
T2b	*	Transmitting tubes for communications, ceramic types
T3	PC01**	High-power klystrons
T4	*	Magnetrons for microwave heating
T5	PC02**	Cathode-ray tubes
T6	PC03**	Geiger-Müller tubes
T9	PC04**	Photo and electron multipliers
T10	PC05	Plumbicon camera tubes and accessories
T11	PC06	Circulators and Isolators
T12	PC07	Vidicon and Newvicon camera tubes and deflection units
T13	PC08	Image intensifiers
T15	PC09**	Dry reed switches
C8	PC10	Variable mains transformers; annular fixed transformers
	PC11	Solid state image sensors and peripheral integrated circuits

* These handbooks will not be reissued.
** Not yet issued with the new code in this series of handbooks.

MATERIALS

This series of data handbooks comprises:

current code	new code	handbook title
C4 } C5 }	MA01*	**Soft Ferrites**
C16	MA02**	**Permanent magnet materials**
C19	MA03**	**Piezoelectric ceramics**

* Handbooks C4 and C5 will be reissued as one handbook having the new code MA01.

** Not yet issued with the new code in this series of handbooks.

Philips Components — a worldwide Company

Argentina: PHILIPS ARGENTINA S.A. Div. Philips Components, Vedia 3892, 1430 BUENOS AIRES, Tel. (01) 541-4261.

Australia: PHILIPS COMPONENTS PTY Ltd, 11 Waltham Street, ARTARMON, N.S.W. 2064, TEL. (02) 439 3322.

Austria: OSTERREICHISCHE PHILIPS INDUSTRIE G.m.b.H., UB Bauelemente, Triester Str. 64, 1101 WIEN. Tel. (0222) 60 101-820.

Belgium: N.V. PHILIPS PROF. SYSTEMS — Components Div., 80 Rue Des Deux Gares, B-1070 BRUXELLES, Tel. (02) 5256111.

Brazil: PHILIPS COMPONENTS (Active Devices)
Av. das Nacoes Unidas, 12495-SAO PAULO-SP, CEP 04578, P.O. Box 7383, Tel. (011) 534-2211
PHILIPS COMPONENTS (Passive Devices & Materials)
Av. Francisco Monteiro, 702 — RIBEIRAO PIRES-SP, CEP 09400, Tel. (011) 459-8211.

Canada: PHILIPS ELECTRONICS LTD., Philips Components, 601 Milner Ave., SCARBOROUGH, Ontario, M1B 1M8 Tel. (416) 292-5161.
(IC Products) SIGNETICS CANADA LTD., 1 Eva Road, Suite 433, ETOBICOKE, Ontario, M9C 4Z5, Tel. (416) 626-6676.

Chile: PHILIPS CHILENA S.A., Av. Santa Maria 0760, SANTIAGO, Tel. (02) 773816.

Colombia: IPRELENSO LTDA., Carrera 21 No. 56-17, BOGOTA, D.E., P.O. Box 77621, Tel. (01) 249-7624.

Denmark: PHILIPS COMPONENTS A/S, Prags Boulevard 80, PB1919, DK-2300 COPENHAGEN S, Tel. 01-541133.

Finland: PHILIPS COMPONENTS, Sinikalliontie 3, SF-2630 ESPOO Tel. 358-0-50261.

France: PHILIPS COMPOSANTS, 117 Quai du President Roosevelt, 92134 ISSY-LES-MOULINEAUX Cedex, Tel. (01) 40938000.

Germany (Fed. Republic): PHILIPS COMPONENTS UB der Philips G.m.b.H., Burchardstrasse 19, D-2 HAMBURG 1, Tel. (040) 3296-0.

Greece: PHILIPS HELLENIQUE S.A., Components Division, No. 15, 25th March Street, GR 17778 TAVROS, Tel. (01) 4894339/4894911.

Hong Kong: PHILIPS HONG KONG LTD., Components Div., 15/F Philips Ind. Bldg., 24-28 Kung Yip St., KWAI CHUNG, Tel. (0)-4245 121.

India: PEICO ELECTRONICS & ELECTRICALS LTD., Components Dept., Band Box Building, 254-D Dr. Annie Besant Rd., BOMBAY — 400025, Tel. (022) 4930311/4930590.

Indonesia: P.T. PHILIPS-RALIN ELECTRONICS, Components Div., Setiabudi II Building, 6th Fl., Jalan H.R. Rasuna Said (P.O. Box 223/KBY) Kuningan, JAKARTA 12910, Tel. (021) 517995.

Ireland: PHILIPS ELECTRONICS (IRELAND) LTD., Components Division, Newstead, Clonskeagh, DUBLIN 14, Tel. (01) 693355.

Italy: PHILIPS S.p.A., Philips Components, Piazza IV Novembre 3, I-20124 MILANO, Tel. (02) 6752.1.

Japan: PHILIPS JAPAN LTD., Components Division, Philips Bldg. 13-37, Kohnan 2-chome, Minato-ku, TOKYO (108), Tel. (03) 7405028

Korea (Republic of): PHILIPS ELECTRONICS (KOREA) LTD., Components Division, Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL, Tel. (02) 794-5011.

Malaysia: PHILIPS MALAYSIA SDN BHD, Components Div., 3 Jalan SS15/2A SUBANG, 47500 PETALING JAYA, Tel. (03) 7345511.

Mexico: PHILIPS COMPONENTS, Paseo Triunfo de la Republica, No 215 Local 5, Cd Juarez CHIHUAHUA 32340 MEXICO, Tel. (16) 18-67-01/02.

Netherlands: PHILIPS NEDERLAND B.V., Marktgroep Philips Components, Postbus 90050, 5600 PB EINDHOVEN, Tel. (040) 783749.

New Zealand: PHILIPS NEW ZEALAND LTD., Components Division, 110 Mt. Eden Road, C.P.O. Box 1041, AUCKLAND, Tel. (09) 605-914.

Norway: NORSK A/S PHILIPS, Philips Components, Box 1, Manglerud 0612, OSLO, Tel. (02) 680200.

Pakistan: PHILIPS ELECTRICAL CO. OF PAKISTAN LTD., Philips Markaz, M.A. Jinnah Rd., KARACHI-3, Tel. (021) 725772.

Peru: CADESA, Carretera Central 6.500, LIMA 3, Apartado 5612, Tel. 51-14-350059.

Philippines: PHILIPS ELECTRICAL LAMPS INC., Components Div., 106 Valero St. Salcedo Village, MAKATI, P.O. Box 911, METRO MANILA, Tel. (63-2) 810-0161

Portugal: PHILIPS PORTUGUESA S.A.R.L., Av. Eng. Duarte Pacheco 6, 1009 LISBOA Codex, Tel. (019) 683121.

Singapore: PHILIPS SINGAPORE, PTE LTD., Components Div., Lorong 1, Toa Payoh, SINGAPORE 1231, Tel. 3502000.

South Africa: S.A. PHILIPS PTY LTD., Components Division, JOHANNESBURG 2000, P.O. Box 7430.

Spain: PHILIPS COMPONENTS, Balmes 22, 08007 BARCELONA, Tel. (03) 3016312.

Sweden: PHILIPS COMPONENTS, A.B., Tegeluddsvagen 1, S-11584 STOCKHOLM, Tel. (0)8-7821000.

Switzerland: PHILIPS A.G., Components Dept., Allmendstrasse 140-142, CH-8027 ZURICH, Tel. (01) 4882211.

Taiwan: PHILIPS TAIWAN LTD., 581 Min Sheng East Road, P.O. Box 22978, TAIPEI 10446, Taiwan, Tel. 886-2-509-7666.

Thailand: PHILIPS ELECTRICAL CO. OF THAILAND LTD., 283 Silom Road, P.O. Box 961, BANGKOK, Tel. (02) 233-6330-9.

Turkey: TURK PHILIPS TICARET A.S., Philips Components, Talatpasa Cad. No. 5, 80640 LEVENT/ISTANBUL, Tel. (01) 1792770.

United Kingdom: PHILIPS COMPONENTS LTD., Mullard House, Torrington Place, LONDON WC1E 7HD, Tel. (01) 5806633.

United States: (Color Picture Tubes - Monochrome & Colour Display Tubes) PHILIPS DISPLAY COMPONENTS COMPANY, 1600 Huron Parkway, P.O. Box 963, ANN ARBOR, Michigan 48106, Tel. (313) 996-9400.
(IC Products) SIGNETICS COMPANY, 811 East Arques Avenue, SUNNYVALE, CA 94088-3409, Tel. (408) 991-2000.
(Passive Components, Discrete Semiconductors, Materials and Professional Components) PHILIPS COMPONENTS, Discrete Products Division, 2001 West Blue Heron Blvd., P.O. Box 10330, RIVIERA BEACH, Florida 33404, Tel. (407) 881-3200

Uruguay: PHILIPS COMPONENTS, Coronel Mora 433, MONTEVIDEO, Tel. (02) 70-4044.

Venezuela: MAGNETICA S.A., Calle 6, Ed. Las Tres Jotas, CARACAS 1074A, App. Post. 78117, Tel. (02) 241 7509.

Zimbabwe: PHILIPS ELECTRICAL (PVT) LTD., 62 Mutare Road, HARARE, P.O. Box 994, Tel. 47211.

For all other countries apply to: Philips Components Division, Strategic Accounts and International Sales, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands. Telex 35000 phtcnl, Fax 23753.

98-7001-230

1002B/60M/CR4/0191

Signetics

Philips Components

PHILIPS

PHILIPS