



**National
Semiconductor**

400027

1989

Advanced Peripherals

Graphics Handbook

National Semiconductor

**Advanced
Peripherals**

**Graphics
Handbook**



A Corporate Dedication to Quality and Reliability

National Semiconductor is an industry leader in the manufacture of high quality, high reliability integrated circuits. We have been the leading proponent of driving down IC defects and extending product lifetimes. From raw material through product design, manufacturing and shipping, our quality and reliability is second to none.

We are proud of our success . . . it sets a standard for others to achieve. Yet, our quest for perfection is ongoing so that you, our customer, can continue to rely on National Semiconductor Corporation to produce high quality products for your design systems.

A handwritten signature in black ink, reading 'Charles E. Sporck'. The signature is written in a cursive, flowing style with a prominent initial 'C'.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Wir fühlen uns zu Qualität und Zuverlässigkeit verpflichtet

National Semiconductor Corporation ist führend bei der Herstellung von integrierten Schaltungen hoher Qualität und hoher Zuverlässigkeit. National Semiconductor war schon immer Vorreiter, wenn es galt, die Zahl von IC Ausfällen zu verringern und die Lebensdauern von Produkten zu verbessern. Vom Rohmaterial über Entwurf und Herstellung bis zur Auslieferung, die Qualität und die Zuverlässigkeit der Produkte von National Semiconductor sind unübertroffen.

Wir sind stolz auf unseren Erfolg, der Standards setzt, die für andere erstrebenswert sind. Auch ihre Ansprüche steigen ständig. Sie als unser Kunde können sich auch weiterhin auf National Semiconductor verlassen.

La Qualité et La Fiabilité: Une Vocation Commune Chez National Semiconductor Corporation

National Semiconductor Corporation est un des leaders industriels qui fabrique des circuits intégrés d'une très grande qualité et d'une fiabilité exceptionnelle. National a été le premier à vouloir faire chuter le nombre de circuits intégrés défectueux et a augmenter la durée de vie des produits. Depuis les matières premières, en passant par la conception du produit sa fabrication et son expédition, partout la qualité et la fiabilité chez National sont sans équivalents.

Nous sommes fiers de notre succès et le standard ainsi défini devrait devenir l'objectif à atteindre par les autres sociétés. Et nous continuons à vouloir faire progresser notre recherche de la perfection; il en résulte que vous, qui êtes notre client, pouvez toujours faire confiance à National Semiconductor Corporation, en produisant des systèmes d'une très grande qualité standard.

Un Impegno Societario di Qualità e Affidabilità

National Semiconductor Corporation è un'industria al vertice nella costruzione di circuiti integrati di alta qualità ed affidabilità. National è stata il principale promotore per l'abbattimento della difettosità dei circuiti integrati e per l'allungamento della vita dei prodotti. Dal materiale grezzo attraverso tutte le fasi di progettazione, costruzione e spedizione, la qualità e affidabilità National non è seconda a nessuno.

Noi siamo orgogliosi del nostro successo che fissa per gli altri un traguardo da raggiungere. Il nostro desiderio di perfezione è d'altra parte illimitato e pertanto tu, nostro cliente, puoi continuare ad affidarti a National Semiconductor Corporation per la produzione dei tuoi sistemi con elevati livelli di qualità.



Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

GRAPHICS HANDBOOK

1989 Edition

Advanced Graphics Chipset

DP8500 Development Tools

Application Notes

Physical Dimensions/Appendices

1

2

3

4

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

Abuseable™	FAIRCAD™	MST™	SCX™
Anadig™	Fairtech™	Naked-8™	SERIES/800™
ANS-R-TRAN™	FAST®	National®	Series 900™
APPST™	5-Star Service™	National Semiconductor®	Series 3000™
ASPECT™	GENIX™	National Semiconductor Corp.®	Series 32000®
Auto-Chem Deflasher™	GNX™	NAX 800™	Shelf✓Chek™
BCP™	HAMR™	Nitride Plus™	SofChek™
BI-FET™	HandiScan™	Nitride Plus Oxide™	SPIRE™
BI-FET II™	HEX 3000™	NML™	START™
BI-LINE™	HPCT™	NOBUST™	Starlink™
BIPLAN™	I ³ L®	NSC800™	STARPLEX™
BLC™	ICM™	NSCISE™	Super-Block™
BLX™	INFOCHEX™	NSX-16™	SuperChip™
Brite-Lite™	Integral ISET™	NS-XC-16™	SuperScript™
BTL™	Intelisplay™	NTERCOM™	SYS32™
CheckTrack™	ISE™	NURAM™	TapePak®
CIM™	ISE/06™	OXISS™	TDST™
CIMBUST™	ISE/08™	P ² CMOST™	TeleGate™
CLASIC™	ISE/16™	PC Master™	The National Anthem®
Clock✓Chek™	ISE32™	Perfect Watch™	Time✓Chek™
COMBO™	ISOPLANAR™	Pharma✓Chek™	TINAT™
COMBO I™	ISOPLANAR-Z™	PLANAT™	TLCT™
COMBO II™	KeyScan™	PLANART™	Trapezoidal™
COPST™ microcontrollers	LMCMOST™	Plus-2™	TRI-CODE™
Datachecker®	M ² CMOST™	Polycraft™	TRI-POLY™
DENSPAK™	Macrobus™	POSilink™	TRI-SAFET™
DIB™	Macrocomponent™	POSITalker™	TRI-STATE®
Digitaltalker®	MAXI-ROM®	Power + Control™	TURBOTRANSCEIVER™
DISCERN™	Meat✓Chek™	POWERplanar™	VIPT™
DISTILL™	MenuMaster™	QUAD3000™	VR32™
DNR®	Microbus™ data bus	QUIKLOOK™	WATCHDOG™
DPVM™	MICRO-DACT™	RAT™	XMOST™
ELSTAR™	μtalker™	RTX16™	XPUT™
Embedded System Processor™	Microtalker™	SABR™	Z START™
E-Z-LINK™	MICROWIRE™	Script✓Chek™	883B/RETSTM™
FACT™	MICROWIRE/PLUSTM™		883S/RETSTM™
	MOLETM™		

GAL® is a registered trademark of Lattice Semiconductor.

PAL® is a registered trademark of and is used under license from Monolithic Memories, Inc.

UNIX® is a registered trademark of AT & T.

DEC™ and Ultrix™ are trademarks of Digital Equipment Corp.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000 TWX (910) 339-9240

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry or specifications.



Introduction

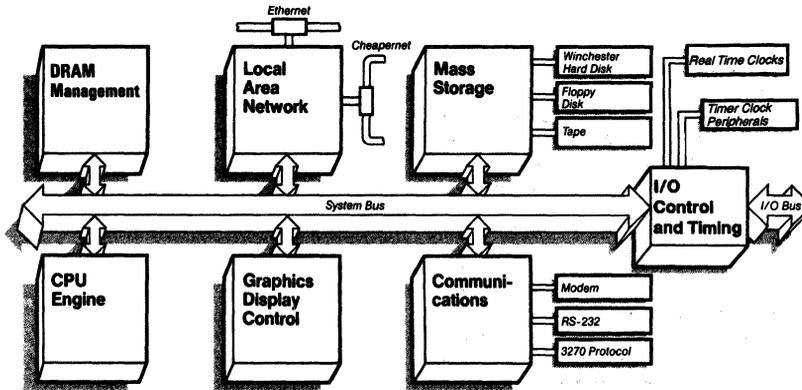
National Semiconductor's graphics and imaging products include complex VLSI peripheral circuits designed to serve a variety of applications. These products are especially well suited for microcomputer and microprocessor systems such as workstations, terminals, personal computers, and many others.

GRAPHICS

Sophisticated human interface is the mark of the newest computer systems. Today's personal computer may have better graphics display capability than engineering workstations of a few years ago. National Semiconductor has developed a new family of Advanced Graphics products to provide extremely high performance, high resolution monochrome or color graphics displays. The graphics chip set is designed to provide the highest level of performance without placing constraints on the overall system design or performance. Flexibility is as important as is overall performance. That flexibility is provided by the partitioned functionality, modular building block approach, open architecture, programmability of all components and the ability to address the frame buffer in a planar (parallel) mode or in a pixel-wise mode. The graphics system may be expanded to any number of color planes with virtually unlimited resolution. The Graphics Databook provides all of the details to make display system design easy.



Introduction Advanced Peripherals



TL/XX/0058-1

National Semiconductor Advanced Peripherals products include complex VLSI peripheral circuits designed to serve a variety of applications. The Advanced Peripherals products are especially well suited for microcomputer and microprocessor systems such as graphics workstations, personal computers, and many others. National Semiconductor Advanced Peripherals devices are fully described in a series of databooks and handbooks.

Among the Advanced Peripherals books are the following titles:

MASS STORAGE

The National Semiconductor family of mass storage interface products offers the industry's highest performance and broadest range of products for Winchester hard disks and floppy disks. The Mass Storage Handbook includes complete product information and datasheets as well as a comprehensive design guide for disk controller systems.

DRAM MANAGEMENT

Today's large Dynamic Random Access Memory (DRAM) arrays require sophisticated high performance devices to provide timing access arbitration on board drive and control. National Semiconductor offers the broadest range of DRAM controllers with the highest "No-waitstate" performance available on the market. Controllers are available in Junction Isolated LS, Oxide Isolated ALS, and double metal CMOS for DRAMs from 64k bit through 4M bit devices, supporting memory arrays up to 64 Mbyte in size with only one LSI/VLSI device. For critical applications, National Semiconductor has developed several 16- and 32-bit Error Checking and Correction (ECC) devices to provide maximum data integrity. The Memory Support Handbook contains complete product information and several application notes detailing complete memory system design.

LOCAL AREA NETWORKS AND DATA COMMUNICATIONS

Today's computer systems have created a huge demand for data communications and Local Area Networks (LANs).

National Semiconductor provides a complete three-chip solution for an entire IEEE 802.3 standard for Ethernet/CheaperNet LANs. National Semiconductor offers a completely integrated solution for the IBM 370 class mainframes, System 3X and AS/400 systems for physical layer front end and processing of the IBM 3270/3299 "coaxial" and 5250 "twinaxial" protocols. To drive the communications lines, National Semiconductor has drivers and receivers designed to meet all the major standards such as RS-232, RS-422, and RS-485. Datasheets and applications information for all these products are in the LAN/DATA COMM Handbook.

GRAPHICS

Sophisticated human interface is a mark of the newest computer systems designs. Today's personal computer may have better graphics display capability than engineering workstations of a few years ago. National Semiconductor has developed a new family of Advanced Graphics products to provide extremely high performance, high resolution color graphics displays. The graphics chip set is designed to provide the highest level of performance with minimum demands and loading on the system CPU. The graphics system may be expanded to any number of color planes with virtually unlimited resolution. The Graphics Databook lays it all out and makes the display system design easy.

REAL TIME CLOCKS

National offers a family of Real Time Clocks (RTCs) and advanced Timer Clock Peripherals (TCPs). The RTC family provides a simple μ P bus compatible interface to any system requiring accurate, reliable, on-going real time and calendar functions. The TCP family offers the RTC, RAM and two 16-bit programmable timers with fast μ P bus handshake controls for chip select, read and write. The Real Time Clock handbook includes complete product information and datasheets as well as applications information.



Product Status Definitions

Definition of Terms

Data Sheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This data sheet contains preliminary data, and supplementary data will be published at a later date. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

National Semiconductor Corporation reserves the right to make changes without further notice to any products herein to improve reliability, function or design. National does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

Table of Contents

Alphanumeric Index	ix
Section 1 Advanced Graphics Chipset	
DP8500 Raster Graphics Processor	1-3
NS32CG16-10, NS32CG16-15 High-Performance Printer/Display Processor	1-62
DP8510 BITBLT Processing Unit	1-63
DP8511 BITBLT Processing Unit (BPU)	1-84
DP8512 Video Clock Generator	1-120
DP8513 Multi-Board Video Clock Generator	1-135
DP8514 Crystal Clock Generator	1-153
DP8515/DP8515-350/DP8516/DP8516-350 Video Shift Register (VSR)	1-159
DP8520A/DP8521A/DP8522A microCMOS Programmable 256k/1M/4M Video RAM Controller/Driver	1-174
DP8530 Clock Generator	1-262
Section 2 DP8500 Development Tools	
DP850EB Raster Graphics Processor (RGP) 4-Plane Evaluation System	2-3
DP850DB8 Raster Graphics Processor (RGP) 8-Plane Development System	2-4
DP8500 Raster Graphics Processor Software Tools	2-5
Section 3 Application Notes	
AN-451 An Architectural Solution for High Performance Graphics	3-3
AN-547 Interfacing the DP8500 Raster Graphics Processor	3-11
AN-391 The LM1823 A High Quality TV Video IF Amplifier and Synchronous Detector for Cable Receivers	3-26
AN-402 LM2889 R.F. Modulator	3-42
AN-580 A 16-Bit Video Shift Register with On-Board FIFO Operates at Rates Up to 350 Million Pixels per Second	3-53
AN-553 Mid-Scan-Line Load Techniques Using the DP8500 Raster Graphics Processor	3-60
AN-554 Accurate Timing for Multi-Board Graphics Systems	3-65
AN-604 DP8512, DP8513, DP8514 Video Clock Generator Evaluation Board	3-69
AN-609 A Graphics Acceleration Card for the AT Using the Advanced Graphics Chip Set	3-82
Section 4 Physical Dimensions/Appendices	
Appendix A: Related Datasheets	4-3
Physical Dimensions	4-4
Bookshelf	
Distributors	

Alpha-Numeric Index

AN-391 The LM1823 A High Quality TV Video IF Amplifier and Synchronous Detector for Cable Receivers	3-26
AN-402 LM2889 R.F. Modulator	3-42
AN-451 An Architectural Solution for High Performance Graphics	3-3
AN-547 Interfacing the DP8500 Raster Graphics Processor	3-11
AN-553 Mid-Scan-Line Load Techniques Using the DP8500 Raster Graphics Processor	3-60
AN-554 Accurate Timing for Multi-Board Graphics Systems	3-65
AN-580 A 16-Bit Video Shift Register with On-Board FIFO Operates at Rates Up to 350 Million Pixels per Second	3-53
AN-604 DP8512, DP8513, DP8514 Video Clock Generator Evaluation Board	3-69
AN-609 A Graphics Acceleration Card for the AT Using the Advanced Graphics Chip Set	3-82
DP850DB8 Raster Graphics Processor (RGP) 8-Plane Development System	2-4
DP850EB Raster Graphics Processor (RGP) 4-Plane Evaluation System	2-3
DP8500 Raster Graphics Processor Software Tools	2-5
DP8500 Raster Graphics Processor	1-3
DP8510 BITBLT Processing Unit	1-63
DP8511 BITBLT Processing Unit (BPU)	1-84
DP8512 Video Clock Generator	1-120
DP8513 Multi-Board Video Clock Generator	1-135
DP8514 Crystal Clock Generator	1-153
DP8515 Video Shift Register (VSR)	1-159
DP8515-350 Video Shift Register (VSR)	1-159
DP8516 Video Shift Register (VSR)	1-159
DP8516-350 Video Shift Register (VSR)	1-159
DP8520A microCMOS Programmable 256k/1M/4M Video RAM Controller/Driver	1-174
DP8521A microCMOS Programmable 256k/1M/4M Video RAM Controller/Driver	1-174
DP8522A microCMOS Programmable 256k/1M/4M Video RAM Controller/Driver	1-174
DP8530 Clock Generator	1-262
NS32CG16-10 High-Performance Printer/Display Processor	1-62
NS32CG16-15 High-Performance Printer/Display Processor	1-62



Section 1
Advanced Graphics Chipset



Section 1 Contents

DP8500 Raster Graphics Processor	1-3
NS32CG16-10, NS32CG16-15 High-Performance Printer/Display Processor	1-62
DP8510 BITBLT Processing Unit	1-63
DP8511 BITBLT Processing Unit (BPU)	1-84
DP8512 Video Clock Generator	1-120
DP8513 Multi-Board Video Clock Generator	1-135
DP8514 Crystal Clock Generator	1-153
DP8515/DP8515-350/DP8516/DP8516-350 Video Shift Register (VSR)	1-159
DP8520A/DP8521A/DP8522A microCMOS Programmable 256k/1M/4M Video RAM Controller/Driver	1-174
DP8530 Clock Generator	1-262

DP8500 Raster Graphics Processor

General Description

National's DP8500 Raster Graphics Processor (RGP) is a microprocessor specifically tuned for graphics applications. A member of the Advanced Graphics Chip Set, it provides the set of functions required for display buffer update and video refresh in mid-to-high-performance color or monochrome raster graphics systems employing both graphics primitives and text. The RGP combines the following elements: a general-purpose microcoded microprocessor, a programmable video refresh generator, a vector generator, a BITBLT controller and a rectangular clipper. As such, it may be used in standalone applications or as a dedicated graphics engine in conjunction with any general-purpose microprocessor.

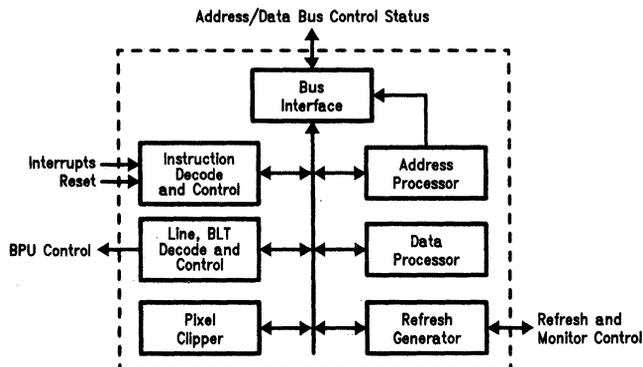
The RGP supports a system architecture that features constant drawing speed, measured in pixels/second, independent of the depth (number of bits) of the pixel. This key feature arises from the RGP's use of an external data path device, the BITBLT Processing Unit (BPU), for all drawing functions. By employing a BPU on each plane of memory, the traditional "bottleneck" is removed from the data manipulation path. In effect, the data bus width (for drawing purposes) is made proportional to the pixel depth, thus preserving the drawing speed as pixel depth is increased from one bit to any number of bits.

During video refresh, the RGP produces all synchronization and blanking signals for CRT displays and generates memory cycles, appropriate for the type of memory used, on behalf of the video shift registers. Any type of memory may be used: SRAM, DRAM or video RAM. In addition, the RGP supports the use of fast access modes in dynamic RAMS, such as page mode or static column mode.

Features

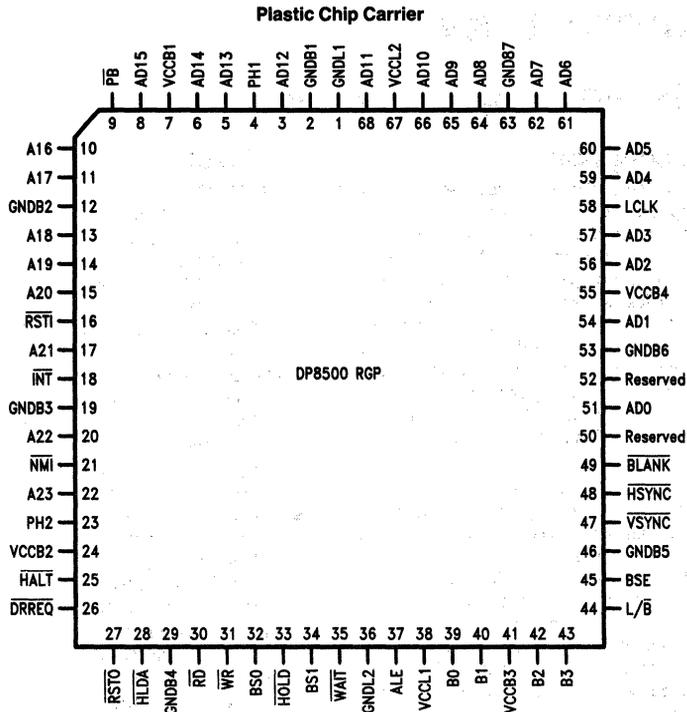
- 20 MHz operation
- Fully programmable
- Large, uniform address space
 - 28-bit bit (pixel) address
 - 24-bit word address
 - 16-bit data bus
 - Program, data, and display memories can reside anywhere
- Flexible bus interface
 - Processor independent
 - Conventional HOLD/HLDA mechanism
- Large drawing space
 - Up to 16384 by 16384 pixels per bitmap
 - Pixels of any depth
- Dedicated graphics hardware
 - Vector generator
 - Line pattern generator
 - BITBLT controller
 - Rectangular clipper
- Efficient text support
 - Multiple fonts/sizes
 - Proportional spacing
- Programmable video refresh
 - Can be disabled for laser printer applications
 - Pixel rates to 250 MHz and beyond
 - Display formats to 65536 pixels by 4096 scan lines
 - Interlaced or non-interlaced
 - Genlock support
- microCMOS technology
- 68-lead PLCC package

Block Diagram



TL/F/9427-1

Connection Diagram



TL/F/9427-2

Order Number DP8500V
See NS Package Number V68A

Pin Descriptions

Pin	Description
SUPPLIES	
VCC1- VCC2	Positive supply for internal logic: 5 Vdc ± 10%.
GND1- GND2	Ground for internal logic.
VCCB1- VCCB4	Positive supply for on-chip buffers: 5Vdc ± 10%.
GNDB1- GNDB7	Ground for on-chip buffers.
INPUTS	
PH1	Phase 1. PH1 is an MOS-level clock normally provided by the VCG. It must have no overlap with the PH2 clock.
PH2	Phase 2. PH2 is an MOS-level clock normally provided by the VCG. It must have no overlap with the PH1 clock.

Pin	Description
INPUTS (Continued)	
LCLK	Load Clock. LCLK is a TTL-compatible clock normally supplied by the VCG's LCLK0. It provides the basic time unit used in the registers that specify digitally the video refresh functions: sync, blanking and display-refresh bus cycles.
RSTI	Reset In (active low) RSTI is used to place the RGP into the reset state; this is typically done at power-up.
WAIT	Wait (active low). WAIT is used to cause the RGP to insert one or more wait states into the current bus cycle. This mechanism can be used to accommodate the RGP to relatively slow memory devices; it can also serve as a bus-not-ready indication from a bus arbiter.

Pin Descriptions (Continued)

Pin	Description
INPUTS (Continued)	
HOLD	Hold Request (active low). $\overline{\text{HOLD}}$ serves as a request from another master for the RGP's bus. In response, the RGP will complete the current bus cycle (if any), TRI-STATE® only the address and data buses and assert Hold Acknowledge.
INT	Interrupt Request (active low). $\overline{\text{INT}}$ causes the RGP to suspend normal processing after completion of the current instruction (if any), to save the Program Counter and Processor Status Register on the stack and to enter the user's interrupt service routine. This function can be disabled via the Processor Status Register.
NMI	Non-Maskable Interrupt Request (negative edge triggered). NMI causes the RGP to suspend normal processing after the completion of the current instruction (if any), to save the Program Counter and Processor Status Register on the stack and to enter the user's non-maskable interrupt service routine. This function is disabled upon Reset, and is enabled upon the first store operation to the PSR register.
RESERVED	Reserved for NSC testing. Must be tied to ground.
OUTPUTS	
A16–A23	Address Lines. A16–A23 provide the most significant eight bits of memory addresses during bus cycles. A23 is the most significant. A16–A23 are at TRI-STATE whenever $\overline{\text{HLDA}}$ is asserted by the RGP. Addresses are guaranteed to be valid at the falling edge of ALE.
BS0–BS1	Bus Status Lines. BS0–BS1 are used to indicate the type of bus cycle to be performed by the RGP. (See Table I.) BS0–BS1 become valid during the first T-STATE of a bus cycle (before the rising edge of ALE) and remain valid through the end of the last T-state of that cycle.
$\overline{\text{RD}}$	Read (active low status). $\overline{\text{RD}}$ indicates that the current cycle will cause memory to be read. $\overline{\text{RD}}$ becomes valid during the first T-state of a bus cycle (before the rising edge of ALE) and remains valid through the end of the last T-state of that cycle. Both $\overline{\text{RD}}$ and $\overline{\text{WR}}$ may be asserted in the same cycle. This indicates that the RGP is requesting a read-modify-write operation in the current cycle.

Pin	Description
OUTPUTS (Continued)	
$\overline{\text{WR}}$	Write (active low status). $\overline{\text{WR}}$ indicates that the current cycle will cause memory to be written. $\overline{\text{WR}}$ becomes valid during the first T-state of a bus cycle (before the rising edge of ALE) and remains valid through the end of the last T-state of that cycle. Both $\overline{\text{WR}}$ and $\overline{\text{RD}}$ may be asserted in the same cycle. This indicates that the RGP is requesting a read-modify-write operation in the current cycle.
ALE	Address Latch Enable. ALE indicates the beginning of a bus cycle. Its rising edge indicates that bus status, consisting of BS1–0, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are valid. Its falling edge indicates that the $\overline{\text{PB}}$ and the address, consisting of AD0–15, A16–23 and (if used in this cycle) B0–3, are valid.
$\overline{\text{PB}}$	Page Break (active low status). $\overline{\text{PB}}$ indicates that the 16 most significant address bits (AD8–15 and A16–23) of the RGP's current bus cycle are not equal to the 16 most significant address bits generated by the RGP in the previous bus cycle. $\overline{\text{PB}}$ can be used by the system's memory controller to allow it to generate page-mode accesses to memory, resulting in shorter access times. Since the eight least significant bits of addresses are ignored in the above comparison, the page size is always 256 words.
$\overline{\text{HLDA}}$	Hold Acknowledge (active low). $\overline{\text{HLDA}}$ indicates that the RGP has put the address and data buses at TRI-STATE and has entered an internal <i>hold</i> condition. The RGP will not exit the hold condition until the $\overline{\text{HOLD}}$ input has been removed.
HSYNC	Horizontal Synchronization (active low). HSYNC indicates to the CRT monitor that the horizontal sweep should begin its retrace. The RGP can be programmed to use this output as <i>composite sync</i> , that is, the exclusive-NOR of HSYNC and VSYNC. This option is selected via the RGP's Video Control Register.
BLANK	Composite Blanking (active low). $\overline{\text{BLANK}}$ is the result of ORing the internal vertical and horizontal blanking signals. $\overline{\text{BLANK}}$ indicates to the CRT monitor that the screen should be blanked.
$\overline{\text{DRREQ}}$	Display Refresh Request (active low). $\overline{\text{DRREQ}}$ starts generation of a Video Refresh bus cycle. $\overline{\text{DRREQ}}$ can be used by an external bus arbiter as a high-priority bus request. It can also be used for precise control of video DRAM transfer cycles, e.g., when performing mid-scanline transfers.

Pin Descriptions (Continued)

Pin	Description
OUTPUTS (Continued)	
L/ \bar{B}	Line draw/BITBLT mode. L/ \bar{B} sets the operating mode of the BPU during drawing operations. L/ \bar{B} is high to place the BPU into line-drawing mode, low to place the BPU into BITBLT mode. L/ \bar{B} is asserted throughout execution of any drawing instruction that uses the BPU, starting prior to the generation of any bus cycles within that instruction.
BSE	BITBLT Source Enable. BSE directs the flow of data within the BPU. When high, BSE indicates to the BPU that the current bus cycle is associated with BITBLT data in the source rectangle. BSE is low during BITBLT destination data bus cycles and during non-BITBLT cycles. It is valid on the rising edge of ALE and remains valid throughout the bus cycle.
B0/LME	Bit Select 0 or Left Mask Enable. This output to the BPU serves two purposes. When L/ \bar{B} is high it functions as B0 (see below). When L/ \bar{B} is low it functions as LME, which serves to enable the left mask. LME is valid at the falling edge of ALE and remains valid throughout the bus cycle.
B1/RME	Bit Select 1 or Right Mask Enable. This output to the BPU serves two purposes. When L/ \bar{B} is high it functions as B1 (see below). When L/ \bar{B} is low it functions as RME, which serves to enable the right mask. RME is valid at the falling edge of ALE and remains valid throughout the bus cycle.
B2/FWR	Bit Select 2 or FIFO Write. This output to the BPU serves two purposes. When L/ \bar{B} is high it functions as B2 (see below). When L/ \bar{B} is low it functions as FWR, which causes the BPU's barrel-shifter output to be written to the BPU's FIFO. FWR is valid on the rising edge of PH1, two clock periods after \overline{WAIT} is sampled high during the next T2-state bus cycle.
B3/FRD	Bit Select 3 or FIFO Read. This output to the BPU serves two purposes. When L/ \bar{B} is high it functions as B3 (see below). When L/ \bar{B} is low it functions as FRD, which causes the BPU's FIFO output to be read into the BPU's logic unit. FRD is valid at the rising edge of a PH1 during the fetch of the corresponding destination data word.
B0-B3	Bit Select. When L/ \bar{B} is low, these four outputs have other functions (see above). When L/ \bar{B} is high, B0-3 select a specific bit within the word addressed by AD0-15 and A16-23. B0-3 become valid prior to the falling edge of ALE and remain valid throughout the bus cycle.

Pin	Description
OUTPUTS (Continued)	
HALT	Halt (active low). \overline{HALT} indicates that the RGP has executed a HALT instruction and entered the <i>halt</i> state, a state in which no instructions are processed, but video refresh functions continue. The halt state can be exited via an interrupt or reset.
\overline{RSTO}	Reset Out (active low). \overline{RSTO} is driven low whenever \overline{RSTI} is driven low. It is also driven low (for two clock cycles) at the end of execution of FILLA or FILLT instructions. The INITB instruction drives \overline{RSTO} low for two clocks. \overline{RSTO} is normally connected to the BPU's RESET input.
INPUT/OUTPUTS	
AD0-AD15	Multiplexed Address and Data lines. AD0-15 serve as outputs early in a bus cycle, providing the 16 least significant memory address bits. Later in the bus cycle they serve as the data bus. AD0 is the least significant bit of data or address. As a data bus, these lines can be inputs (during reads), outputs (during writes) or can be ignored by the RGP (e.g., during BITBLTs). These lines are at TRI-STATE whenever Hold Acknowledge is asserted by the RGP. Addresses are guaranteed to be valid at the falling edge of ALE.
\overline{VSYNC}	Vertical Synchronization (active low). \overline{VSYNC} can serve as an input or an output. As an output, it indicates to the CRT monitor that the vertical sweep should begin its retrace. As an input, it clears the internal counters associated with vertical sync generation within the RGP. As such, it allows the RGP to synchronize itself with an external video source (this assumes the use of horizontal synchronization features of the VCG). This option is selected via the RGP's Video Control Register.

Table 1

Bus Status (BS1, BS0)	Cycle Type (Function)
0, 0	Operand Read or Write
0, 1	Instruction Fetch
1, 0	BITBLT/Draw (Address Only)
1, 1	Video Refresh (Address Only)

Architectural Description

AGCS OVERVIEW

The RGP, serving as the core of a bitmapped graphics system, is designed to work in concert with the other members of the Advanced Graphics Chip Set (AGCS). Other components of the chip set include the DP8512 Video Clock Generator (VCG), the DP8515/16 Video Shift Register (VSR) and the DP8511 BITBLT Processing Unit (BPU). Additional functions required for system implementation are provided by National through such components as the DP8520/22 Video DRAM Controller and the Bus State Machine (BSM).

The components that comprise the Advanced Graphics Chip Set are fabricated in a variety of technologies, each appropriate to the function performed by that component. Both CMOS and bipolar technologies are used in the family. As a result, the family exhibits both VLSI functionality and 250 MHz operation.

The Advanced Graphics Chip Set supports a high-performance architecture without imposing a particular bus protocol, timing or memory type upon the system designer. As a result, AGCS-based systems can be realized with a range of solutions to the cost/performance tradeoff. Advances in memory technology can be capitalized upon in future systems while retaining software compatibility.

A major feature of the system architecture is the support of simultaneous data manipulation (during drawing) at each of the bitplanes. This allows the system to retain its drawing speed (in pixels per second) as pixel depth increases from one to any number of bits.

The functional interconnection of these components is illustrated in *Figure 1*, which represents a minimal bitmapped graphics system. In this system, the intensity of each pixel is described by a single bit, i.e., each pixel is either on or off. The roles of the various components are described below, as are design considerations for color systems.

DP8500 Raster Graphics Processor (RGP)

The RGP is designed to be the overall control mechanism in graphics systems. It draws graphics objects in the display buffer, refreshes the video display, and performs general purpose computing tasks.

The implementation of the RGP reflects these functions (see *Figure 2*). A general purpose microcoded microprocessor core is augmented by dedicated hardware for the setup and execution of graphics primitives. In addition, a programmable state machine handles all video synchronization functions and produces addresses for video refresh. Internal bus arbitration logic controls access to the external bus.

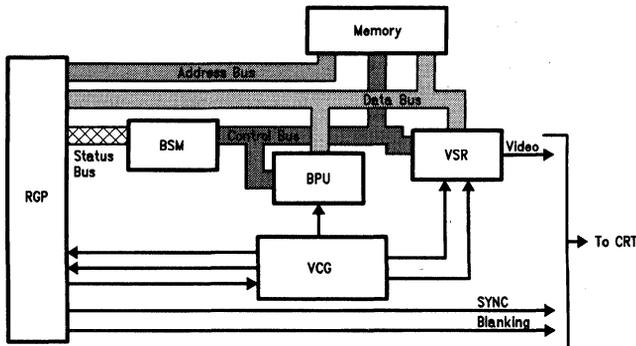


FIGURE 1. Minimal Bit-Mapped Graphics System

TL/F/9427-3

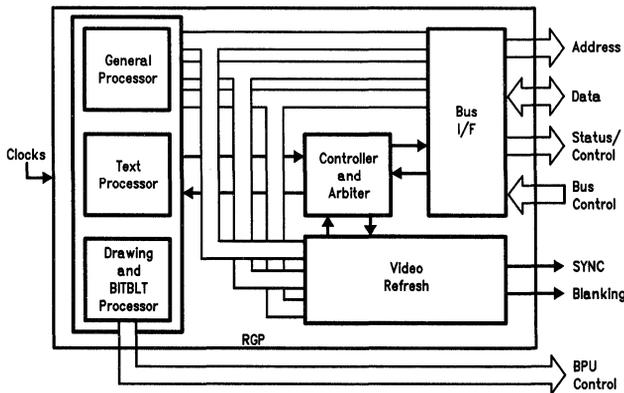


FIGURE 2. RGP Organization Diagram

TL/F/9427-4

Architectural Description (Continued)

DP8511 BITBLT Processing Unit (BPU)

The DP8511 BPU provides a data path for changing the contents of the frame buffer. The drawing functions of the RGP, e.g., line drawing and polygon filling, rely on the BPU to accept data from memory, modify it, and return it to memory. The general scheme underlying these operations is BITBLT (BIT boundary BLock Transfer). During BITBLT, the RGP serves as an address generator to initiate the appropriate memory cycles and as a strobe generator to control the BPU's operation.

The BPU is a microCMOS device intended to provide hardware support for BITBLT. It includes a barrel shifter, used to shift source data into bit alignment with the destination and a FIFO to contain a series of shifted source words. A 16-function logic unit allows all possible bitwise combinations between the source and destination data. The BPU has hardware to support pixel operations and line drawing. In particular, it can address a single bit within the current word and read and write the selected bit. Details of BPU operation can be obtained from the BPU data sheet.

The use of BITBLT as the underlying mechanism for all drawing operations promotes a common method for handling both simple systems, as shown in *Figure 1*, and more complex, high performance systems, as shown in *Figure 3*. This latter figure illustrates a multi-plane system, i.e., one in which several bits (one from each plane) describe the value of a pixel. This might be used to describe the intensity (in monochrome) or the hue (in color) of the pixel. The use of a BPU per plane, permitting simultaneous update of all planes, leads to the highest performance. Alternatively, a cost-performance trade-off might be made by sharing one or more BPU's across two or more planes. At one extreme (single BPU) updates are done serially to each plane. At the other extreme (a BPU per plane) updates are performed in a purely parallel manner. A middle ground could be implemented, updating the planes in a serial-parallel manner. The mechanism of implementing the data path function in the BPU, rather than the RGP, produces this design flexibility.

DP8515/16 Video Shift Register (VSR)

The DP8515/16 VSR shifts pixel information at the data rate required by the CRT. It is implemented in National's

Bipolar-CMOS process. It combines CMOS control logic and a CMOS input FIFO with an ECL shifter. As a result, it provides the system designer additional timing flexibility in the load path coupled with up to 350 MHz shift capability without excessive current consumption. The DP8515 provides ECL 10k compatible outputs, while the DP8516 is ECL 100k compatible. Parallel loading of the VSR is initiated by the RGP and assisted by the VCG.

DP8512 Video Clock Generator (VCG)

The DP8512 VCG, implemented in National's oxide-isolated bipolar process, provides all clocks in the system. It generates all clocks from a relatively low frequency (less than 20 MHz) crystal or external clock, simplifying system design and reducing system cost.

A two-phase MOS processor clock is supplied to the RGP and BPU; TTL clocks are generated for the RGP video refresh logic and for the VSR's FIFO control functions. The VSR's load and shift functions are controlled by ECL clocks generated by the VCG. An on-chip phase-locked loop (PLL) multiplies the reference crystal/clock in order to generate a pixel clock as high as 225 MHz. The VCG also includes another PLL for synchronizing the horizontal sync (generated by the RGP) to an external source.

PROGRAMMING MODEL

The RGP is a microprocessor combined with a concurrent video-refresh machine. This section will discuss the programmer-visible aspects of both the microprocessor and the video-refresh machine. For additional information, refer to the DP8500 RGP Programmer's Reference Manual (PRM).

The processor section of the RGP is a general purpose microprocessor with an instruction set expanded to include graphics operations. While the processor is microcode driven, certain graphics operations, notably BITBLT, line drawing and clipping, are implemented via dedicated hardware for increased throughput.

The organization of the processor, shown in *Figure 4*, reflects the duality, present in raster-graphics systems, of

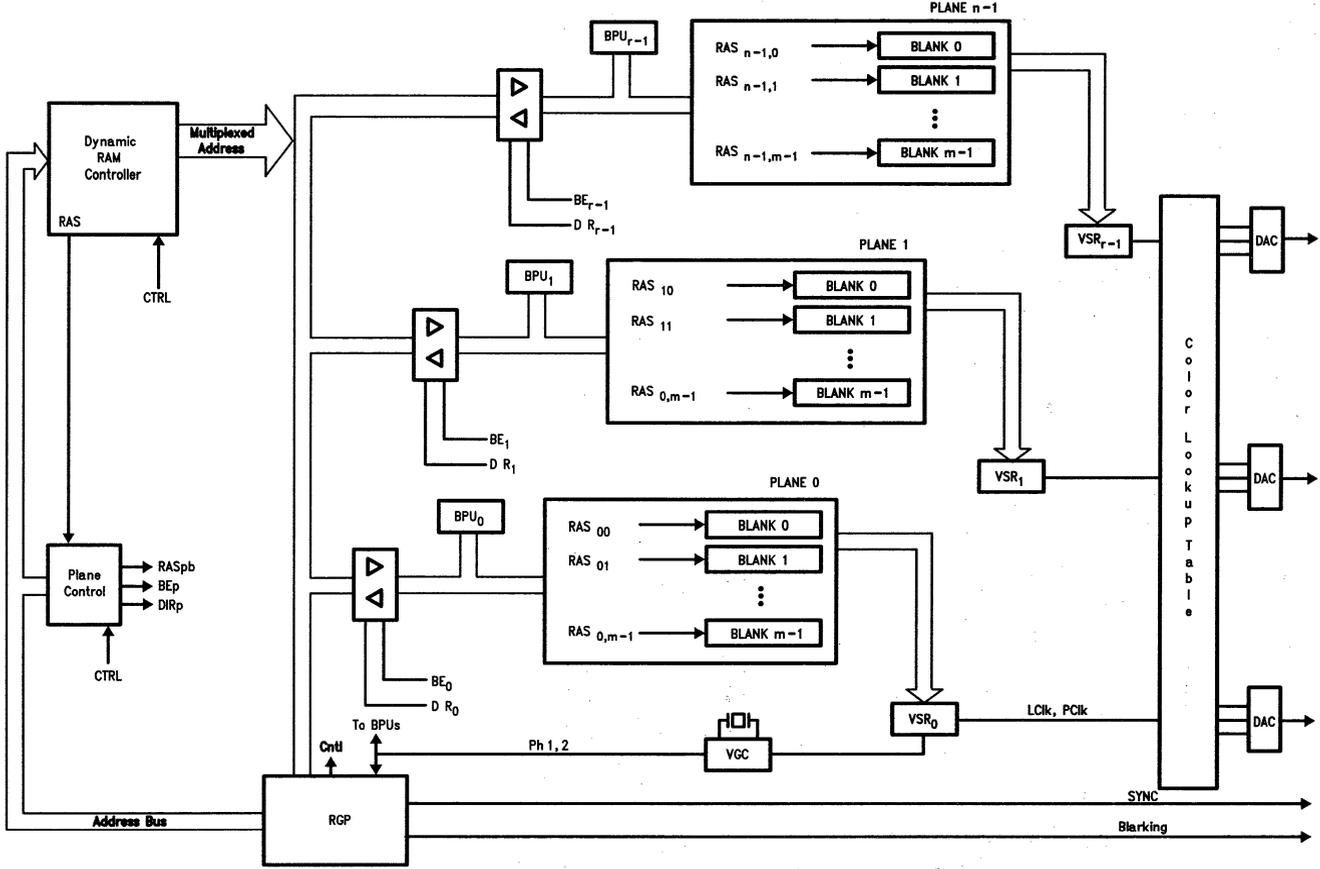
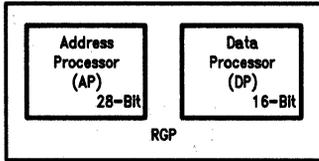


FIGURE 3. High-End AGCS-Based Color Bit-Mapped Graphics System

TL/F/9427-5



Architectural Description (Continued)



TL/F/9427-6

FIGURE 4. RGP Dual-Processor Configuration

a virtual (Cartesian, x-y) drawing space and a physical (memory address) space. That is, the RGP's processor section actually consists of two processors, the Address Processor (AP) and the Data Processor (DP), operating concurrently, driven by common microcode. The RGP's register complement, described below, is shown in *Figure 5*.

The AP consists of a 28-bit Arithmetic and Logic Unit (ALU), having a relatively simple instruction set, and a private bank of sixteen 28-bit registers. The DP is composed of a 16-bit ALU, having a relatively rich instruction set, and a bank of sixteen 16-bit registers. These instruction sets operate register-to-register only and then only within their respective register banks.

The remaining registers of the RGP have dedicated functions in support of the graphics environment or the video refresh mechanism. Examples include the registers of the Clipper and the Display Buffer Base Address register (DBB).

A single stream of instructions, fetched from external memory, serves both processors via microcode control. This includes the register-to-register instructions of both the AP and DP and the instructions for data transfers between RGP registers and memory. Additional RGP instructions, not belonging exclusively to the AP or DP, use both processors and, often, additional on-chip resources; this is typical of all of the drawing instructions. An example of this is the DRLN (draw line) instruction, which uses both processors, the clipper, the line drawing controller, the line-pattern generator and the BITBLT controller.

While simple instructions like ADD and MOV leave unmodified any registers not specified in their encodings as being operands, the more complex drawing instructions like DRLN use certain registers of the AP and DP as implicit arguments or as temporary storage. These *side effects* of RGP drawing instructions are detailed in the Programmer's Reference Manual PRM.

Memory Organization and Data Types

The RGP supports uniform 24-bit addressing; the unit of storage in memory is the word. That is, the memory space of the RGP consists of 2^{24} 16-bit words. Parts of the memory space can be designated by the user as Program, Data, Stack, Display Buffer, etc., at the user's discretion; the RGP imposes no restrictions upon these allocations.

For purposes of drawing, the RGP views the memory as being bit addressable (see Memory Addressing Section). In this case, the address is a 28-bit quantity, called a Bit Address, which consists of a 24-bit Word Address, left-shifted four bits and added to a 4-bit bit-selection field.

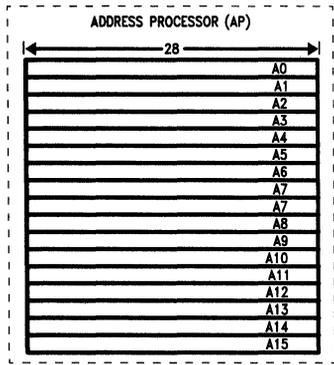
Data elements treated by the RGP vary in length (number of bits) from 3 to 28 bits according to their function. However, they are always stored right justified (that is, justified toward bit 0) in registers or in memory (see *Figure 6*). Further, when stored in memory, multiple-word quantities are always stored with the least significant word at the lowest storage address. Supporting what is known as "Little-Endian" architecture. The address of a memory-resident quantity is the address of its least significant word.

The following data element lengths are used by the RGP:

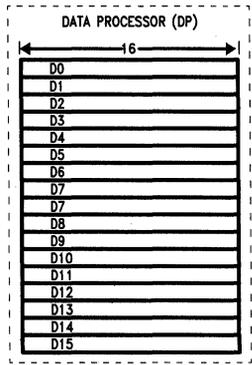
word	16 bits or less
word address	24 bits
bit address	28 bits

If an element is written from memory to a register that is shorter than the element, the more significant portion of the element is truncated to allow the less significant portion to fit into the register. If an element is written from a register to memory, the element is right justified in the memory location and unused bits are set to zero.

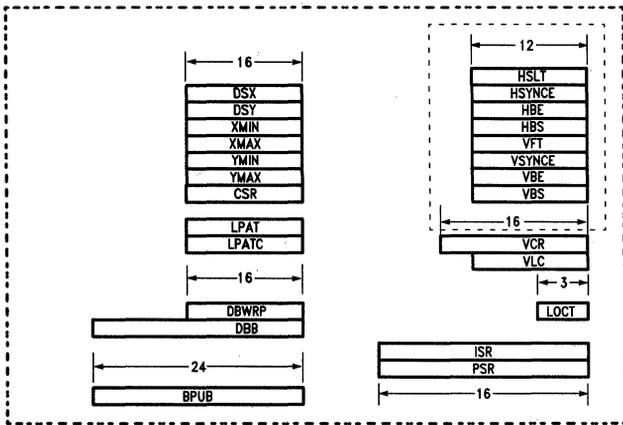
Architectural Description (Continued)



TL/F/9427-7

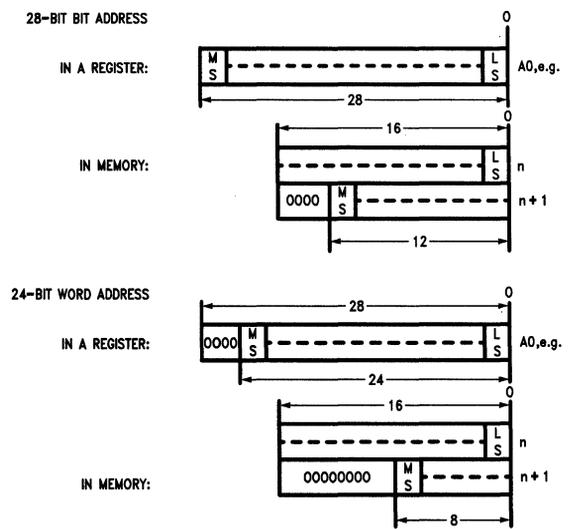


TL/F/9427-8



TL/F/9427-9

FIGURE 5. RGP Register Summary



TL/F/9427-10

FIGURE 6. Memory Storage Conventions

Architectural Description (Continued)

Address Processor Registers

The registers of the Address Processor (Figure 7) are 28 bits in length. They are used to hold operands for the Address Processor or to hold Word Addresses or Bit Addresses. Registers A0, A1 and A2 are not subject to side effects of RGP drawing instructions.

A0	
A1	
A2	
A3	FILL/TNXC
A4	MICRO/TTCH
A5	MICRO/TTHWB
A6	BSAD
A7	BSWRP/MICRO
A8	DSAD
A9	DSWRP
A10	TCDB/LDA1
A11	LDA2/TFAD
A12	INTB
A13	SP
A14	PC
A15	MICRO

← 28 →

- RESERVED FOR USER
- RESERVED FOR USER
- RESERVED FOR USER
- POLYGON FILL/CHARACTER NEXT FONT ADDRESS
- MICROCODE USAGE/CHARACTER TRAP VALUE
- MICROCODE USAGE/CHARACTER TRAP PARAMETER VALUES
- BITBLT SOURCE ADDRESS
- BITBLT SOURCE WARP/MICRO CODE USAGE
- BITBLT DESTINATION ADDRESS
- DRAWING SPACE WARP
- CHARACTER DESCRIPTOR TABLE/BRESENHAM BIT ADDRESS INCREMENT
- BRESENHAM BIT ADDRESS INCREMENT/TEXT FONT BITMAP
- INTERRUPT VECTOR TABLE BASE
- STACK POINTER
- PROGRAM COUNTER
- MICROCODE USAGE

TL/F/9427-11

FIGURE 7. Address Processor Registers

Data Processor Registers

The registers of the Data Processor (Figure 8) are 16 bits in length. All of these registers are subject to side effects of RGP drawing instructions, but D0-5 are preserved through execution of all RGP instructions except those used for polygon filling.

D0	DSXT
D1	LOCTT
D2	LLENT
D3	LDE1T
D4	LDE2T
D5	LERRT
D6	FX/MICRO
D7	FXT/MICRO
D8	MICRO
D9	LLEN
D10	LDE1
D11	MICRO/LDE2
D12	MICRO/LERR
D13	BWD/MICRO
D14	BHT/MICRO
D15	MICRO

← 16 →

- RESERVED FOR USER*
 - RESERVED FOR POLGON FILL
 - RESERVED FOR POLGON FILL
 - RESERVED FOR POLGON FILL
 - RESERVED FOR LINE LENGTH
 - BRESENHAM ERROR INCREMENT1
 - BRESENHAM ERROR INCREMENT2
 - BRESENHAM ERROR
 - BITBLT WIDTH
 - BITBLT HEIGHT
 - RESERVED FOR MICROCODE
- * except during FILL instructions

TL/F/9427-12

FIGURE 8. Data Processor Registers

Architectural Description (Continued)

Status/Control Registers

The following registers indicate or control instruction or interrupt status of the RGP. They are each 16 bits in length.

The Processor Status Register (PSR, see *Figure 9*) indicates the status returned as a result of instruction execution. It also controls the enabling of maskable interrupts and the enabling of the clipper. The individual bits are defined as follows (all fields are asserted when set to one, negated when set to zero):

- Z: Data Processor Zero bit
 - C: Data Processor Carry bit
 - N: Data Processor Negative bit
 - V: Data Processor Overflow bit
 - AZ: Address Processor Zero bit
 - AC: Address Processor Carry bit
 - K: An undrawn character code is present in register TNXC.
 - W: The current x,y is within the bounds of the clipper.
 - EIP: External Maskable Interrupt Pending
 - VIP: Video Interrupt Pending
 - CLE: Enable Clipper (1 to enable)
 - PTE: Enable Pick Trap
 - EIE: Enable External Maskable Interrupt
 - VIE: Enable Video Interrupt
- See the PRM for more information

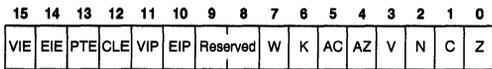


FIGURE 9. Processor Status Register

Clipper Registers

All drawing operations executed by the RGP are subject to the action of a rectangular clipper, when the clipper is enabled. The action of the clipper is to suppress the drawing of pixels outside its boundary. Clipping always takes place to bit resolution independently of the nature of the graphics primitive/operation.

That is, all lines, polygons, BITBLTs and text primitives are clipped exactly to the coordinate values defined in the clipper.

The clipper consists of six 16-bit coordinate registers and a 16-bit clipper status register, as follows:

- XMIN The x-coordinate of the left boundary of the clipper
- XMAX The x-coordinate of the right boundary of the clipper
- YMIN The y-coordinate of the top boundary of the clipper
- YMAX The y-coordinate of the bottom boundary of the clipper
- DSX The x-coordinate of the current drawing point
- DSY The y-coordinate of the current drawing point
- CSR The 16-bit Clipper Status Register

When enabled, the clipper is invoked by the RGP for all drawing instructions. The clipper may also be utilized by drawing routines written by the user. The organization of the CSR (*Figure 10*) facilitates this use of the clipper. The CSR contains the results of the four meaningful comparisons between the clipper boundary registers and the current point registers. These results are expressed in "outcode" format in the four least significant bits of the CSR.

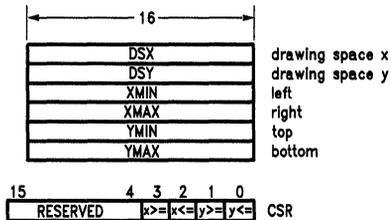
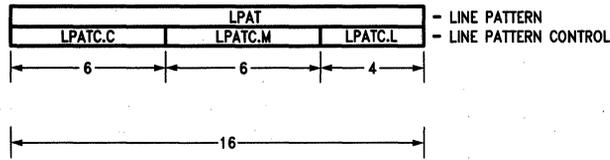


FIGURE 10. Clipper Registers

TL/F/9427-13

Architectural Description (Continued)



TL/F/9427-14

FIGURE 11. Line Style Generator

Line Style Registers

Those graphics primitives which are drawn by the RGP one pixel at a time, i.e., points, lines and polylines, are drawn subject to the action of a line pattern generator within the RGP. This mechanism produces a pattern along the length of a (presumed) line by suppressing the memory cycles associated with drawing certain points along the line, according to values contained in the registers of the line pattern generator. The line pattern generator includes a mechanism for pattern magnification. The registers of the line pattern generator (Figure 11) are defined as follows:

- LPAT 16-bit Pattern Register
- LPATC 16-bit Pattern Control Register, contains three fields:
 - LEN Pattern Length (4 bits)
 - MAG Pattern Magnification (6 bits)
 - CTR Pattern Counter (6 bits)

During pointwise drawing operations, the RGP examines the least significant bit of the LPAT register to determine whether or not to produce a memory cycle to draw the current point. A one in this position will enable drawing; a zero will inhibit drawing. At this point the memory cycle associated with drawing the current point will be executed if appropriate.

The least significant bit of LPAT remains in place for the number of points specified in the MAG field of LPATC; then a circular right shift is performed on the n least significant bits of LPAT, where n is the value specified by the LEN field of LPATC. The CTR field of LPATC serves as a counter used by the RGP to implement magnification. See the PRM for more information.

Since the length of the LPAT register is 16 bits, the maximum pattern length (assuming a magnification of one, i.e., no magnification) is 16 bits. However, shorter patterns can be implemented by setting the LEN field to an appropriate value. The length of the pattern is the number of least significant bits of LPAT that participate in the circular shift. LEN, MAG and CTR all have biases of +1. That is, values of 0 in these fields will program the line pattern generator for a pattern of length 1 and a magnification of 1.

The registers LPAT and LPATC are never reinitialized implicitly by RGP instructions. Once set, a pattern persists through all subsequent points, lines or polylines drawn.

Video Refresh Registers

The video refresh registers (Figures 12 and 13) contain the parameters for the programmable video refresh machine. They are of two types, distinguished by the mode of accessing them.

The first type is composed of eight registers that share a common register address (VIDEO); these can be accessed serially by successive MOV instructions. A 3-bit circular pointer (the VRX field of the VCR register) advances after each MOV and can be set to point to any of the eight registers.

The second type consists of 3 registers with separate addresses. The video refresh registers and their interpretations are detailed below. The eight serially addressable registers are shown first, in order. HSLT corresponds to a VRX value of 0, VBS to a value of 7.

- HSLT Horizontal Scan Line Time (12 bits)
- HSE Horizontal Sync End (12 bits)
- HBE Horizontal Blanking End (12 bits)
- HBS Horizontal Blanking Start (12 bits)
- VFT Vertical Frame Time (12 bits)
- VSE Vertical Sync End (12 bits)
- VBE Vertical Blanking End (12 bits)
- VBS Vertical Blanking Start (12 bits)

← 12 →	
HSLT	HORIZONTAL SCAN LINE TIME
HSYNCE	HORIZONTAL SYNC END
HBE	HORIZONTAL BLANK END
HBS	HORIZONTAL BLANK START
VFT	VERTICAL FRAME TIME
VSYNCE	VERTICAL SYNC END
VBE	VERTICAL BLANK END
VBS	VERTICAL BLANK START

FIGURE 12. Video Refresh Parameter Block

15	14	13-12	11-10	9-8	7	6	5	4	3	2-0
ODD	VBL	Rsvd	RAI	RM	SC	SI	SM	SE	Rsvd	VRX

FIGURE 13. Video Control Register

Architectural Description (Continued)

VCR	Video Control Register (16 bits), composed of the following fields:
VRX	Video Register Index (3 bits)
SE	Scan Enable (1 bit)
SM	Master Sync (1 bit)
SI	Interlaced Mode (1 bit)
SC	Composite Sync Mode (1 bit)
RM	Video Refresh Mode (2 bits)
RAI	Video Refresh Address Increment (2 bits)
VBL	The Vertical Blanking Flag
ODD	The Odd Video Field Flag
DBB	Display Buffer Starting Address (24 bits)
DBWRP	Display Buffer Warp (16 bits)

THE GRAPHICS ENVIRONMENT

This section discusses the conventions adopted by the RGP and the resultant environment within which all graphics operations take place.

Memory Addressing

For operations, such as instruction and operand fetching, stack operations and interrupt service, the RGP accesses memory as a uniform space of 2^{24} 16-bit words, starting at address 0 and extending to address FFFFFF hex. Bytes are not directly addressable by the RGP.

During drawing operations, the RGP can access individual bits in memory by means of a 28-bit quantity called a bit address. The correspondence between a bit address and the physical bit in memory is as follows: the 24 most significant bits of the bit address provide the address of the word in memory containing the addressed bit, while the four least-significant bits of the bit address select a bit within that word. If the four LSBs are zero, bit zero (i.e., the memory bit corresponding to the AD0 pin of the RGP) is addressed. Thus, the bit-addressed memory starts at Bit 0 of Word 0 and extends linearly to Bit 15 of Word $2^{24} - 1$.

In practice, the RGP does not read and write individual memory bits directly. When it generates a 28-bit bit address, the most-significant 24 bits are used to access a 16-bit word in the linear address space, while the 4 least significant bits go to the BPU as B0-3. The BPU contains hardware to read

and write exactly one bit in the current word. In color systems with one BPU per plane, all BPUs operate in parallel, reading and/or writing the corresponding bit in each plane (that is, all the bits of a pixel). As a result, the 28-bit quantity can be viewed as a pixel address as well as a bit address, though this interpretation is system hardware dependent and may in fact be operation-dependent. This is discussed more fully in the Multiple-Bit Pixels Section.

Cartesian Drawing Space

Drawing operations performed by the RGP execute in a logical drawing space which is Cartesian. This x-y space is defined as having the origin in the upper left. Movement to the right increases the x-coordinate; movement downward increases the y-coordinate. Each axis may be a maximum of 2^{16} bits in length. Therefore, each coordinate may take on any integer value in the range of 0 to $2^{16} - 1$. Note that while the coordinate space can range from 0 to $2^{16} - 1$, vector drawing instructions (such as DRLN and DRPLN) have the requirement that a vector length not exceed $2^{14} - 1$ pixels.

A flexible mechanism controls the correspondence between the logical (x-y) location and the physical (bit) address in memory. Once the correspondence has been initialized by the user, the RGP maintains it throughout all drawing operations. The correspondence can be changed explicitly at any time.

The correspondence is shown in Figure 14. The Cartesian origin ($x=y=0$) corresponds to the hypothetical bit address **org**. Incrementing the logical x-coordinate increments the bit address. Incrementing the y-coordinate adds the contents of the DSWRP register (the *warp* of the drawing space) to the bit address. The current bit address is maintained in the DSAD register. Thus,

$$\text{DASD} = \text{org} + \text{DSY} \times \text{DSWRP} + \text{DSX}$$

The correspondence between Cartesian coordinates and bit addresses is established by explicitly loading the registers DSX, DSY and DSAD. This correspondence is maintained by the RGP during the execution of all instructions if DSWRP contains a value appropriate for the current drawing space. Note that the drawing space can coincide with or overlap the display buffer or can be entirely distinct from it, and the two spaces can have separate warps.

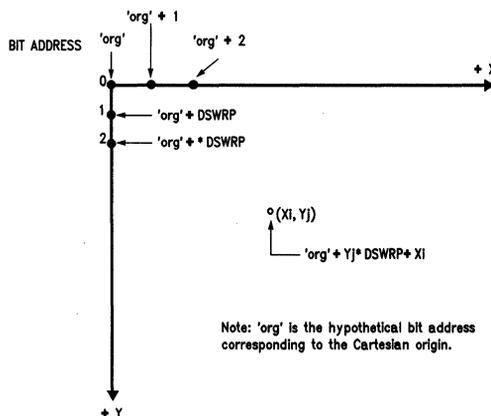


FIGURE 14. Logical and Physical Addresses

TL/F/9427-15

Architectural Description (Continued)

Dot/Line/Polyline/Polygon Drawing Operations

During the execution of the RGP's drawing instructions, the flow of data is between the memory and its associated BPU. The RGP's drawing operations can be divided into two classes, based upon the operating mode of the BPU: BITBLT or LINE, according to the state of the L/B line of the RGP and BPU(s). LINE mode is used for all pointwise drawing operations; BITBLT mode is used for all wordwise drawing operations.

Pointwise drawing operations include reading and writing individual bits (RDPT, DRPT), drawing lines (DRLN), drawing polylines (DRPLN) and drawing polygons (DRPGN). During all pointwise operations, the line pattern generator is active. The clipper (if enabled), is also active.

The clipper performs its function by suppressing the memory cycle associated with drawing the current point if that point is outside the clipper.

The line pattern generator operates in a continuous fashion within and between these instructions; it is not forced to a given state at the beginning of a pointwise drawing instruction. Therefore, a line pattern will continue around the vertices of a polyline. Also, a curve drawn with a series of DRPT instructions will be subject to the line pattern generator. Since the line pattern generator cannot be disabled, it must be loaded with a solid (all ones) pattern if a solid line (no pattern) is desired. Similarly, a pattern of all zeros will suppress all pointwise drawing.

The RDPT instruction is not affected by the clipper or line pattern generator, nor does it affect them.

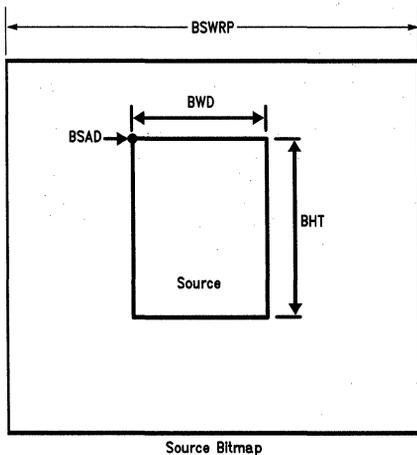
BITBLT Operations

Bit Boundary Block Transfer (BITBLT) operations are carried out by the RGP with the BPU(s) in BITBLT mode. BITBLT is the performance of sixteen specific bitwise logical operations between two rectangular arrays of bits, each having the same height and width and an arbitrary bit alignment (see *Figure 15*). In RGP-based color graphics systems, each DP8511 BPU participating in the BITBLT operation can be programmed to perform its logical operation independently of the others. This permits such effects as transparency and foreground-background color rendering.

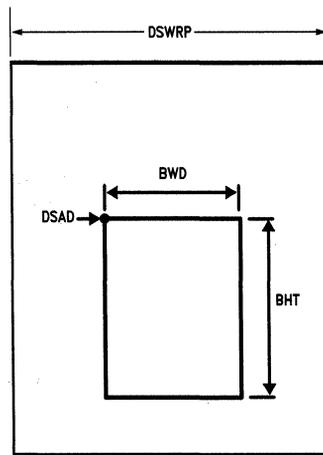
During BITBLT, the RGP produces all memory (word) addresses and the set of BPU control signals necessary to control the FIFO and masking functions. The BPU is responsible for assembling and shifting source words, storing them in the FIFO, receiving destination words, logically combining them with the corresponding shifted source words, (masking any destination bits necessary) and returning the result to the destination.

As during pointwise operations, the clipper is active and clips the destination to pixel resolution. In the process, the BITBLT left and right masks may change from those predicted by the destination alignment alone. Thus, the minimum number of memory cycles required, considering the clipper, is always performed.

The two directions of execution of BITBLTs can be specified independently: right to left (versus left to right) and bottom to top (versus top to bottom).



TL/F/9427-16



Destination Bitmap

TL/F/9427-48

- BSAD — 28 bits, bit address
- BITBLT Source Address
- DSAD — 28 bits, bit address
- Drawing Space Address

- BSWRP — 24 bits, word count
- BITBLT Source Warp
- DSWRP — 24 bits, word count
- Drawing Space Warp

- BWD — 16 bits, bit count
- BITBLT Width
- BHT — 16 bits, line count
- BITBLT Height

- BWD(source) = BWD(dest)
- BHT(source) = BHT(dest)

FIGURE 15. BITBLT Parameter Definitions

Architectural Description (Continued)

The reading of the source rectangle and the reading of the destination rectangle can be suppressed by the user independently of one another. This should be done after taking into account the requirements of the complete set of logic operations being performed by the BPU's in the system. For example, if no function other than Function 0 (fill with zeros) or Function 15 (fill with ones) is being performed, both source and destination reads may be suppressed. If a two-operand function, such as XOR, is being performed by any of the BPU's, both the source and destination must be read. Various intermediate cases may arise. the leftmost and rightmost words of the destination are always read, since this is necessary in order to use the BITBLT left/right masks.

Each BITBLT rectangle (source and destination) has a starting address (BSAD and DSAB) and a warp (BSWRP and DSWRP); they have a common width and height (BWD and BHT). These must be initialized prior to executing the BITBLT instructions. Additionally, DSX and DSY must be in correspondence with DSAD. The use of independent warps in the source and destination spaces permits packing and unpacking of bitmaps (e.g., for fonts), providing a better environment for memory pool management and generally leading to more efficient use of memory.

Polygon-Fill Operations

The RGP supports the drawing of pattern-filled vertically-convex polygons via the FILLA and FILLT instructions. (See the PRM for more information.) The argument of the FILLA instruction is a list of vertices in counter-clockwise drawing order. The first (and last) vertex of the polygon is the current point, DSX, DSY, which must be an uppermost vertex of the polygon. The RGP's filling algorithm traverses the data structure, determining the polygon's intercept points with each

horizontal line in the vertical extent of the polygon. Each line is filled, using a pattern contained in the BPU's FIFO.

The FILLT instruction provides a similar mechanism for filling trapezoids with horizontal top and bottom edges. The key feature of this instruction is that the lines that form the right and left edges of the trapezoid need not start and end at the intersections with the top and bottom edges of the trapezoid.

Each horizontal line used in filling a polygon is treated as a BITBLT of one-bit height. The first horizontal line is filled by logically combining the first word from the BPU's FIFO with the successive words of the horizontal line. the second line is filled, in the same manner, utilizing the next word from the BPU's FIFO, etc. The FIFO read is non-destructive; as a result, the pattern repeats vertically every 16 words. Since the FIFO is 16 bits wide, the pattern is 16 by 16 bits.

Text Operations

Text primitives are handled by the RGP within an environment designed to support flexibly and efficiently such features as multiple fonts, multiple type sizes and styles, sub- and superscripts and proportional spacing. This environment maps cleanly into the logical x-y drawing space and does not require dedicated text planes.

Text is rendered from monochrome or full color storage into the drawing space by BITBLT. Therefore, a bitwise logical operation takes place between the text and the current information in the drawing space. The text instructions determine from the current state of the text environment (a set of pointers and data structures) the necessary BITBLT parameters, perform the BITBLT (subject to bitwise clipping, if the clipper is enabled) and update the state of the text environment in preparation for the next text instruction.

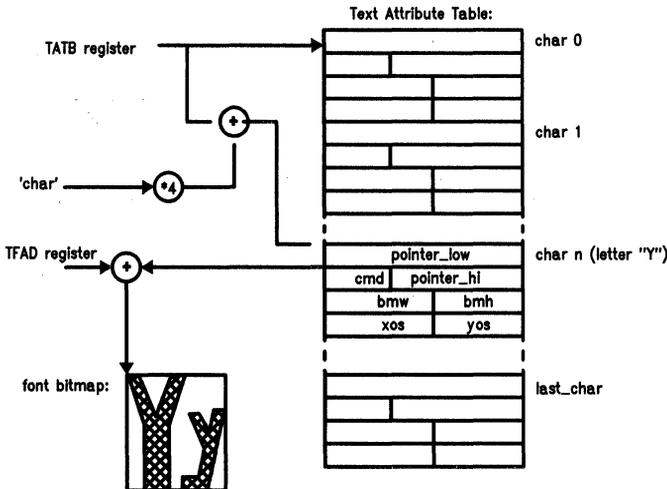


FIGURE 16. Text Address Calculations

TL/F/9427-17

Architectural Description (Continued)

The text environment is maintained in the AP as follows:

- TFAD : Text Font Base Address (A11)
- TNXC : Next Text Character (A3)
- TCDB : Text Character Descriptor Table Base (A10)

The major data structure used in support of text is the Text Character Descriptor Table (Figure 16). This table contains a four-word entry for each character in the current font. Character codes are interpreted by the text instructions as offsets into the table. This gives the RGP access to the source bitmap of the character and additional information used in rendering the character and updating the text environment.

The entry consists of:

- POINTER A relative pointer (28 bits) to the source bitmap
- key A field specifying the interpretation of the table entry:
 - Render as portrait mode
 - Render as landscape mode
 - Trap through INTB + 8
- WD Source bitmap width (8 bits)
- HT Source bitmap height (8 bits)
- DX X-offset (signed, 8 bits)
- DY Y-offset (signed, 8 bits).

If the character is to be rendered, the source bitmap is taken from TFAD + POINTER, the logical destination is (DSX + DX, DSY + DY), and the BITBLT height and width are HT and WD.

The environment will be updated as follows in the portrait mode:

$$\begin{aligned} \text{DSX} &= \text{DSX} + \text{DX} + \text{WD} \\ \text{DSY} &= \text{DSY} \\ \text{DSAD} &= \text{DSAD} + \text{DX} + \text{WD} \end{aligned}$$

The environment will be updated as follows in the landscape mode:

$$\begin{aligned} \text{DSX} &= \text{DSX} \\ \text{DSY} &= \text{DSY} + \text{HT} + \text{DY} \\ \text{DSAD} &= \text{DSAD} + (\text{HT} + \text{DY}) * \text{DSWRP} \end{aligned}$$

The subroutine call (trap) option is a general-purpose escape mechanism for handling EOT characters, line feeds, characters larger than 256 by 256 pixels, etc. When the key field indicates *trap*, the RGP does not interpret the other fields of the entry, leaving them free for other purposes.

Characters can have eight-bit codes or sixteen-bit codes. Eight-bit characters can be packed into sixteen-bit words and unpacked by means of the DCL or DCH instruction.

See the PRM for more information.

Multiple-Bit Pixels

The RGP hardware supports high-speed rendering of graphics primitives into a drawing space. In the process, it maintains the correspondence between logical and physical memory. This correspondence is independent of pixel size (depth, number of bits per pixel). In fact, the RGP architecture has no parameters for pixel size. This system-level parameter is effectively a hardware/software layer applied above the RGP architecture. This allows designers of AGCS-based systems some flexibility in trading off cost and performance.

The lowest cost approach is to use a single BPU per system, time-multiplexing it across the bit planes. Each graphics primitive must be rendered once in each bit plane. This can be done by assigning a block of memory addresses to each bit plane, and considering each bit plane as an independent drawing space. By changing the physical address corresponding to the drawing origin (modifying DSAD, typically by means of the SETPT instruction) the same display list can be executed in each plane.

The highest performance solution is obtained by using one BPU for each plane. The display list is executed once, with all planes being updated simultaneously. One way to implement this is for each plane to have a unique block of addresses where the corresponding words in each plane have addresses whose m least-significant bits match, where 2^m is greater than or equal to the size of the plane. Then the higher-order address bits (above $m - 1$) can be decoded to select the plane, and external logic can address the planes individually or in parallel (selecting some or all of the planes simultaneously). This requires a mechanism to isolate or connect, as required, the local data buses of the individual planes, typically a TRI-STATE® buffer per plane, plus the logic to control the buffers and to distribute memory control signals, typically RAS, among the planes. This logic is referred to as the video plane control logic.

An intermediate solution might use a BPU for every other plane, multiplexing each BPU two ways. In this case, the display list would be executed twice. Other intermediate solutions are possible as well.

VIDEO REFRESH

The RGP's video refresh logic is designed to support a variety of raster graphics applications, including non-video applications such as laser printers. This is accomplished by means of flexible programming of the video parameters and through a section of video refresh modes.

The user must determine the appropriate video parameters for the chosen display device, additionally taking into account the chosen values for PCLK and LCLK (the pixel

Architectural Description (Continued)

and load clocks, respectively). These values are then written to the video parameter block.

Additional parameters associated with the video refresh logic are contained in the RGP's Video Control Register (VCR).

Signal Definitions

The RGP's video refresh logic is driven by a single signal, LCLK. From this clock, according to its operating modes, the RGP generates the signals required to directly control a video monitor: Horizontal Sync, Vertical Sync and composite Blanking. Bus cycles required for video refresh purposes are also initiated by this logic.

Horizontal Sync causes the monitor to initiate the horizontal retrace period. Vertical Sync causes the monitor to initiate the vertical retrace period. Blanking causes the monitor to shut off the video output, in order to prevent writing to the screen during retrace periods. This prevents spurious retrace lines from appearing in the display.

Video Parameter Definitions

The correspondence between the register-resident parameters and the video waveforms is depicted in Figure 17. Horizontal parameters are expressed in units of LCLKs; vertical parameters are expressed in units of scan lines.

The video refresh logic is a pair of counter-driven machines. The first, responsible for horizontal sync and blanking, increments on each LCLK. Its counting sequence is 0,1, . . . HSLT,0,1 As a result, a scan line (including sync) is HSLT + 1 LCLKs in duration.

The second counter-driven machine is responsible for generation of the vertical sync and blanking. It increments once per scan line, on the LCLK during which the horizontal counter is cleared. Its counting sequence is 0,1, . . . VFT - 1,0,1 As a result, a frame (including sync) is VFT scan lines in length.

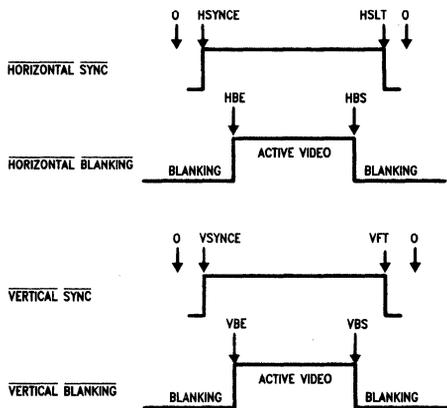
Refer to the PRM for more information.

Interlaced Refresh

The video refresh logic may be set up for interlaced scan mode by setting the SI-bit of the Video Control Register (VCR). In this mode, the screen is refreshed in two fields, the even and odd fields.

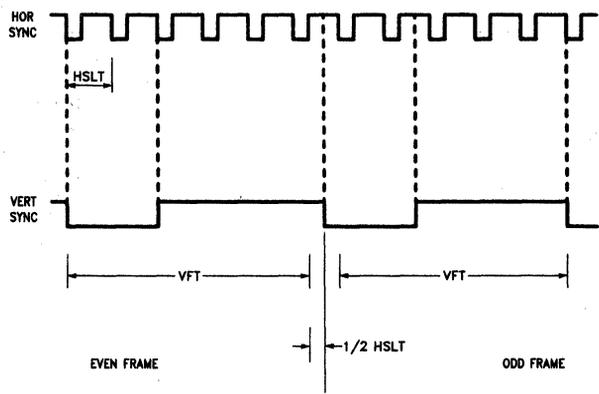
The even field consists of all even-numbered lines; the odd field consists of all odd-numbered lines.

Interlaced sync is generated by a slight modification to the vertical sync as shown in Figure 18.



TL/F/9427-18

FIGURE 17. Video Parameter Definitions



TL/F/9427-19

FIGURE 18. Interlaced Video Timing

Architectural Description (Continued)

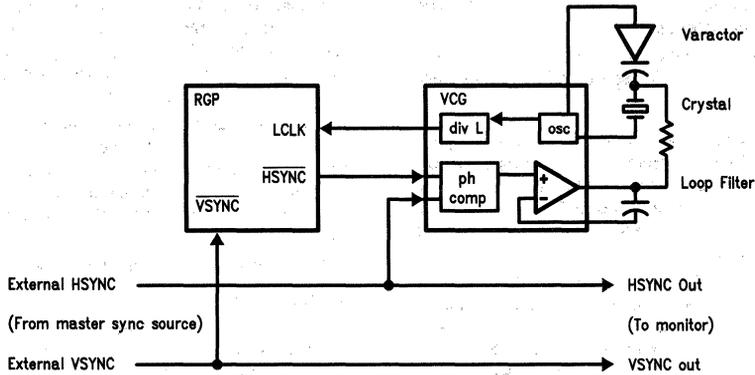


FIGURE 19. External Synchronization (Genlock)

TL/F/9427-20

Video Refresh Modes

The RGP provides four operating modes for the video refresh logic. The fundamental difference among the various modes is the way in which they allocate the available memory bandwidth between update (drawing) operations and video refresh operations. In this context, "drawing operations" refers to ALL operations other than video refresh. The mode is selected by means of the RM field of the Video Control Register (VCR).

- Mode 0** No memory cycles are generated for refresh purposes. All memory bandwidth is available for drawing.
- Mode 1** During active (non-blanked) video time, each LCLK causes the RGP to request one bus cycle for refresh purposes. The request is indicated by the RGP's assertion of DRREQ (Display Refresh Request) output. Drawing operations may take place at all other times.
- Mode 2** During active video time a bus cycle is requested on each LCLK, alternating between video refresh cycles and drawing cycles. During blanking, only drawing cycles are generated.
- Mode 3** A refresh cycle is requested by the RGP at the beginning of horizontal sync, in preparation for the next scan line. A refresh cycle will also be requested when the refresh address falls on a 256 word boundary. Drawing cycles can be generated at all other times. This mode is intended to support Video DRAMs.

External Sync Sources

The video refresh can be programmed for synchronization to an external source via the SM-bit of the Video Control Register (VCR). When in the Master Sync mode, the RGP generates all syncs and blanking signals with no external reference (except LCLK).

When in Slave Sync mode (see Figure 19), the RGP's Vertical Sync line becomes an input and is driven by the vertical

sync from an external source. The negative transition of this input will clear the vertical counter in the RGP's video refresh logic. Thus, the RGP is forced into vertical sync with the outside source.

A phase-locked loop in the VCG forces the RGP's Horizontal Sync output into synchronization with the externally-supplied horizontal sync by adjusting the frequency of LCLK (indirectly; the RGP's processor clock is actually adjusted directly. LCLK is divided down from the processor clock).

INSTRUCTION SET

Addressing Modes

The Address and Data processors of the RGP are register-to-register machines. MOV instructions are included in the instruction set to provide a mechanism for transfers between registers and memory. MOV instructions can use the following addressing modes, which are also supported, to varying degrees, by the remainder of the instruction set:

- Immediate** The operand is contained in the word(s) of memory immediately following the instruction.
- Direct** The operand is contained in the memory location(s) pointed to by the 24-bit address quantity contained in the two memory words immediately following the current instruction. The least significant word of the address is stored at the lowest address.
- Register Relative** The operand is contained in the memory location(s) pointed to by the sum of the contents of the indicated AP register and the two's complement displacement quantity contained in the word immediately following the current instruction.
- Register Indirect with Post-Increment** The operand is contained at the memory location(s) pointed to by the contents of the indicated AP register, subject to the following convention: The AP register is incremented, by the number of words in the operand, after performing the operation.

Architectural Description (Continued)

Register The operand is contained at the memory location(s) pointed to by the contents of the indicated AP register subject to the following convention: The AP register is decremented, by the number of words in the operand, prior to performing the operation.

Instruction Set Summary

The instruction set of the RGP can be divided into the following categories:

Load and Store Instructions Instructions that provide transfers between registers and memory.

General Processing Instructions Register-to-register operations that take place in either the AP or DP.

Program Control Instructions Instructions that control the flow of programs and the general software environment.

Graphics Instructions Instructions that draw or that control drawing parameters only.

Load and Store Instructions

MOV Load register from memory or store from register into memory.

General Processing Instructions

The General Processing instructions are those that execute within either the AP or the DP, or from DP to the U Bank or vice versa. They utilize register-to-register addressing only. All of these instructions will execute in the DP; some will execute in the AP as well. Both operands must belong to the same processor, except for those that act on registers from the DP and U Bank.

ADD Add (AP or DP)

ADDC Add with carry

AND Bitwise And

CMP Compare

DEC Decrement (AP or DP)

INC Increment (AP or DP)

MOV Move within DP

Move within AP

Move between DP and U Bank

EX Exchange registers within AP
Exchange registers within DP
Exchange registers between DP and U-bank

EXLN Exchange six registers used in line drawing with alternate register block

MULS Multiply signed

MULU Multiply unsigned

NOT Bitwise Complement

OR Bitwise Or

ROL Rotate Left with Carry

RORC Rotate Right with Carry

SHL Shift Left

SHRA Shift Right, Arithmetic

SHRL Shift Right, Logical (AP or DP)

SUB Subtract (AP or DP)

SUBC Subtract with Carry

TEST Test with Mask

XOR Bitwise Exclusive-Or

Program Control Instructions

The Program Control instructions affect the program flow by causing non-sequential instruction execution or by suspending processing. A number of these instructions execute conditionally; these are indicated by the **cc** field in their mnemonics, which is understood to mean one of the following interpretations of the condition code bits of the PSR:

Z	Zero	NZ	Not_Zero	(DP)
C	Carry	NC	No_Carry	(DP)
V	oVerflow	NV	No_oVerflow	(DP)
H	Higher	NH	Not_Higher	(DP)
N	Negative	NN	Not_Negative	(DP)
GT	Greater Than	LE	Less or Equal	(DP)
LT	Less Than	GE	Greater or Equal	(DP)
AZ	ApZero	NAZ	Not_ApZero	(AP)
AC	ApCarry	NAC	No_ApCarry	(AP)
W	Within	NW	NotWithin	(Clipper)
K*		NK†		

*(K = 1 implies "character available in TNXC")

†(NK = 1 implies "NOT" character available in TNXC)

Architectural Description (Continued)

ACKVI	Acknowledge VIDEO Interrupt (clear VIP)
BRKn	Break (trap) n
Bcc	Branch conditionally
BR	Branch unconditionally
CALLcc	Call subroutine conditionally
CALL	Call subroutine unconditionally
HALT	Halt instruction execution
INITB	Initialize BPU(s), i.e., force RSTO low for two clock periods
NOP	No operation
RET	Return from subroutine unconditionally
RETcc	Return from subroutine conditionally
RETI	Return from interrupt

Graphics Instructions

The Graphics instructions include two classes: those that actually draw, that is, those by means of which the RGP modifies memory in the drawing space by means of the BPU(s), and those that do setup in preparation for drawing. The setup instructions are considered first in the following listing.

The RGP maintains the concept of *current point*, analogous to the current location of the pen on a conventional plotter. Certain graphics instructions can be performed in one of two modes: relative to the current point (*relative*) and relative to the origin (*absolute*). Instructions that can be performed in either mode are shown below in both mnemonic forms. The *A* suffix indicates absolute mode.

SETPT	Set drawing point, given DSAD,
SETPTA	DSWRP, x and y
SETPTS	Set source point, given BSAD, BSWRP, x and y
SETLN	Calculate and retain parameters for Bresenham line-drawing algorithm, but do not draw line
SETLNA	
BT <i>ulsd</i>	Perform BITBLT of size BWD and BHT, from source at BSAD with warp
<i>u</i> = {UD}	BSWRP, to destination at DSAD with
<i>l</i> = {LR}	warp DSWRP. <i>u</i> and <i>l</i> control BITBLT
<i>s</i> = {MB}	direction. <i>s</i> and <i>d</i> control reading
<i>d</i> = {CW}	source and destination.
DRPT	Draw point at DSAD
RDPT	Read Point
DRLN	Draw line
DRLNA	

DRLNS	Draw line using previously-calculated parameters (see SETLN/SETLNA)
DRPLN	Draw polyline
DRPLNA	
DRPGN	Draw polygon
DRPGNA	
FILLAd	Fill polygon. <i>d</i> controls reading the destination during the fill.
FILLTd	Fill trapezoid, using previously established register values. <i>d</i> controls reading the destination during the fill.
DCHuld	Draw character high. <i>u</i> , <i>l</i> and <i>d</i> control BITBLT direction and reading the destination
DCLuld	Draw character low. <i>u</i> , <i>l</i> and <i>d</i> control BITBLT direction and reading the destination
DCNuld	Draw character, next. <i>u</i> , <i>l</i> and <i>d</i> control BITBLT direction and reading the destination
DCWuld [An++]	Draw character, word. <i>u</i> , <i>l</i> and <i>d</i> control BITBLT direction and reading the destination

INITIAL OPERATION

When the RGP is placed into the reset state (see RSTI), several registers revert to known states, and instruction execution begins. The following registers are initialized:

PC	Points to location 0
PSR	Bits VIE, EIE, PTE and CLE are zero
VCR	Fields VRX, SE, SM, SI, SC, RM and RAI are zero

Since interrupts cannot be processed properly before certain initialization has been performed, all interrupts are disabled when the RGP is placed into the reset state. Maskable interrupts must be enabled explicitly, while the non-maskable interrupt becomes enabled as soon as a new value is stored into PSR.

The RGP begins execution by alternately fetching and executing instructions starting at address 0. Typically, this initial code is responsible for establishing base pointers for data and stack areas and generally establishing the software environment, as with any microprocessor. Next, necessary data structures and pointers are initialized to support the graphics environment. Any peripherals that reside in the RGP's memory space can be initialized at this point. Finally, the internal video refresh controller is programmed in a manner consistent with the CRT monitor being used (if any) and is enabled.

Architectural Description (Continued)

Operation beyond this point is highly implementation-dependent. In a workstation application, the RGP might execute a communications protocol with another processor upstream in the graphics pipeline, awaiting the arrival of a display list to be executed. Upon receipt, the RGP would directly execute or interpret the display list, rasterizing graphics primitives into the display buffer. Upon executing the final display list instruction, the RGP would typically signal completion, thus completing the protocol with the upstream processor and allowing the process to continue.

Alternatively, in a standalone application like a graphics terminal, the RGP might enter a control program, servicing peripherals and executing a command interpreter. Here, the RGP would be responsible for keyboard, mouse and UART service; at the same time, the RGP might execute a graphics language interpreter, responding to remote host commands by maintaining a graphics environment and drawing into the display buffer.

SYSTEM-LEVEL IMPLEMENTATION OF THE DP8500 FROM A HARDWARE PERSPECTIVE

In order to maintain high efficiency in drawing operations, memory access overhead must be minimized. One of the most efficient and economical techniques for reducing memory access overhead is through use of the pagemode memory access principle. Here, a memory page physically maps to a selected DRAM row, or page. Design of the RGP was made with the pagemode principle in mind. All BITBLT operations take place in a burst format which directly supports pagemode. This process is realized by a sixteen word deep FIFO located in the core of the BPU. During the BITBLT process, the FIFO serves as a temporary storage buffer for storing one operand of the BITBLT. As drawing operations take place, the RGP asserts a pagebreak detect mechanism which serves as a system end of page detector. It is through system use of this system-level indicator that efficient transitions from memory page to page come about.

In addition to the various access techniques for retrieving memory data during drawing operations, a BITBLT operation must deal with the task of calculating relative bit offsets between the source and destination word locations. In situations where the relative bit offset between the source and destination is zero, the number of source reads will always match the number of destination writes. BITBLT operations of this nature are known as 'linear.' However, 'non-linear' situations do exist such as the generation of an extra source read to one destination write. The converse of this is also true: where one source read is accompanied by two destination writes. A set of rules describes the RGP principle of BITBLT:

1. Only the minimum required number of destination writes, or destination read-modify-writes will be generated in order to perform a given BITBLT.
2. Read-modify-write operations will always be generated on the left and right BITBLT edge boundaries.
3. The appropriate masking function(s) are always applied on BITBLT edge boundaries.
4. In situations where the width of the BITBLT is 16 bits or less, a read-modify-write occurs, and both the Left and Right Masks will be applied.

A number of parameters exist in the BITBLT (BT) and FILL instructions. These parameters instruct the RGP on which

bus cycles to produce. In addition they determine the generation of the FIFO Read and Write strobes. In BITBLT operations which require BITBLT operand source read(s), the RGP enters the BITBLT drawing "source" mode. Here, source data is fetched from memory in a "burst" fashion, shifted (if necessary), and stacked in the FIFO as the FWR input is strobed. Stacking continues until either the FIFO fills, or the width of the BITBLT is reached. At such time, the RGP then enters the drawing "destination" mode. For the majority of bus operations, the RGP remains in the "destination" mode, only entering the "source" mode when source drawing is specified as part of the BITBLT instruction.

The DP8511 BPU is designed to operate on the source, destination or combinations of both BITBLT operands as specified in the RGP generated bus cycles, and the function code placed in the FSE register. A typical example of an one-operand BITBLT is a direct drawing write-only operation. Here, the operand is sourced from the destination location, passed through the BPU and written back to the same location.

Two-operand BITBLTs involve reading the destination source operand into the BPU during the read portion of the RMW cycle. When the "modify" portion is reached, the FIFO output produces the second operand. Both operands are then logically combined in the BPU's Logic Unit as dictated by the function code in the FSE register. The operation is made complete by the write back to memory. In short, it is the value placed in FSE and the type of bus cycle generated which determine the number of operands used in a BITBLT. It should be understood that the purpose of a FRD strobe is to produce a *fresh* second source operand. If FRD is not strobed, the FIFO output will reflect the last FRD strobed output. The methodology for producing FRD is described as follows.

In many situations where a polygon fill is taking place, the FIFO output contents are only updated after the width of the BITBLT is reached. This results in a replicating pattern which matches the contents of the 16 x 16 FIFO.

Because the FIFO read and write (FRD and FWR) strobes play such a vital role in storing and retrieving BITBLT operands, generation of FRD and FWR are based on the following rules:

1. FWR is always generated whenever a BITBLT source read cycle is generated,
2. If a BITBLT source read occurred, then FRD will automatically be generated for either a destination read-modify-write or direct memory write only operations,
3. If the BITBLT source read cycle is suppressed, no FRDs will be generated for the destination portion of the BITBLT cycle.
4. The FIFO output will always reflect the current value of the FRD pointer.

See the *Programmers Reference Manual* for details.

BUS OPERATION

This section describes the various bus cycles generated by the RGP. For clarity in documentation, Bus Status refers to the RGP generated Bus Status lines (BS1-0). Read and Write refer to the RGP generated Read (RD) and Write (WR). It is to be understood that RD and WR serve as *status* only.

Architectural Description (Continued)

SYSTEM CONTROL MECHANISM

The key elements in designing with the AGCS are the RGP, BPU and Bus State Machine (BSM). As each element performs a specific task, the net result is a tightly coupled functional hardware array. The primary function of the RGP is to serve as an efficient rendering engine: generating a global address and system status. The task of the BPU is to serve as a data path channel.

With current technology, the system designer has a wide variety of system memory access techniques to choose from. Optimization comes about only if the system designer follows all the special rules which govern each memory access requirement. As technology improves, the rules change. The majority of microprocessor products on the mark today contain an embedded memory access protocol. Although the initial system hardware implementation may be somewhat more straight forward, the result often ends in a product with restricted, or inefficient access protocols. Fortunately, the underlying philosophy of the RGP is to allow the system designer the freedom to optimize specific memory access protocols for the application at hand.

In order to ensure the most efficient access protocol, optimization must be carried out by the system itself. By using the RGP as a system status generator, status information is conveyed to an external Bus State Machine (BSM) which carries out the timing and system control for a given bus status. In effect, as status is generated, the BSM becomes a hardware shell, which gives the RGP a unique system personality.

As each system is developed, a unique personality emerges which best suits the system from the standpoint of improvements in technology, and access efficiency.

The purpose of this section is to render a process whereby a system designer can understand the underlying criteria for designing the BSM. First and foremost in the design process, the designer must understand the functional role of the BSM in a system. Only when the demands of a particular system design are clearly understood is it possible to tailor the BSM to optimize memory access efficiently. It should be understood that different system designs will have different criteria. This section will cover all of the basic elements for a BSM design.

BSM DESIGN METHODOLOGY

In designing the BSM, the first stage the system designer must go through is to gain a thorough understanding of the various relationships, or roles the BSM must play in order to function with various members of the system. This procedure is best understood by concentrating on the following core issues:

1. The role of the BSM to the RPG, and visa-versa.
2. The role of the BSM to the HOST.
3. The role of the BSM to a memory refresh request.
4. The role of the BSM to the BPU.
5. And finally, the role of the BSM to the system memory.

On system reset, the BSM is expected to enter an initialization loop. The purpose of this loop is to allow the BSM to enter a known state, and to synchronize itself to the system. Once synchronized, the BSM begins to sample the Bus Status and ALE. From a system perspective, the purpose of ALE is to demultiplex the address and data lines. From a BSM perspective, ALE serves as an indication that a bus cycle has just begun. In effect, the BSM uses ALE to syn-

chronize itself to the RGP. Because the BSM understands the exact nature of the RGP and system bus latency associated with each bus cycle, it knows when to assert WAIT (if necessary) and when to place data on the system global data bus if the RGP is doing a Read or Write. The BSM must generate feedback in the form of WAIT in order to ensure that the RGP remains synchronized to the system.

Various systems also require that a HOST gain mastership of the bus. Here, the same rules apply as stated above. A bus cycle start is required (often this is generated by the assertion of the bus acknowledge signal), and WAIT is applied until the data transfer is complete.

Accessing the system bus can take place in a variety of forms. Often the HOST forces the RGP completely off the bus via the HOLD/HOLDA mechanism, other times a cycle stealing approach is taken. Cycle stealing allows the HOST to temporarily gain access to the system bus via a direct arbitration mechanism built into the BSM, in effect, directly bypassing the RGP. If a cycle stealing approach is taken the BSM must delay or WAIT any pending RGP requests during the HOST transfer.

One of the main purposes of the BSM is to arbitrate for various incoming processes at the system-level. Not only must it arbitrate for mastership, but it may also need to arbitrate for memory refresh requests. Often, the RAS and CAS functions are included as part of the BSM therefore, a protocol must be established between the memory refresh preempting device and the BSM. Upon assertion of a memory refresh request, the BSM is required to arbitrate itself for the refresh cycle, WAIT the Bus Master (should a bus cycle be pending) and perform the necessary refresh.

During BPU setup and drawing, the BSM is required to apply various strobes to the BPU. One of the main purposes of the functional timing section is to illustrate when the system strobes are asserted. For non-drawing operations, the BPU Control Register Enable (CRE) and Function Select Enable (FSE) strobes are asserted. For drawing operations, Data Latch Enable (DLE), Pixel Data Latch Enable (PDLE), Data Output Enable (DOE) and Pixel Data Output Enable (POE) strobes are asserted.

From a system-level perspective, the BSM is required to apply the following system strobes in order to execute the various system functions:

Non-drawing functions (RGP and HOST);

1. Arbitrate and execute system control Video Refresh functions,
2. arbitrate and produce system control Instruction, Operand and fetch and write access functions.

Drawing functions(RGP):

3. Arbitrate and execute system control drawing source read, and destination write operations,
4. Arbitrate and execute system control drawing destination read-modify-write operations.

Once the basic system arbitration tasks of the BSM are understood, the next step is to develop the overall structure in such a way that the access priority mechanism of both the BSM and RGP (see the section on RGP BUS PRIORITY MECHANISM) correlate. *Figure 20* illustrates the priority mechanism of the BSM: with Display Refresh being the highest priority and drawing being the lowest. This type of BSM structure guarantees that the BSM will execute high priority RGP Video Refresh requests in a timely fashion without unexpected latency or hindrance from other devices.

Architectural Description (Continued)

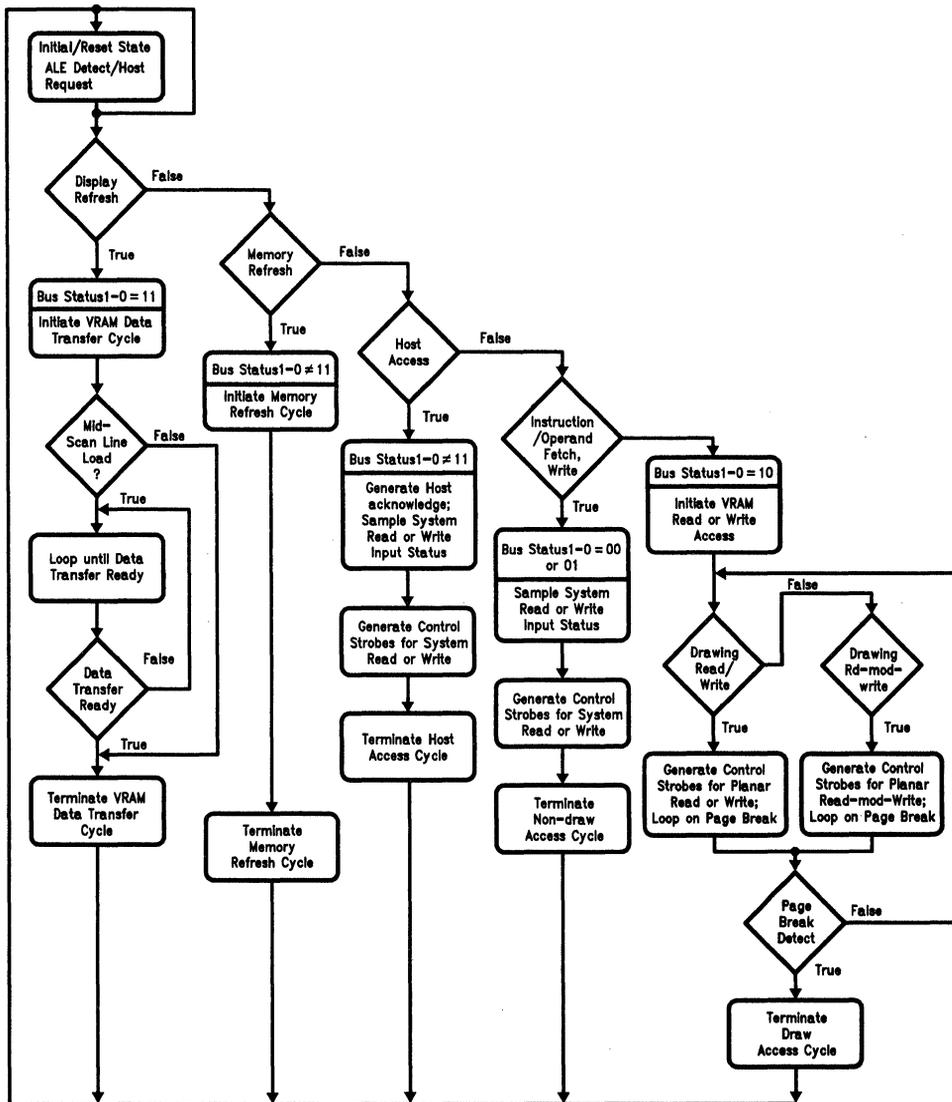


FIGURE 20. Flow Diagram of the Bus State Machine

TL/F/9427-49

Architectural Description (Continued)

Once an understanding of the role, and arbitration of the BSM is worked out, a study begins on the various loop, or branch requirements necessary for the BSM to execute a given function. It is within each of the loop that the various memory access techniques are employed. The number of states required in executing the loop itself is often reflected in the latency associated with each type of memory access.

As the design progresses, the designer gains a deeper insight as to how the various memory access begin to interact, and the functional role of the RGP Page-Break mechanism in memory burst operations. Interaction of access cycles are best understood by understanding the sequence of RGP drawing bus cycles produced, and the effects on the system memory.

Figure 20 gives the system designer a conceptual feel for both the structure and basic system functions required of the BSM. Of special interest are the Display Refresh and Drawing loops. In applications which require real-time display (or screen) refresh, data must be downloaded to the video pipe in such a way as to provide a contiguous stream of valid data to the image device to avoid image folding or tearing.

The methodology of producing real-time display refresh is known as Mid-scan line load. Including Mid-scan line load as part of the BSM requires a nested inner synchronization loop as part of the main Display Refresh loop. Under control of the RGP \overline{DRREQ} and an external counter, the inner loop of the BSM serves to synchronize the BSM to the data contents of the active video pipe. After synchronization, downloading can take place at exactly the prescribed time. Detail of this operation is covered in the application note, *Mid-Scan-Line-Load Techniques using the DP8500 Raster Graphics Processor*.

Recently, VRAM technology has introduced a new technique for loading the serial shift registers called Split Register Load. For systems employing such devices, direct Data Transfer operations can take place without the necessity for BSM system synchronization.

In summary, the BSM understands the system. It is responsible for receiving instruction from the bus master, and executing the control commands from the system-level. As technology changes, so to does the BSM. As a result the RGP allows the system designer to employ any access technique which best serves to optimize the design.

Included in the bus cycle section are a number of functional timing diagrams. The role of the functional timing diagram is to express, at the conceptual timing diagram level, the BSM placement of system control strobes for the required bus cycles.

OVERVIEW OF THE RGP BUS CYCLES

RGP Bus Operations are classified as either drawing or non-drawing and are based on the RGP Read, Write and Bus Status lines. Non-drawing bus cycles are composed of a minimum of three clock cycles. These consist of operand read and write operations which are wholly responsible for fetching instructions, maintaining stacks, queues, performing screen refresh, plane control functions and loading BPU Control and Function Select registers prior to BITBLT.

Drawing bus operations are composed of a minimum of two clock cycles which are allocated into BITBLT and Line drawing operations. Reserved for working in conjunction with the BPU(s) and the frame buffer array, the RGP serves as a bus status and address generator which is to source a variety of

drawing bus cycles such as BITBLT Source read, BITBLT Destination write, BITBLT read-modify-write, and Line read-modify-write operations.

Although the BSM is responsible for strobing the data and pixel latches, the RGP is responsible for directly strobing these particular BPU inputs:

1. B0/LME—Pixel Address Zero/Left Mask Enable,
2. B1/RME—Pixel Address One/Right Mask Enable,
3. B2/FWR—Pixel Address Two/FIFO Write,
4. B3/FRD—Pixel Address Three/FIFO Read,
5. L/ \overline{B} —Line/BITBLT mode,
6. BSE—BITBLT Source Enable.

These lines will be activated either before or interactively during the BITBLT to ensure that the frame buffer data is properly manipulated. The L/ \overline{B} input is somewhat static in the sense that this input is changed a number of clock cycles prior to actually implementing the drawing function.

NON-DRAWING BUS CYCLES

Non-drawing operation are outlined in detail as follows:

READ TRANSFER MECHANISM

Read bus cycles transfer data from memory or peripheral devices to the RGP. A no-wait, or minimum Read bus cycle is composed of three unique "T" state clock cycles of which:

1. state T1 produces ALE, address and Bus Status,
2. state T2 and T2 wait (T2(w)) maintains: upper 8-bit address, bus status and samples WAIT,
3. state T3 maintains: upper 8-bit address, Bus Status and samples data on AD15-0.

During Read bus cycles, the RGP generates the following bus status:

1. BS1 is a logic zero,
2. BS0 is a logic one,
3. \overline{RD} is a logic zero,
4. \overline{WR} is a logic one.

If necessary, Read Bus cycles can be extended on an integral clock cycle basis by the assertion of WAIT (via the RGP's WAIT input). Bus cycle extension is based on the premise that if WAIT is sampled low, (i.e., a logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state: an exact functional replica of the T2 state. Thus, T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one). State T3 follows as a natural progression after T2 or T2(w). Valid data is sampled during the PHI 2 falling edge sample window of T3.

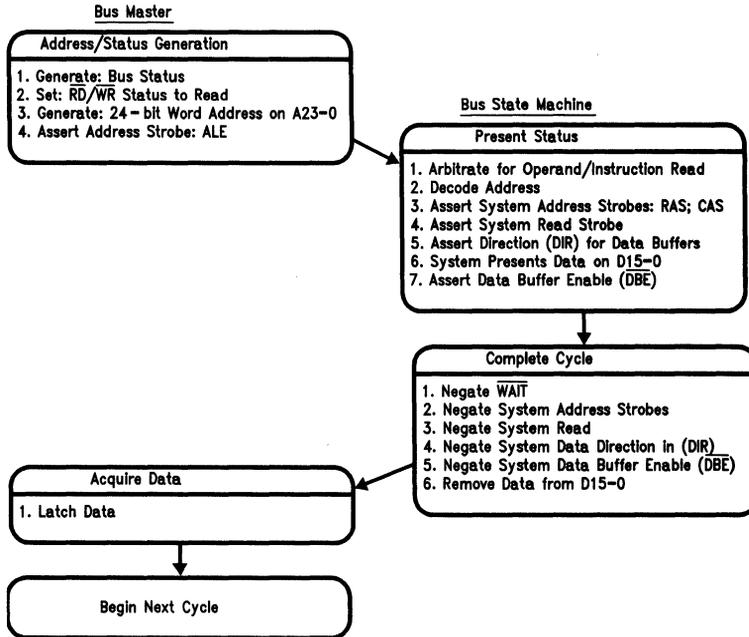
Figure 21 illustrates a flowchart of the read transfer mechanism while Figure 22 illustrates functional bus information.

WRITE TRANSFER MECHANISM

RGP Write bus cycles transfer data from RGP to memory or a peripheral device. A no-wait, or minimum Write operation is composed of three unique "T" state clock cycles of which:

1. state T1 produces ALE, address and bus status,
2. state T2 and T2 wait (T2(w)), which maintains upper 8-bit address, bus status, asserts data and samples WAIT,
3. state T3 which maintains the upper 8-bit address, bus status and drives data valid onto AD15-0.

Architectural Description (Continued)



TL/F/9427-50

FIGURE 21. Read Transfer Flow Chart

Write bus cycles generate the following bus status:

1. BS1 is a logic zero,
2. BS0 is a logic one,
3. $\overline{\text{RD}}$ is a logic one,
4. $\overline{\text{WR}}$ is a logic zero.

If necessary, Write bus cycles can be extended on an integral clock cycle basis by the assertion of WAIT. Bus cycle extension is based on the premise that if WAIT is sampled low, (i.e., a logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state which is an exact functional replica of the T2 state. T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one). State T3 follows as a natural progression with Address, Bus Status and Data reflecting the preceding T2, or T2(w) state.

Figure 23 illustrates a flowchart of the Write transfer mechanism while Figure 24 illustrates functional bus information.

SCREEN REFRESH MECHANISM

Video refresh operations transfer data from the frame buffer to the video screen refresh logic. A no-wait, or minimum Screen Refresh is composed of three unique "T" state clock cycles of which:

1. state T1 produces ALE, address and bus status,
2. state T2 and T2 wait (T2(w)) maintains: all addresses, bus status and samples WAIT,
3. state T3 maintains all addresses and bus status.

Video Refresh bus cycles generate the following status information:

1. BS1 is a logic one,
2. BS0 is a logic one,
3. $\overline{\text{RD}}$ is a logic zero,
4. $\overline{\text{WR}}$ is a logic one.

Video Refresh bus cycles can be extended on an integral clock cycle basis by the assertion of WAIT. Extension is based on the premise that if WAIT is sampled low, (i.e., a

Architectural Description (Continued)

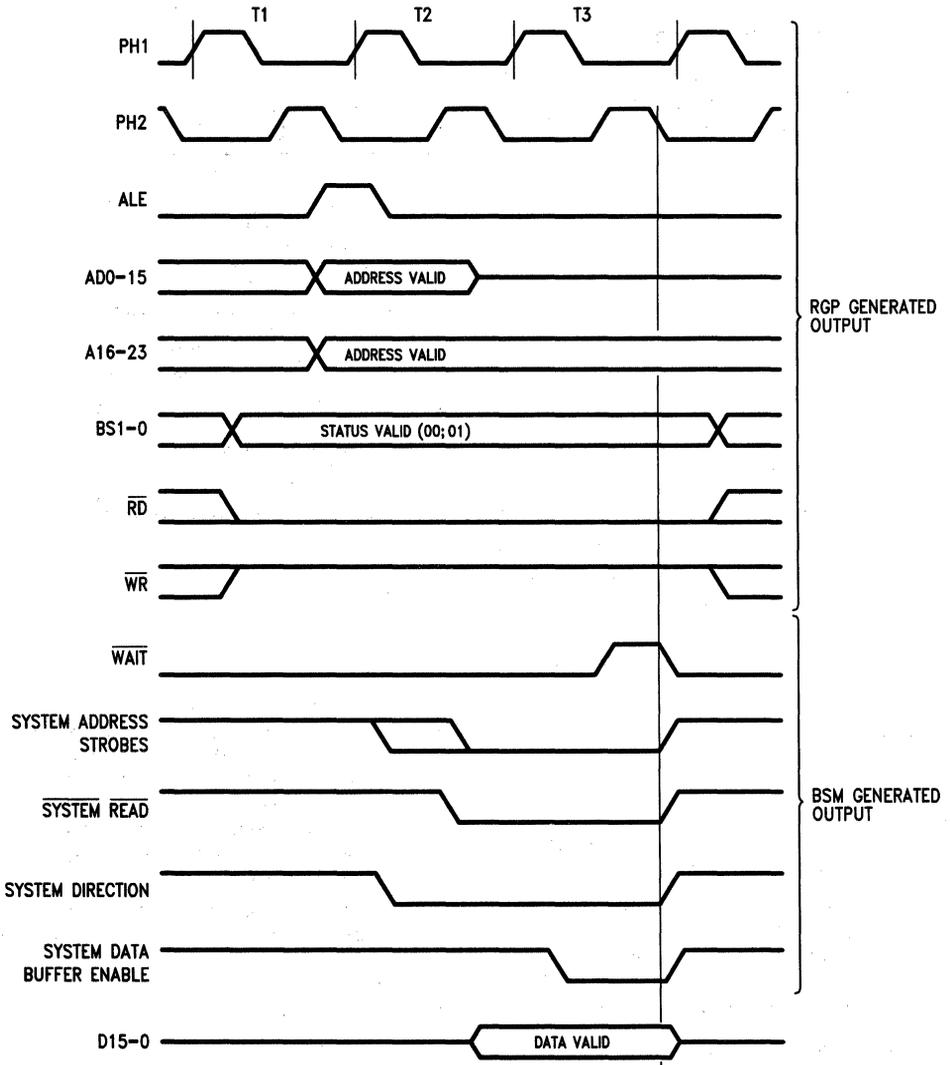


FIGURE 22. Functional Read/Operand Fetch Cycle

TL/F/9427-51

Architectural Description (Continued)

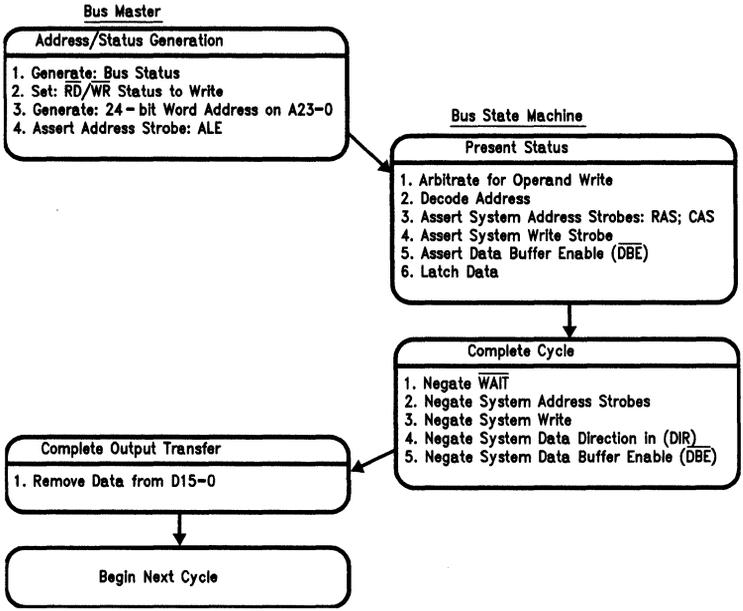


FIGURE 23. Write Transfer Flow Chart

TL/F/9427-52

Architectural Description (Continued)

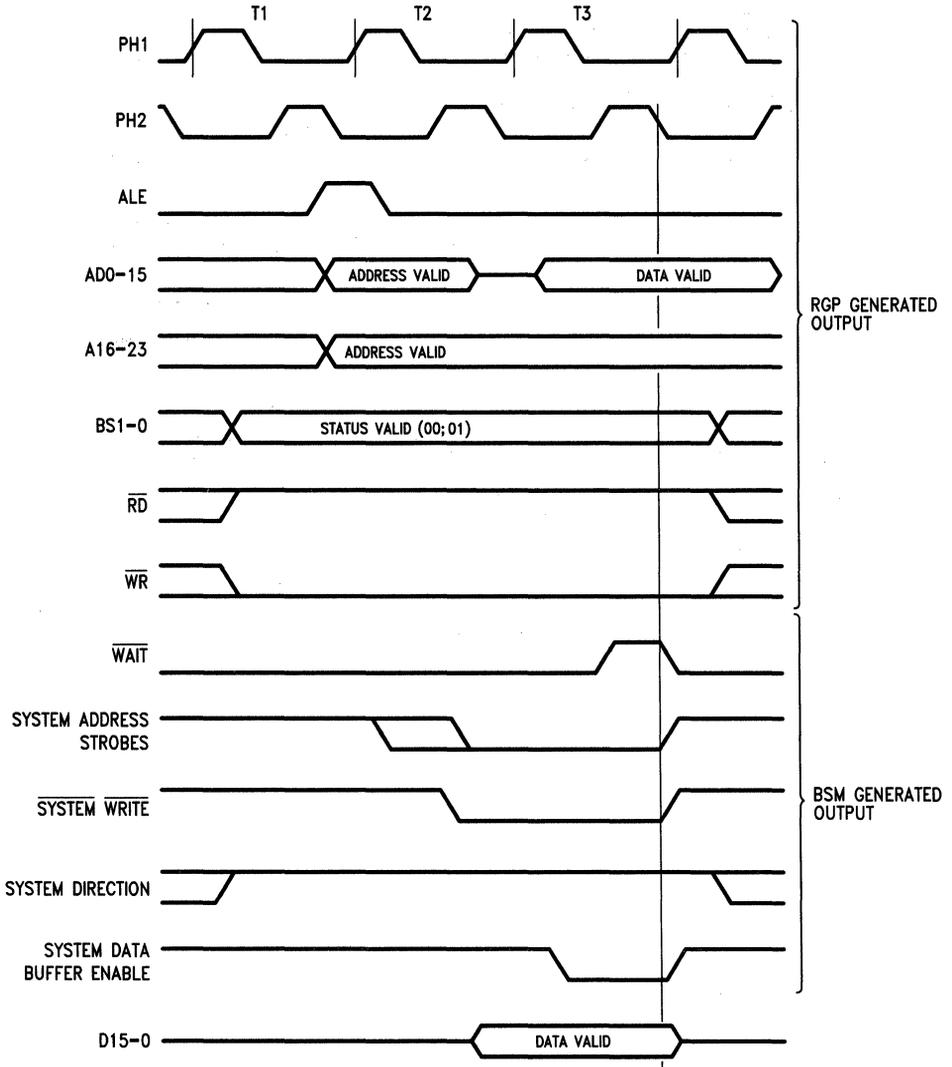


FIGURE 24. Functional Write Cycle

TL/F/9427-53

Architectural Description (Continued)

logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state: an exact functional replica of the T2 state. Thus, T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one). State T3 follows as a natural progression, with Address, Bus Status reflecting a logical equivalent of the preceding T2, or T2(w) state.

PREDICTING DISPLAY REFRESH

Prior to generating a Video Refresh cycle, the RGP will produce an output called Display Refresh Request (\overline{DRREQ}). This signal serves as a forerunner to the Video Refresh Cycle, which could be used as a high priority, or preemption mechanism at the time a HOST is granted the system bus. Because of the deterministic nature of \overline{DRREQ} , this signal serves a vital role in execution of real-time screen refresh operations known as Mid-scan Line Load. Predicting the actual start of the Display Refresh cycle is dependent on the activity of the system at the time of \overline{DRREQ} . Figure 26 illustrates a situation where a Display Refresh cycle is forced to wait until the completion of a Read bus cycle. In any given circumstance, the Read cycle could be any other type of RGP generated cycle. Often, a HOST could possibly be

bus master at the time of \overline{DRREQ} . There are two key points that one must keep in mind when the system requires Mid-Scan line load:

1. The generation of \overline{DRREQ} happens at constant intervals.
2. The start of a Display Refresh is undetermined.
3. The priority mechanism of the BSM is built such that a Display Refresh cycle will take place immediately after the current bus cycle, regardless of a pending memory refresh.
4. Up to, and including the assertion and negation of the memory Data Transfer line, valid data is in the serial data pipe.

Under most circumstances, serial data is transferred out the video shift registers by gating the RGP Blank output with the VRAM shift register clock (normally the shift register clock is LCLK). Video Pipelining is then accomplished by taking the gated output and forcing this through a clocked shift register with the same clock that drives the VRAM serial shift registers.

Figure 25 illustrates a flowchart of the Screen Refresh transfer mechanism while Figure 26 illustrates functional bus information.

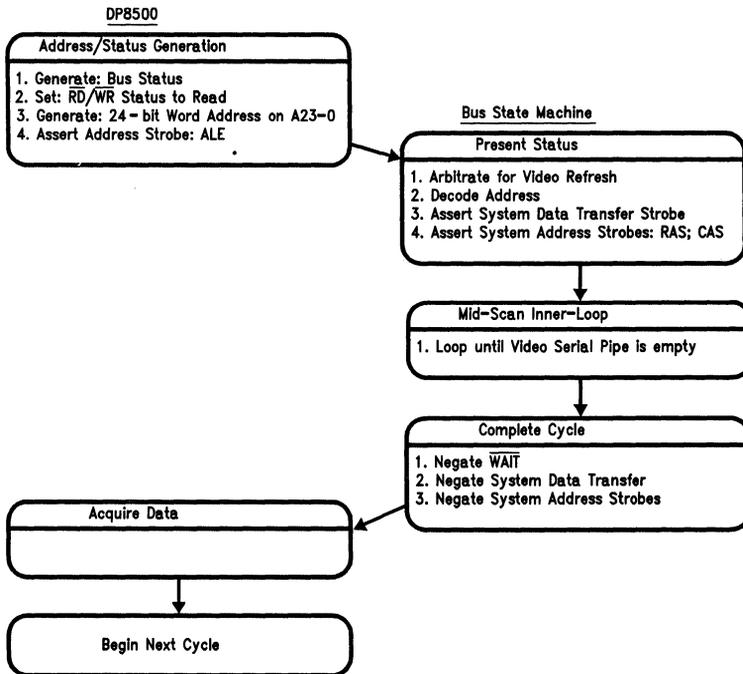


FIGURE 25. Display Refresh Transfer Flow Chart

TL/F/9427-54

Architectural Description (Continued)

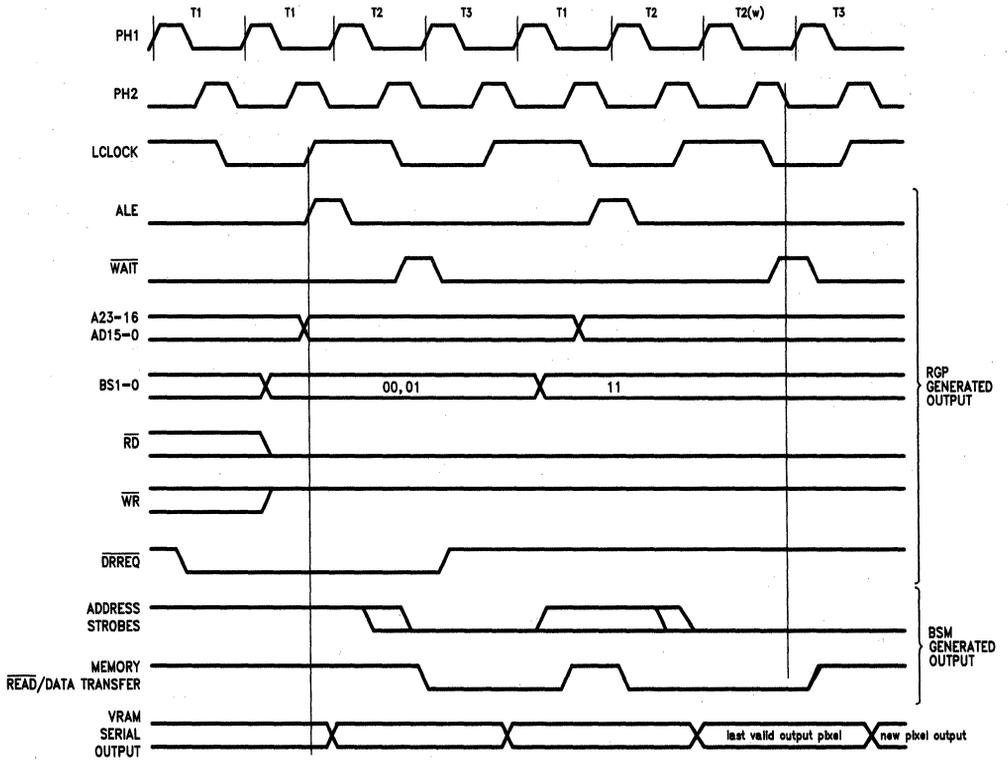


FIGURE 26. Functional Display Refresh Cycle Delayed by a Non-Drawing Read

TL/F/9427-55

Architectural Description (Continued)

DRAWING BUS CYCLES

Drawing operation is described in detail as follows:

BITBLT SOURCE READ MECHANISM

BITBLT Source Read operations transfer data from a drawing source address and place it in the FIFO of the BPU. A no-wait, or minimum BITBLT source operation is composed of two unique "T" state clock cycles of which:

1. state T1 produces ALE, address and bus status,
2. state T2 produces address, bus status, FIFO Write (FWR) and samples WAIT
3. while T2 wait, (T2(w)), maintains all addresses, bus status and samples WAIT.

During BITBLT-Source operations the RGP generates the following bus status:

1. BS1 is a logic one,
2. BS0 is a logic zero,
3. L/\bar{B} is a logic zero,
4. \bar{RD} is a logic zero,
5. \bar{WR} is a logic one.

BITBLT-Source Bus cycles can be extended on an integral clock cycle basis by the assertion of WAIT. Extension is based on the premise that if WAIT is sampled low, (i.e., a logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state. Aside from the FWR generated during T2, T2(w) is an exact functional replica of the T2 state. Thus, T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one).

Upon sampling WAIT high, the RGP will generate a FIFO write (FWR) two clock cycles later. In most cases, drawing bus cycles will occur back to back, therefore, as a result, FWR will occur during T2 of the next BITBLT Source cycle.

In order for the BPU to maintain the proper internal data pipeline, a succession of events must occur of which:

1. the RGP samples WAIT high during T2,
2. on the next clock cycle, or as the RGP enters the next bus cycle T1 state, the BSM asserts DLE,
3. on the next clock cycle, (during the next bus cycle T2) the RGP, completes the previous bus cycle by strobing FWR.

BITBLT source read operations require that the valid data be maintained long enough on the system data bus to ensure that data latched into the BPU is valid. Often, and for this reason, effective pipelining of the system results in a clean and efficient approach when dealing with BITBLT Source Read operations.

Figure 27 illustrates a flowchart of the BITBLT Source Read transfer mechanism while Figure 28 illustrates functional bus information.

BITBLT DESTINATION WRITE MECHANISM

BITBLT Destination write operations, reads data from the BPU FIFO and places the data into the specified destination address. From an RGP perspective, a no-wait, or minimum BITBLT Destination write operation is composed of two unique "T" state clock cycles, of which:

1. state T1 produces ALE, Address Bus Status and FRD,
2. state T2 produces Address, Bus Status, FRD and samples WAIT,
3. while T2 wait (T2(w)) maintains: Addresses, Bus Status and samples WAIT.

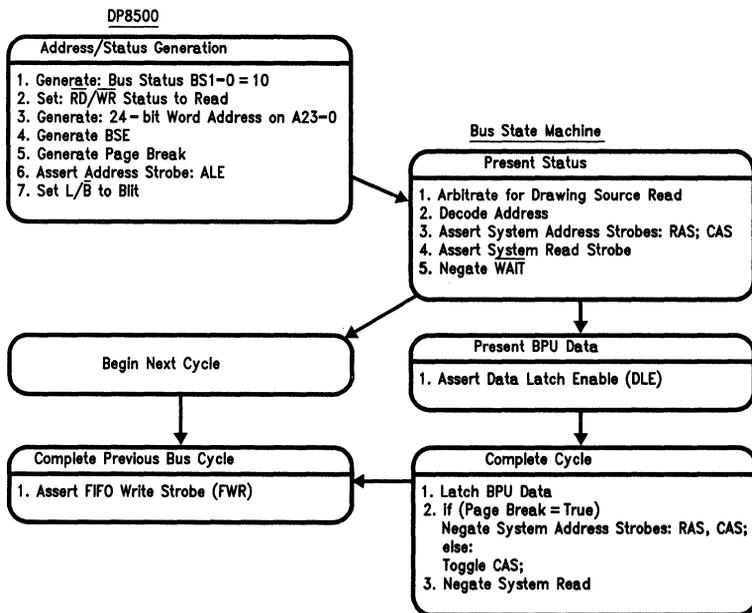


FIGURE 27. BLIT Source Read Transfer Flow Chart

TL/F/9427-56

Architectural Description (Continued)

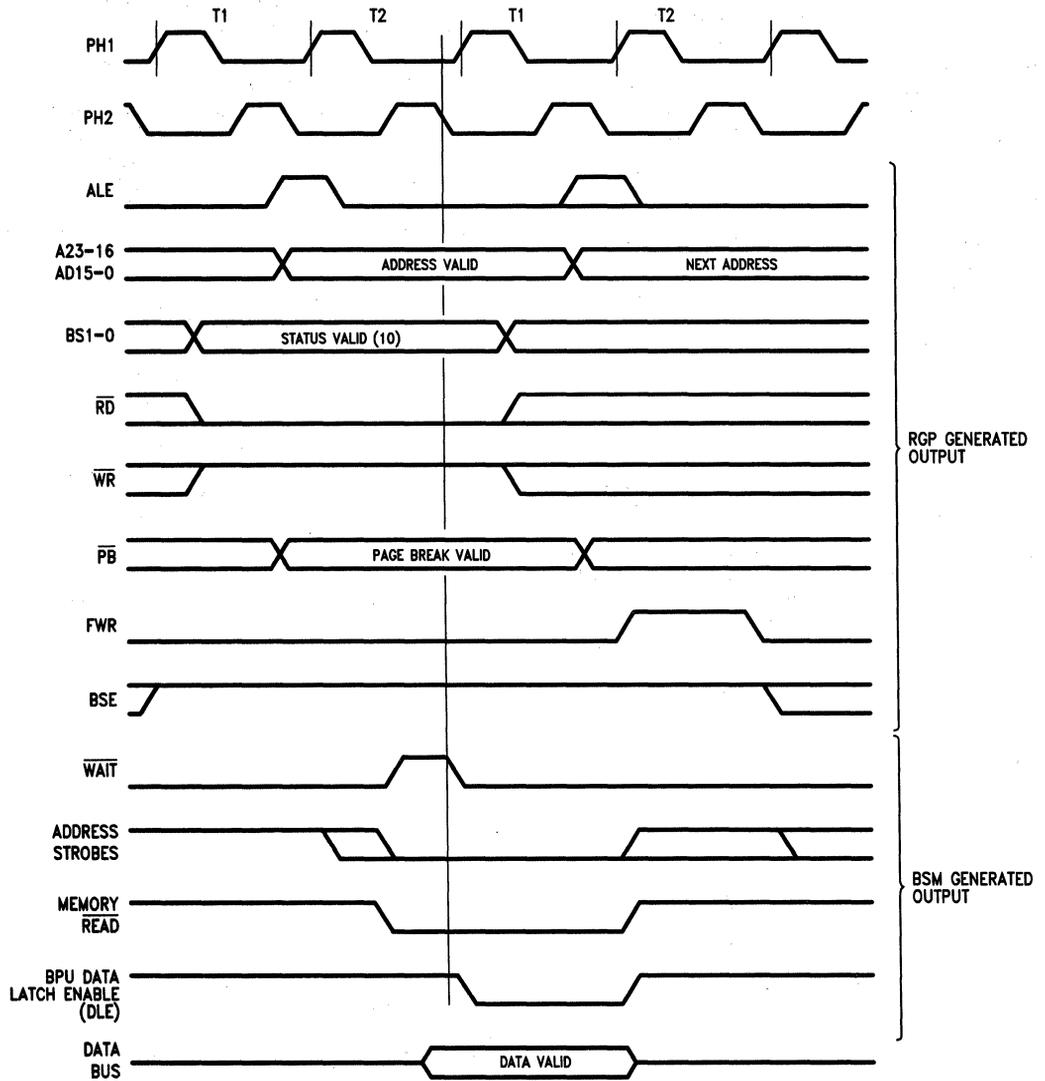


FIGURE 28. Functional BLIT Source Read Cycle

TL/F/9427-57

Architectural Description (Continued)

During BITBLT-Destination write operations the RGP generates the following bus status:

1. BS1 is a logic one,
2. BS0 is a logic zero,
3. L/\bar{B} is a logic zero,
4. \overline{RD} is a logic one,
5. \overline{WR} is a logic zero.

BITBLT Destination Write Bus cycles can be extended on an integral clock cycle basis by the assertion of WAIT. Extension is based on the premise that if WAIT is sampled low, (i.e., a logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state. Aside from the FRD strobe, T2(w) is an exact functional replica of the T2 state. Thus, T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one).

Figure 29 illustrates a flowchart of the BITBLT-Destination-Write transfer mechanism while Figure 30 illustrates functional bus information.

BITBLT DESTINATION READ-MODIFY-WRITE MECHANISM

At the system memory level, BITBLT Read-Modify-Write (RMW) operations are composed of a BITBLT-destination-read directly followed by a BITBLT-destination-write. A BITBLT RMW can best be described as a three stage process of which:

1. the read portion of the bus cycle reads destination data from memory and latches it in the BPU.
2. This data is logically combined in the BPU Logic Unit with the source contents read from the BPU's FIFO (the modify portion).

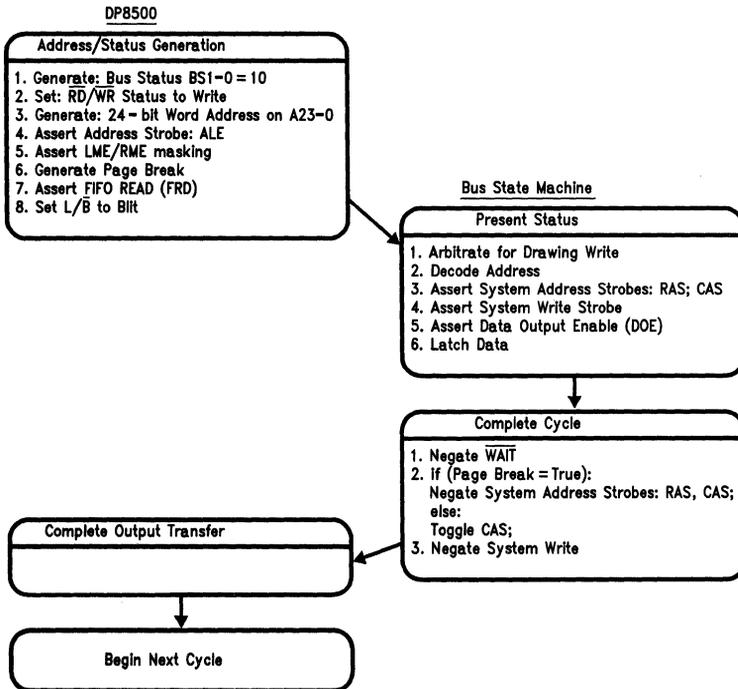


FIGURE 29. BLIT Destination Write Transfer Flow Chart

TL/F/9427-58

Architectural Description (Continued)

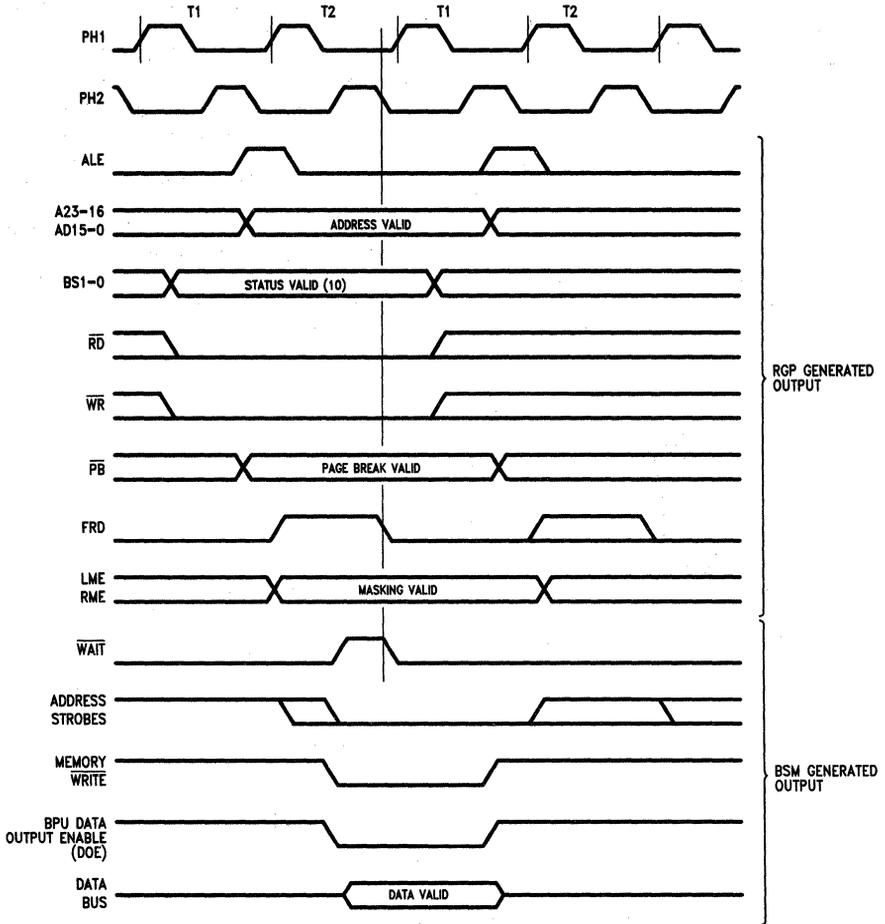


FIGURE 30. Functional BLIT Destination Write Cycle

TL/F/9427-59

Architectural Description (Continued)

3. A destination write cycle completes the bus operation by writing the resultant data back to the original destination memory location.

From the RGP perspective, a no-wait, or minimum BITBLT RMW cycle is composed of two unique "T" state clock cycles, of which:

1. state T1 produces ALE, address and bus status and FRD (see below),
2. state T2 and T2 wait (T2(w)) maintains all addresses, bus status and samples WAIT.

BITBLT RMW operations are composed of the following bus status:

1. BS1 is a logic one,
2. BS0 is a logic zero,
3. L/B is a logic zero,
4. LME is automatically generated (a logic one) for left edge of BITBLT,
5. RME is automatically generated (a logic one) for right edge of BITBLT,
6. \overline{RD} is a logic zero,
7. \overline{WR} is a logic zero.

A BITBLT RMW Bus cycle can be extended on an integral clock cycle basis by the assertion of WAIT. Extension is based on the premise that if WAIT is sampled low, (i.e., a

logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state. Aside from the FRD generated during T2, T2(w) is an exact functional replica of the T2 state. Thus, T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one).

Figure 31 illustrates a flowchart of the BITBLT RMW transfer mechanism while Figure 32 illustrates functional bus information.

LINE DRAWING

Conceptually, line drawing operations are the same as BITBLT RMW operations with the exception that only one bit of the destination word is altered during each RMW cycle. This is realized by using the BPU's Pixel Port contents as the "source" data rather than the data from the FIFO and selecting the bit position via the BPU's pixel address inputs. Line drawing can best be described as a three stage process of which:

1. destination data word is read from memory to the BPU,
2. one of sixteen bits is selected (as determined by B3-0) which is mixed with the contents of the Pixel port (the modify portion).
3. The write portion completes the bus operation by writing the accumulated results back to the destination memory location.

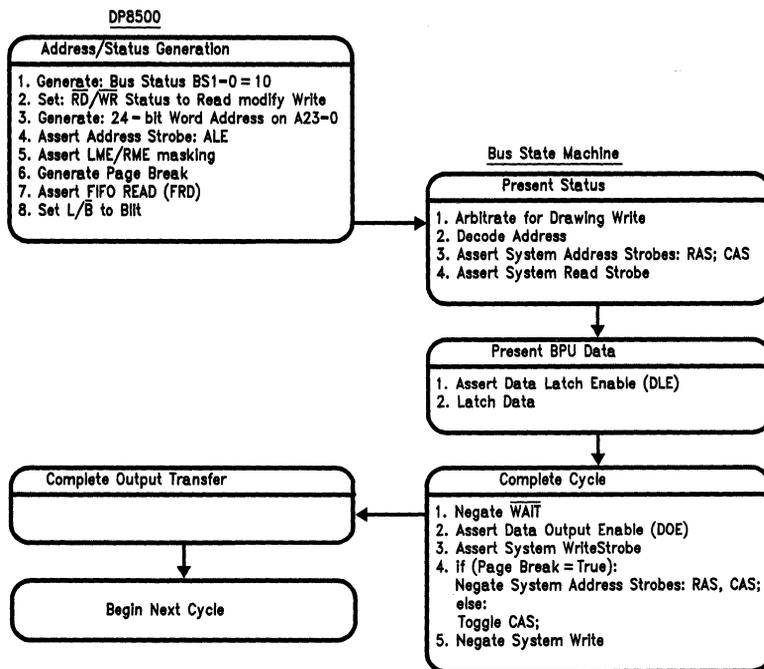


FIGURE 31. BLIT Read Modify Write Transfer Flow Chart

TL/F/9427-60

Architectural Description (Continued)

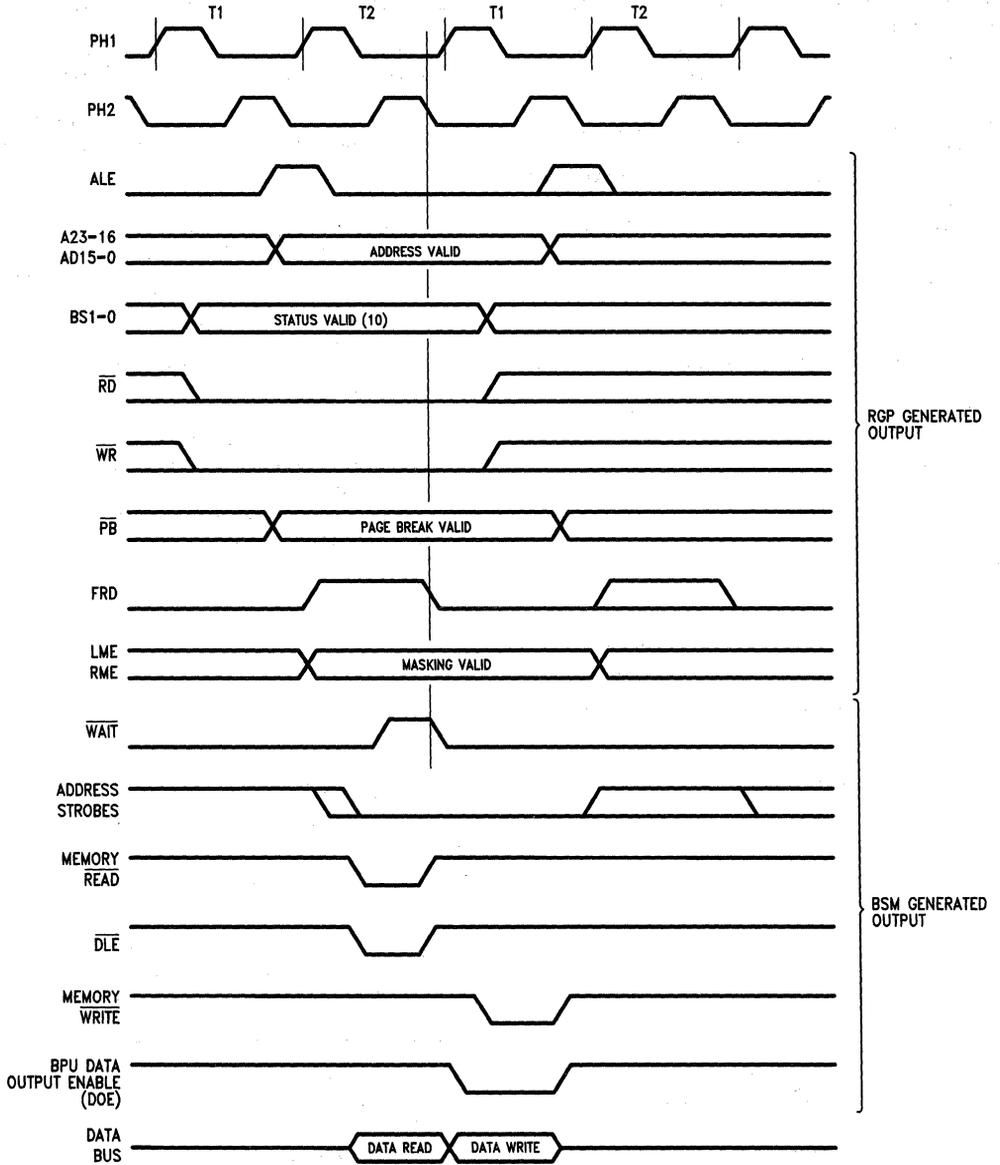


FIGURE 32. Functional BLIT Read-Modify-Write Cycle

TL/F/9427-61

Architectural Description (Continued)

From the RGP perspective, a no-wait, or minimum BITBLT Destination write operation is composed of two unique "T" state clock cycles, of which:

1. state T1 produces ALE, address, bus status,
2. state T2 and T2 wait (T2(w)) maintains all addresses, bus status, B3-0 and samples WAIT.

The RGP's Line operations generate the following bus status:

1. BS1 is logic one,
2. BS0 is a logic zero,
3. L/\bar{B} is a logic one,
4. B3-0 are generated to select one bit of the destination word,
5. \bar{RD} is a logic zero,
6. \bar{WR} is a logic zero.

Line drawing operations can be extended on an integral clock cycle basis by the assertion of WAIT. Extension is based on the premise that if WAIT is sampled low, (i.e., a logic zero at the PHI 2 falling edge sample window) during the T2 state, the RGP will produce a successive T2(w) state. T2(w) is an exact functional replica of the T2 state. Thus, T2(w) cycles will continuously be produced until WAIT is sampled high (i.e., a logic one).

Figure 33 illustrates a flowchart of the Line drawing mechanism while Figure 34 illustrates functional bus information. Not included as part of the diagram is the generation L/ \bar{B} . The reason for this is because L/ \bar{B} is normally set to line mode a number of clock cycles prior to executing the actual drawing function.

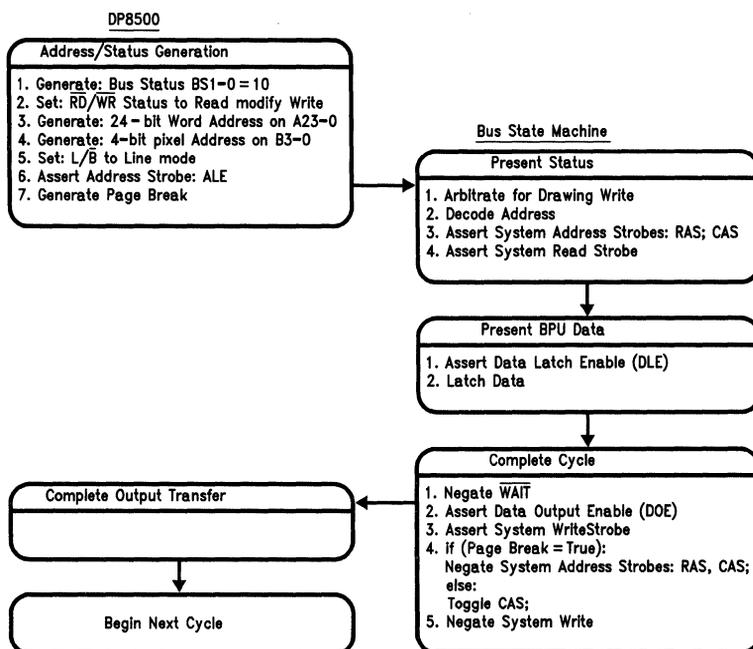


FIGURE 33. Line Drawing Transfer Flow Chart

TL/F/9427-82

Architectural Description (Continued)

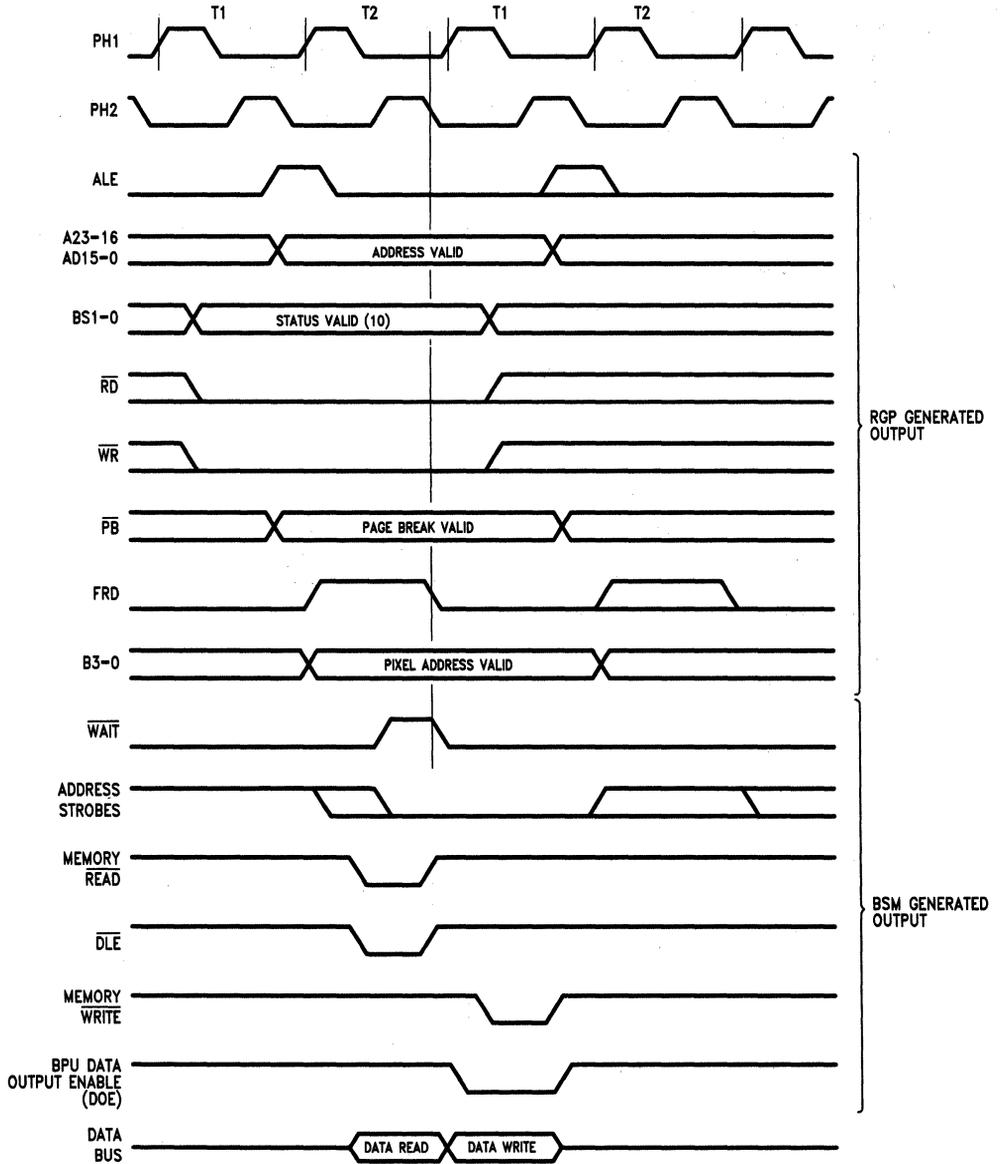


FIGURE 34. Functional Line Read-Modify-Write Cycle

TL/F/9427-83

Architectural Description (Continued)

SYSTEM PIPELINE CONSIDERATIONS

In many high speed system applications, pipelining serves a crucial role in increasing the overall setup time required to perform a given system operation. Synchronous pipelining works on the principle of deliberately delaying various processor outputs multiples of the system clock. In effect, the system views a delayed copy of the original Bus Master bus cycle. It is important to understand that a pipelined system doesn't give something for nothing, and for this reason it is important to ensure that the amount of time the system processor spends (based on clock cycles) on the bus is equivalent to the time the system spends executing the bus cycle. Systems based on having an external BSM lends well to effective pipelining.

When properly executed, pipelining gives the system extra time for decoding addresses, detecting possible changes in address, and allows much tighter control over critical output signals.

Aside from providing a tighter control of time critical signals, RGP drawing bus cycles can benefit significantly in a pipelined system. In particular, the inherent nature of the pipelined BITBLT Source Read bus cycle can be optimized in a pipelined system such that with a one phase clock delay, the BSM assertion of DLE can easily be met during the next bus cycle T1. Please refer to *Figure 28*.

The signals best suited for pipelining are:

1. BSE,
2. L/ \bar{B} ,
3. B0/LME,
4. B1/RME,
5. B2/FWR,
6. B3/FRD.

A one clock pipelined delay is sufficient for optimum operation.

INTERRUPT OPERATION

The RGP provides to the user two hardware interrupt inputs: a Non-Maskable Interrupt (NMI) and an Interrupt (INT). NMI is sampled on a negative edge transition, while the INT input functions as a level detect.

The nature of the INT input is such that it should be asserted as a system status, and maintained true until the RGP clears the Interrupt generating device. Because INT is sampled internally, proper setup and hold time relative to the falling edge of phase two guarantee that INT will be sampled in accordance with the timing specifications.

Assertion of NMI and INT can happen asynchronously to the RGP phase clock without causing any internal metastability problems. However, in the event that INT violates setup, recognition will not be guaranteed to happen until the next clock cycle.

Unlike the RGP HOLD and HOLDA mechanism where the acknowledgement latency is based on a "per-bus-cycle" basis, acknowledgement of either NMI or INT takes place after the current instruction is complete. Consequently, Interrupt latency will exist prior to the RGP servicing the Interrupt Handling routine.

HALT OPERATION

Included as an RGP output is HALT. Depending on the system implementation, HALT can serve as a hardware or software handshaking mechanism. As an example HALT could be used for an "end of display-list processing" indicator. HALT is often used as a convenient mechanism for implementing double-buffering schemes.

Whenever the HALT instruction is executed, the RGP will enter an internal HALT state, and the HALT output will be asserted. In this state, the RGP's bus will remain driven, yet quiescent. However, some bus activity, in the form of Video refresh cycles will continue to be produced if that function has been enabled. During the HALT state, a HOST can assert HOLD, with the RGP immediately relinquishing the bus by providing the handshake response HOLDA (see BUS ARBITRATION).

The RGP will continue to remain in the HALT state until either an NMI or INT is asserted. At that time the RGP will resume normal bus operation by branching to the appropriate interrupt service routine.

RESET OPERATION

Driven by a clean, external logical transition, the RGP Reset (RSTI) forces the RGP to respond in the following manner:

1. Forces the Program Counter to return to address zero.
2. The Video Attribute Register is cleared.
3. RSTO is asserted, and will continue asserted as long as RSTI is zero.
4. ALE goes to logic zero,
5. \bar{RD} and \bar{WR} goes to logic one,
6. and $BS1-0 = 00$.

When RSTI is asserted, it will not affect the contents of the Data, Address Registers or Program Flags, nor will it cause any type of internal metastability problems should it be asserted asynchronously. However, in asynchronous applications, it will require an additional internal clocked synchronization cycle in order to function in accordance with the timing specification.

During reset, it is recommended the BSM be forced to a known state. Eight clock cycles after the release of Reset, the first bus cycle will appear on the bus. *Figure 35* illustrates the sequence.

Architectural Description (Continued)

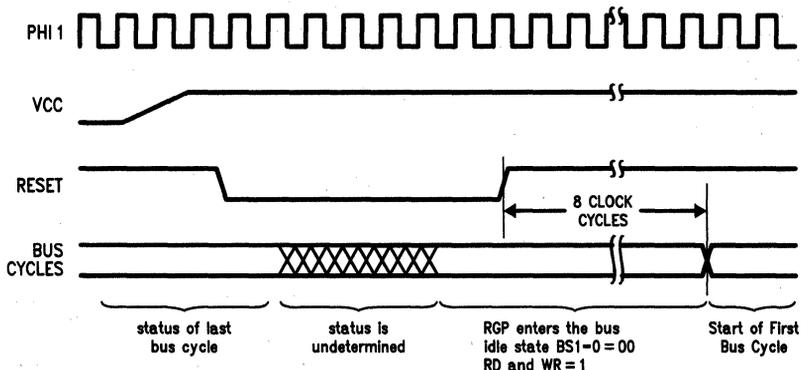


FIGURE 35. External Reset Operation Timing

TL/F/9427-64

BUS ARBITRATION MECHANISM

Bus arbitration allows a HOST to arbitrate for the system bus. The bus arbitration protocol takes on the following form:

1. An external device asserts a HOLD request to the RGP,
2. the RGP will assert HOLD acknowledge (HOLDA) after the current bus cycle, if no internal Video Refresh cycle is pending,
3. during HOLDA, the RGP will float the Address lines, but will continue to drive the remaining outputs in a quiescent state. The outputs will reflect the last T-state of the previous bus cycle.

Unlike NMI and INT where acknowledgement is processed at the completion of the instruction, the HOLD and HOLDA arbitration mechanism is processed on a "per-bus-cycle" basis. The HOLD input is sampled on every falling edge of phase two. In asynchronous applications, assertion of HOLD will cause no internal metastability problems and is sampled true whenever the setup time is met according to the timing specifications.

BUS ARBITRATION PRIORITY SCHEME

Figure 2 illustrates the internal architecture of the RGP. The internal bus arbitration scheme of the RGP is based on the following:

1. Video Refresh,
2. HOLD, HOLDA mechanism

3. Non-Maskable Interrupt (NMI),
4. Interrupt (INT),
5. Non-Drawing and Drawing Bus Cycles.

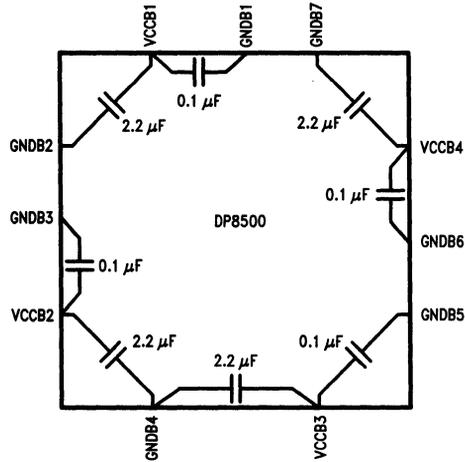
Generation of Video refresh must be the highest priority to ensure real-time operation. In situations where the RGP has relinquished the bus to the HOST, \overline{DRREQ} can possibly serve as a Preempting, or high priority system interrupt in order to indicate an impending Video refresh. Because the internal RGP bus arbitration mechanism allows Video Refresh cycles to have highest priority, it is important that a similar priority structure is reflected in the structure of the BSM.

RELATIONSHIP OF RGP PHASE AND LOAD CLOCKS

The RGP is designed to clock under direction of an external Two-Phase, and Load Clock (LCK) sources. Both non-overlapping Phase clocks are used to drive the main internal control logic block, while the LCLK input synchronizes the RGP to the external video data shift rate. Generation of \overline{DRREQ} , Horizontal, Vertical and Blanking sync signals are based on the LCLK source.

In order to maintain consistency in the assertion of \overline{DRREQ} , the RGP samples LCLK internally with the Phase input clocks. As a result, LCLK must always function at a sub-multiple of the phase clock frequency (see AC timing specifications), otherwise, generation of \overline{DRREQ} , and internal metastability could result from such practice.

Architectural Description (Continued)



Optimum DP8500 Buffer Bypassing

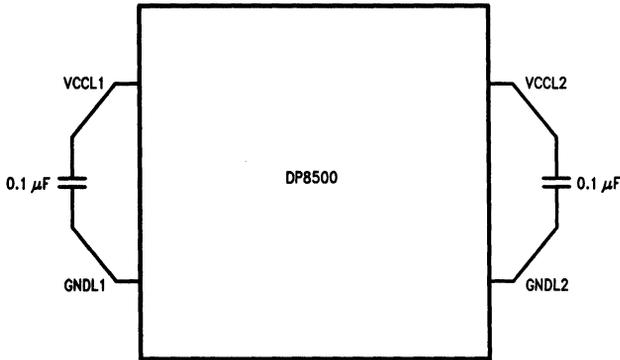
TL/F/9427-65

Note 1: Capacitor type: Ceramic

Note 2: Keep all capacitor lead length as short as possible; use leadless if possible.

Note 3: Keep logic supply separate from buffer supply.

Note 4: For optimum Buffer performance, the diagram illustrates a recommended layout for capacitor and capacitor values. Depending on the system parasitic capacitance, actual bypass capacitor values may vary.



Optimum DP8500 Logic Bypassing

TL/F/9427-66

Absolute Maximum Ratings*

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature Range -65°C to + 150°C
 Supply Voltage (V_{CC}) -0.5V to 7V
 Voltage at Any Pin
 with Respect to GND -0.5V to V_{CC} + 0.5V
 Package Power Dissipation @ 20 MHz 2.5W @ 25°C

Note: Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.

*Note: These are preliminary specifications.

DC Electrical Characteristics

Commercial: T_A = 0°C to +70°C, V_{CC} = 5V ±10%, GND = 0V
 Military: T_A = -55°C to +125°C; V_{CC} = 5V ±10%, GND = 0V

Symbol	Characteristics	Conditions	Commercial		Military†		Unit
			Min	Max	Min	Max	
V _{IH}			2.0		2.0		V
V _{IL}				0.8		0.8	V
V _{CH}	MOS Clock High	PH1, PH2 Pins Only, MOS	V _{CC} - 0.5		V _{CC} - 0.5		V
V _{CL}	MOS Clock Low	PH1, PH2 Pins Only, MOS		0.3		0.3	V
V _{CLT}	MOS Clock Ringing	PH1, PH2 Pins Only, MOS	-0.5	0.5	-0.5	0.5	V
V _{OH}		I _{OL} = -3 mA	2.4		2.4		V
V _{OL}		I _{OL} = 3 mA		0.5		0.5	V
I _{IN}	Input Leakage Current	V _{IN} = V _{IH} or V _{IL}		±10		±10	μA
I _{OZ}	TRI-STATE Leakage for A16-A23	V _O = V _{CC} or GND		±60		±75	μA
	TRI-STATE Leakage for All Other Outputs	V _O = V _{CC} or GND		±10		±10	μA
I _{CC1}	Quiescent Current	PH1, PH2 at 20 MHz		50		50	mA
I _{CC2}	Supply Current	PH1, PH2 at 100 kHz		10		16	mA
I _{CC3}	Supply Current	PH1, PH2 at 20 MHz		70		70	mA
C _{IN}	Input Capacitance	f _{in} at 1 MHz		10		10	pF

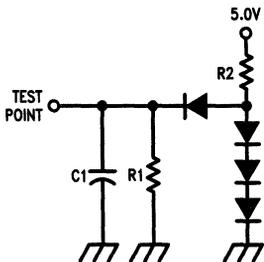
Thermal Characteristics

PGA Package

Symbol	Characteristics	Max	Units
θ _{JA}	Thermal Resistance—Ceramic Junction to Ambient	50*	°C/W
θ _{JC}	Junction to Case	3.5*	°C/W

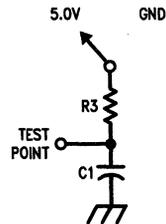
Note: *Estimated

† Preliminary



RGP Output Test Load Circuitry

TL/F/9427-46



TL/F/9427-47

RGP Output TRI-STATE Test Load Circuitry

Note 1: C1 = 50 pF

R1 = 6 kΩ

R2 = 1.3 kΩ

R3 = 1.8 kΩ

Note 2: Connect SW to +5V for t_{pLZ} and t_{pZL} measurements.

Note 3: Connect SW to GND for t_{pHZ} and t_{pZH} measurements.

DP8500 AC Timing CharacteristicsV_{CC} = 5.0 ± 10%; GND = 0V; Commercial T_A = 0°C to 70°C; Military T_A = -55°C to +125°C; Loading = 50 pF

Symbol	Figure	Description	Reference	Military		Commercial	
				Min	Max	Min	Max
tCp	23	PH1 or PH2 Clock Period	RE to Next RE	50		50	
tCh	23	PH1 or PH2 High Time	RE 50% to Next FE 50%	20		19	
tnOVL	23 23	PH1, PH2 Non-Overlap Time	PH1 (PH2) FE 50% to Next PH2 (PH1) RE 50%	3		3	
tLCp	23	LCK Period Mode 0 & 3	LCK RE to Next LCK RE Mode 1 & 2		3*tCp 2*tCp		3*tCp 2*tCp
tLCh	23	LCK High Time	LCK RE 50% to FE 50%	15		15	
tLCl	23	LCK Low Time	LCK FE 50% to RE 50%	15		15	
tLCKs	23	LCK High Setup Time	Before PH1 RE 50%	5		5	
tLCKh	23	LCK High Hold Time	After PH1 RE 50%	5		5	
tDis	24	Data in Setup Time	Before PH2 FE 50%	5		5	
tDih	24	Data in Hold Time	After PH2 FE 50%	10		10	
tDv	26	Data Valid Time	After PH2 RE 50%		43		38
tDiv	26	Data Invalid Time	After PH2 RE 50%	5		5	
tADf	20	AD15-0 Bus Floating	After PH2 RE 50%		42		35
tALv	20	Address 15-0 Valid	After PH2 RE 50%		43		38
tALf	24	Address 15-0 Float	After PH2 RE 50%		40		35
tALav	24	Address 15-0 Valid	Before ALE FE 50%	2		5	
tALiv	26	Address 15-0 Invalid	After PH2 RE 50%	5		5	
tALaiv	27	Address 15-0 Invalid	After ALE FE 50%	15		15	
tAHv	20	Address 23-16 Valid	After PH2 RE 50%		39		36
tAHiv	24	Address 23-16 Invalid	After PH2 RE 50%	5		5	
tAHf	20	Address 23-16 Floating	After PH2 RE 50%		40		35
tAHav	24	Address 16-23 Setup	Before ALE FE 50%	2		5	
tLMEv	28	LME Valid	After PH1 RE 50%		43		38
tRMEv	28	RME Valid	After PH1 RE 50%		43		38
tFWRv	27	FWR Valid	After PH1 RE 50%		43		38
tFRDv	28	FRD Valid	After PH1 RE 50%		43		38
tLMEiv	28	LME Invalid	After PH1 RE 50%	8		8	
tRMEiv	28	RME Invalid	After PH1 RE 50%	8		8	
tFWRiv	27	FWR Invalid	After PH1 RE 50%	8		8	
tFRDiv	28	FRD Invalid	After PH1 RE 50%	8		8	
tLBv		L/ \bar{B} Valid	After PH1 RE 50%		43		38
tBSEv	27	BSE Valid	After PH1 RE 50%		36		35
tLBiv		L/ \bar{B} Invalid	After PH1 RE 50%	8		8	
tBSEiv	27	BSE Invalid	After PH1 RE 50%	8		8	
tHALTv	34	\bar{H} alt Valid	After PH1 RE 50%		42		38

DP8500 AC Timing CharacteristicsV_{CC} = 5.0 ± 10%; GND = 0V; Commercial T_A = 0°C to 70°C; Military T_A = -55°C to +125°C; Loading = 50 pF (Continued)

Symbol	Figure	Description	Reference	Military		Commercial	
				Min	Max	Min	Max
tHALTiv	34	$\overline{\text{HALT}}$ Invalid	After PH1 RE 50%	8		8	
tHOLDs	20	$\overline{\text{HOLD}}$ Setup	Before PH2 FE 50%	5		5	
tHOLDiv	20	$\overline{\text{HOLD}}$ Invalid	After PH2 FE 50%	12		12	
tHLDAv	20	$\overline{\text{HLDA}}$ Valid	After PH2 RE 50%		40		36
tHLDAiv	20	$\overline{\text{HLDA}}$ Invalid	After PH2 RE 50%	6		6	
tRSTIs	37	$\overline{\text{RSTI}}$ Setup	Before PH2 FE 50%	15		15	
tRSTIh	37	$\overline{\text{RSTI}}$ Hold	After PH2 FE 50%	10		10	
tPWR	37	Min RESET Low Time	After Power On		12*tCp		12*tCp
tRSTOv	38	$\overline{\text{RSTO}}$ Valid	After PH1 RE 50%		40		35
tRSTOiv	38	$\overline{\text{RSTO}}$ Invalid	After PH1 RE 50%	9		10	
tALEv	20	ALE Valid	After PH2 RE 50%		30		26
tALEiv	20	ALE Invalid	After PH2 FE 50%	10		11	
tALEw	20	ALE Width	ALE RE 0.8V to 20V ALE FE 0.8V	20		19	
tWAITs	21	$\overline{\text{WAIT}}$ Setup	Before PH2 FE 50%	5		5	
tWAITh	21	$\overline{\text{WAIT}}$ Hold	After PH2 FE 50%	12		10	
tBSv	20	BS 1-0 Valid	After PH1 RE 50%		37		34
tRDv	20	$\overline{\text{RD}}$ Valid	After PH1 RE 50%		45		35
tWRv	20	$\overline{\text{WR}}$ Valid	After PH1 RE 50%		45		35
tBSiv	24	BS 1-0 Invalid	After PH1 RE 50%	12		12	
tRDiv	24	$\overline{\text{RD}}$ Invalid	After PH1 RE 50%	10		10	
tWRiv	24	$\overline{\text{WR}}$ Invalid	After PH1 RE 50%	10		10	
tBSAv	20	BS0-1 Valid	Before ALE RE 50%	3		5	
tRDAv	20	$\overline{\text{RD}}$ Valid	Before ALE RE 50%	2		5	
tWRAv	20	$\overline{\text{WR}}$ Valid	Before ALE RE 50%	3		5	
tBv	30	B3-0 Valid	After PH1 RE 50%		43		38
tBiv	30	B3-0 Hold	After PH1 RE 50%	9		8	
tPBv	20	$\overline{\text{PB}}$ Valid	After PH2 RE 50%		38		33
tPBiv	24	$\overline{\text{PB}}$ Invalid	After PH2 RE 50%	8		9	
tPBav	24	$\overline{\text{PB}}$ Setup	Before ALE FE 50%	2		5	
tHSYv	39	$\overline{\text{HSYNC}}$ Valid	After PH1 RE 50%		50		46
tHSYiv	39	$\overline{\text{HSYNC}}$ Invalid	After PH1 RE 50%	10		10	
tDRQv	31	$\overline{\text{DRREQ}}$ Valid	After PH1 RE 50%		39		35
tDRQiv	31	$\overline{\text{DRREQ}}$ Invalid	After PH1 RE 50%	9		10	

DP8500 AC Timing Characteristics

$V_{CC} = 5.0 \pm 10\%$; $GND = 0V$; Commercial $T_A = 0^\circ C$ to $70^\circ C$; Military $T_A = -55^\circ C$ to $+125^\circ C$; Loading = 50 pF (Continued)

Symbol	Figure	Description	Reference	Military		Commercial	
				Min	Max	Min	Max
tINTs	35	INT Setup	Before PH2 FE 50%	5		5	
tINTh	35	INT Hold	After PH2 FE 50%	10		10	
tNMIw	36	NMI Min. Width	FE to RE 50%	15		15	
tBLKv	40	\overline{BLANK} Valid	After PH1 RE 50%		39		38
tBLKiv	40	\overline{BLANK} Invalid	After PH1 RE 50%	8		10	
tVSYOv	41	\overline{VSYNC} Output Valid	After PH1 RE 50%		40		40
tVSYOiv	41	\overline{VSYNC} Output Invalid	After PH1 RE 50%	10		10	
tVSYIs	42	\overline{VSYNC} Input Setup	Before PH2 FE 50%	5		5	
tVSYIh	42	\overline{VSYNC} Input Hold	After PH2 FE 50%	24		22	

Notes: Military Specifications are preliminary. Please contact your National Semiconductor Sales Office or Distributor for availability and specifications.

The timing specifications listed above are based on a 50% threshold input voltage, with an output threshold of 1.5V.

Note 1: The column titled **Symbol** holds the symbolic name of the parameter, as it appears on the timing diagrams.

Note 2: The column titled **Figure** holds the figure number(s) of timing diagrams in which the parameter appears.

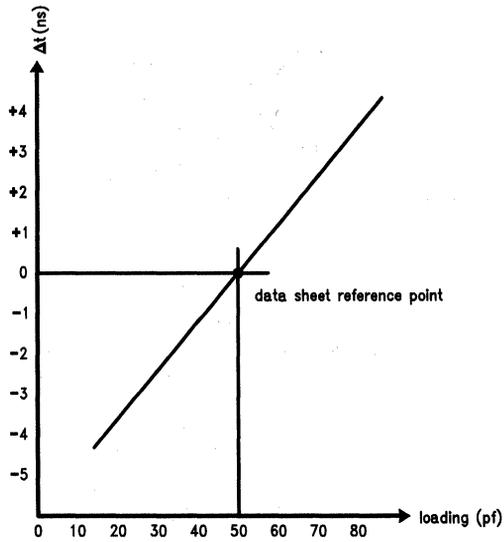
Note 3: All parameters (except tLCP and tPWR) are times in **nanoseconds**. tLCP and tPWR are based on tCP clocks. Usually a number appears either in the column titled **Max** or in the column titled **Min**, since in most cases, only one of these is of significance. In cases where both are important, both appear.

Note 4: The column titled **Reference** gives the name of a reference signal from which the given parameter is measured. The designation RE or FE following a signal name indicates the rising edge or falling edge of that signal. A percentage following RE or FE indicates the percent of the total rise or fall of the signal at the point from which the parameter being described is measured.

Note 5: Minimum tRSTW is one tCp.

Typical Performance Characteristics

DP8500 Output Capacitance Derating Curve



TL/F/9427-33

Note 1: All derated output values are based on the 50 pF data sheet reference point.

Note 2: The graph may not be linear outside range shown.

Note 3: Derated values apply to both the rising and falling edge.

Note 4: The curve reflects worst case @ 70°C.

Note 5: This curve applies ONLY to output drives.

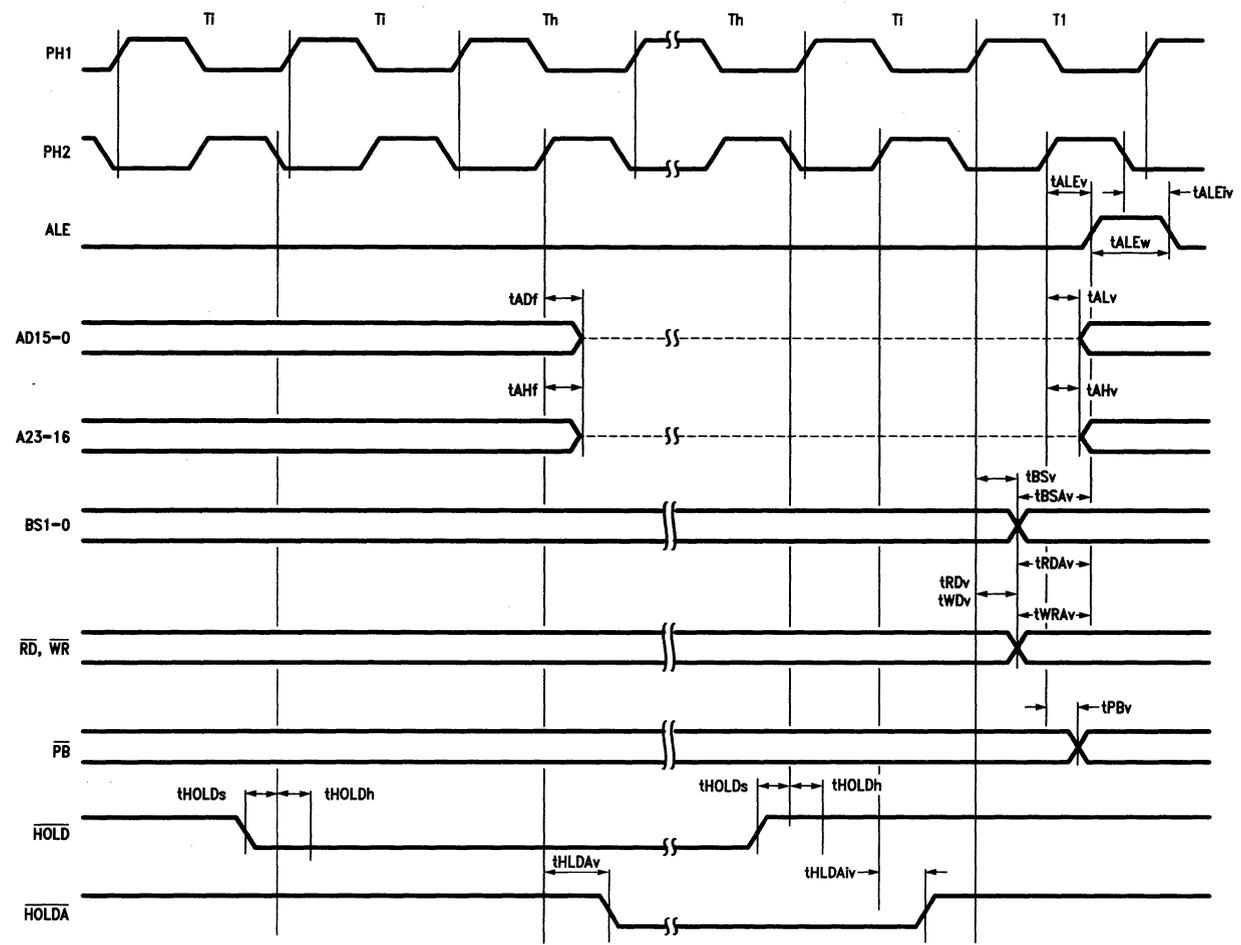
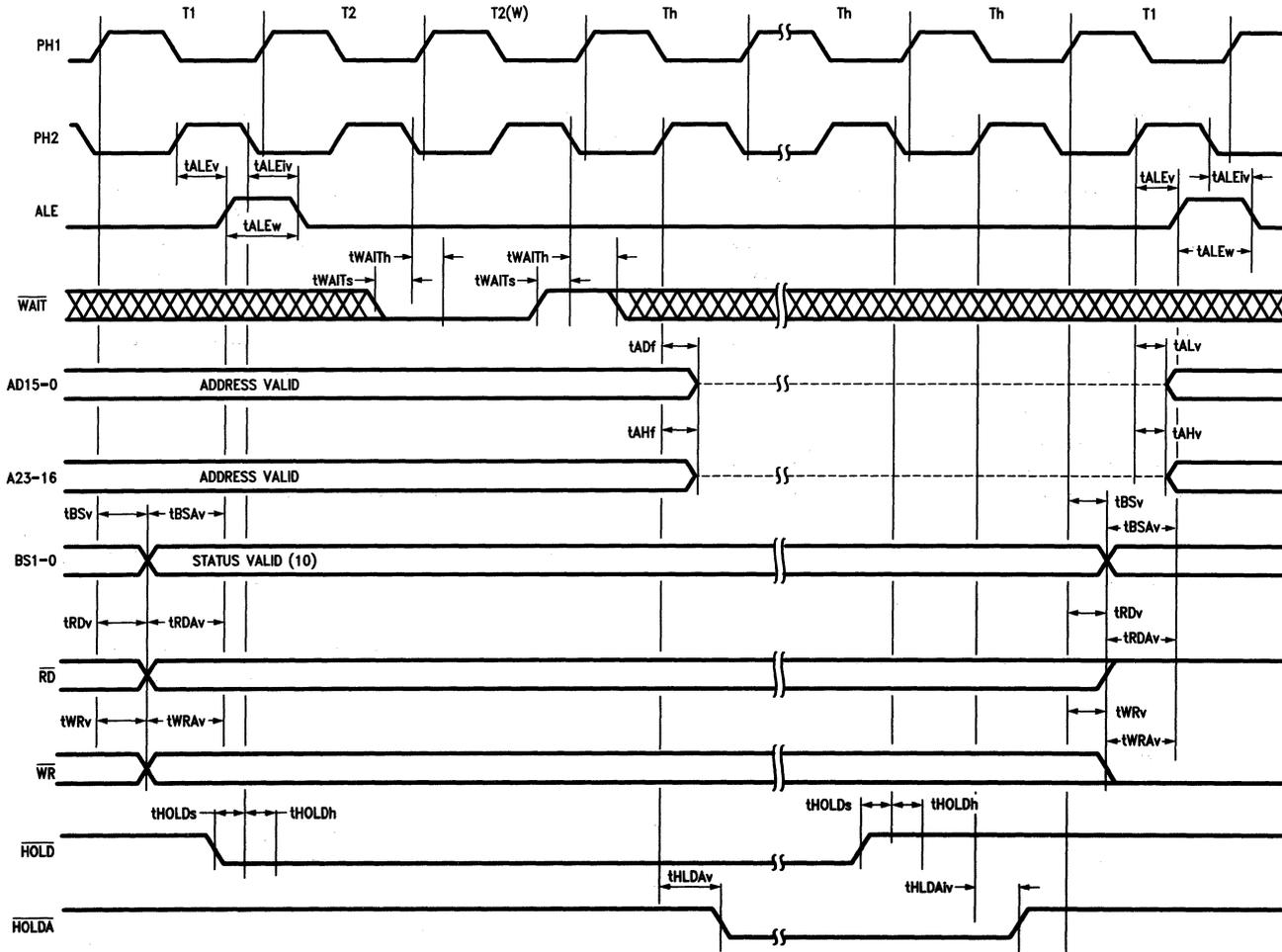


FIGURE 36. HOLD Timing (Bus Initially Idle)

TL/F/9427-21



1-50

FIGURE 37. HOLD Timing (Bus Busy with Drawing Cycle)

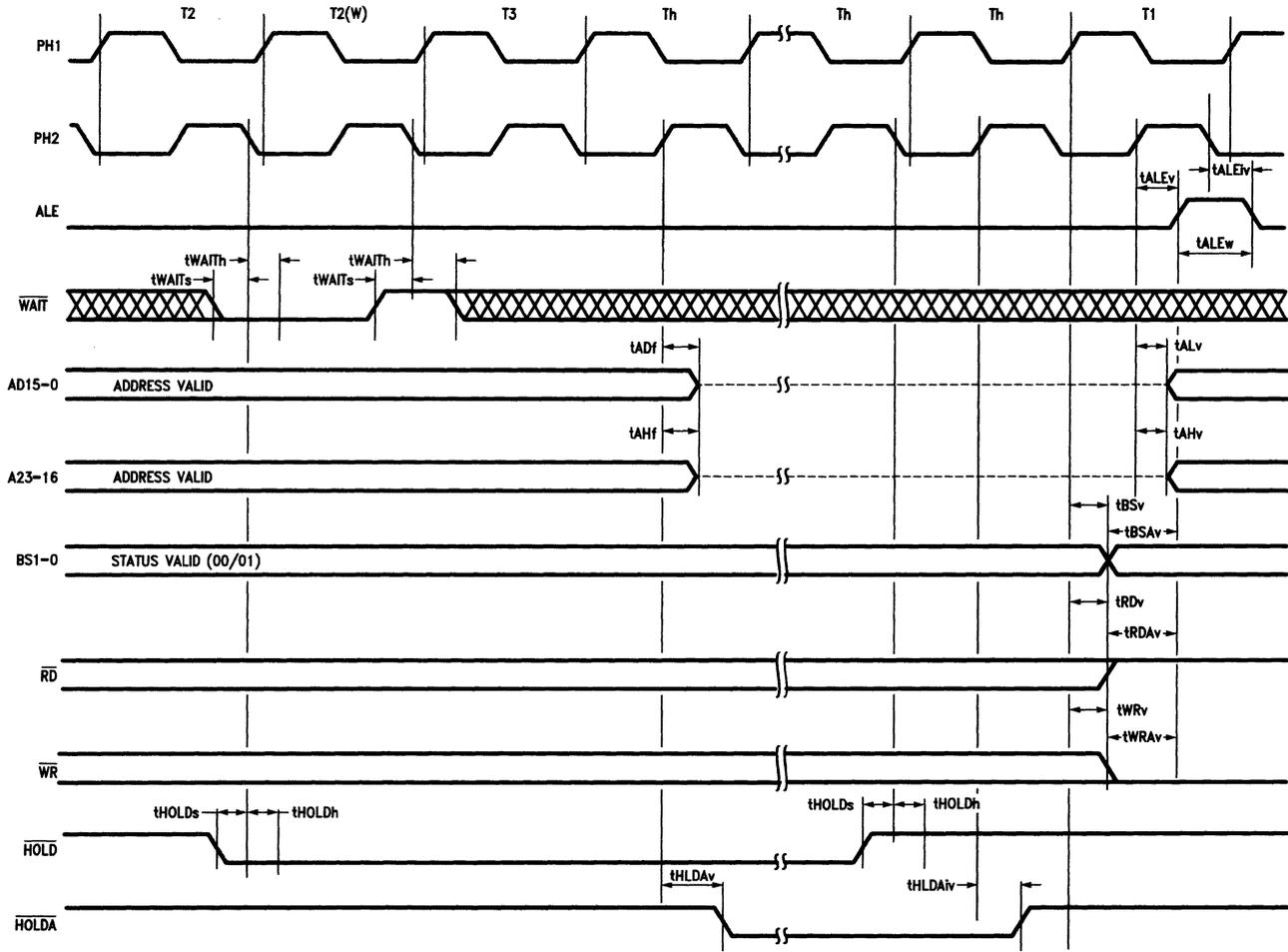


FIGURE 38. Hold Timing (Bus Busy with Non-Drawing Cycle)

TL/F/9427-23

Timing Waveforms (Continued)

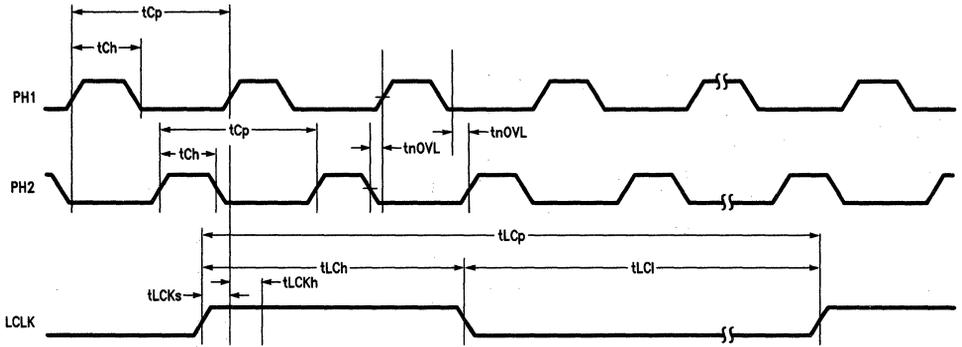


FIGURE 39. RGP LCLK Timing

TL/F/9427-24

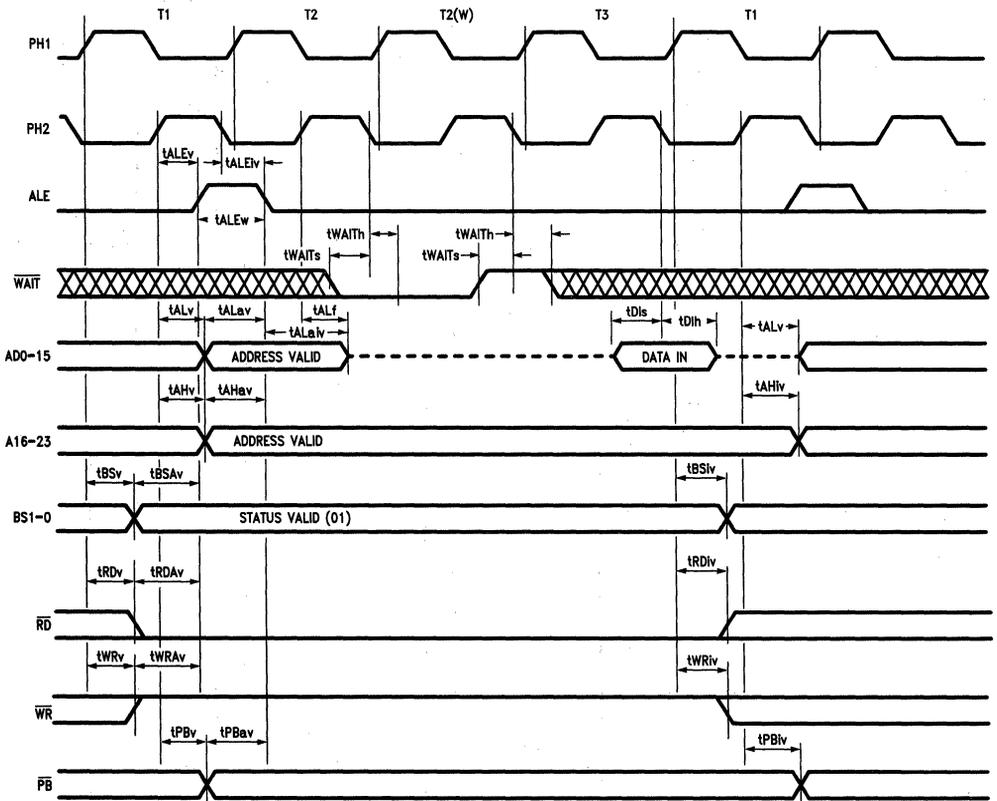


FIGURE 40. Instruction Read Cycle Timing

TL/F/9427-25

Timing Waveforms (Continued)

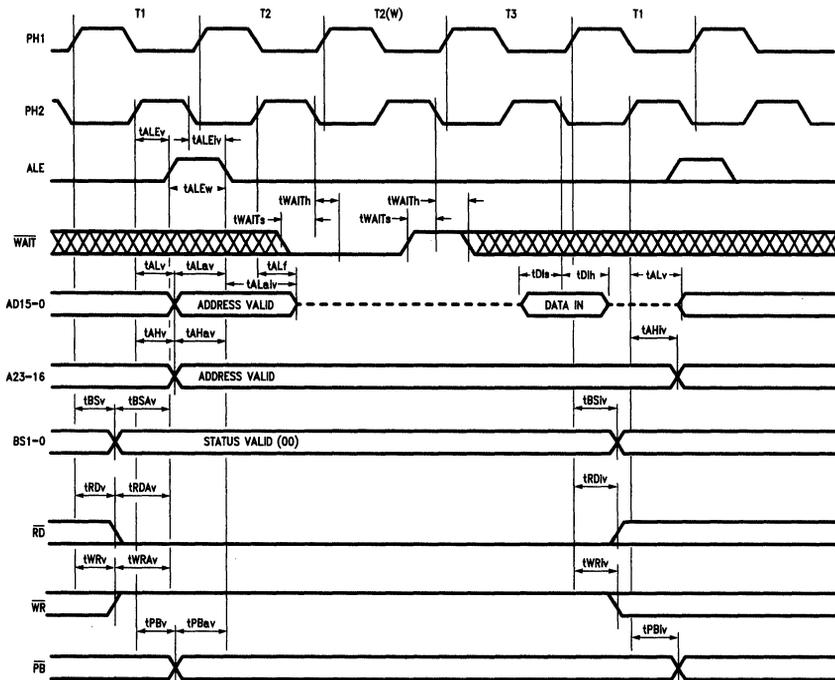


FIGURE 41. Operand Read Cycle Timing

TL/F/9427-26

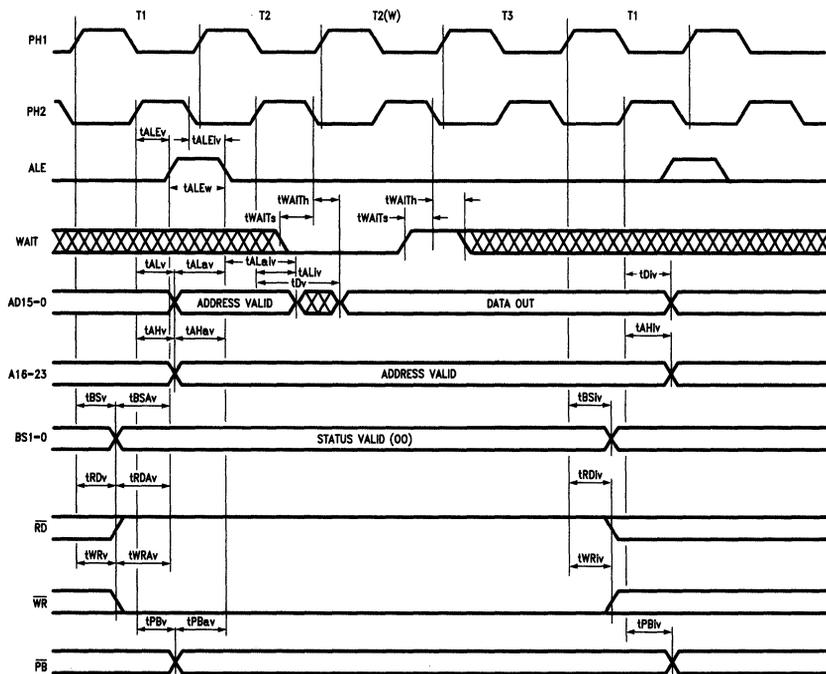


FIGURE 42. Operand Write Cycle Timing

TL/F/9427-27

Timing Waveforms (Continued)

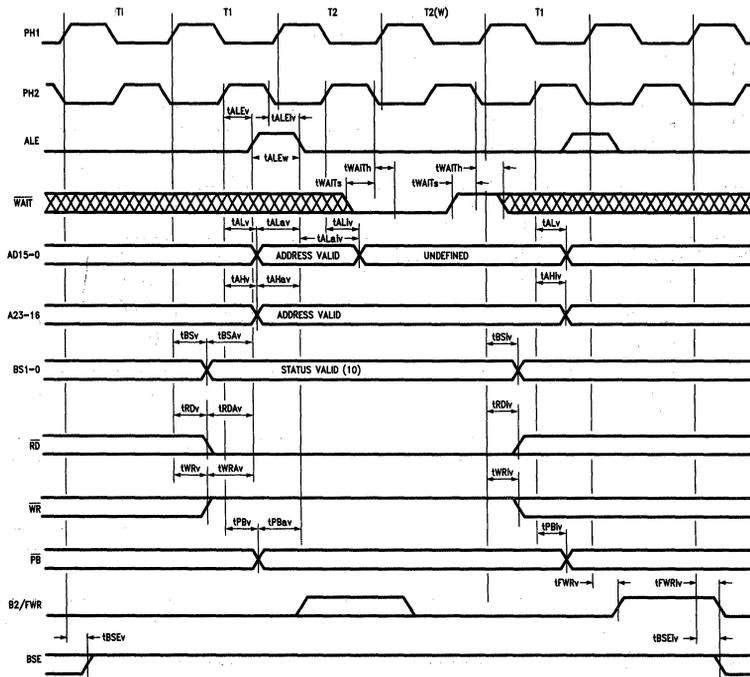


FIGURE 43. BITBLT Source Read Cycle Timing

TL/F/9427-28

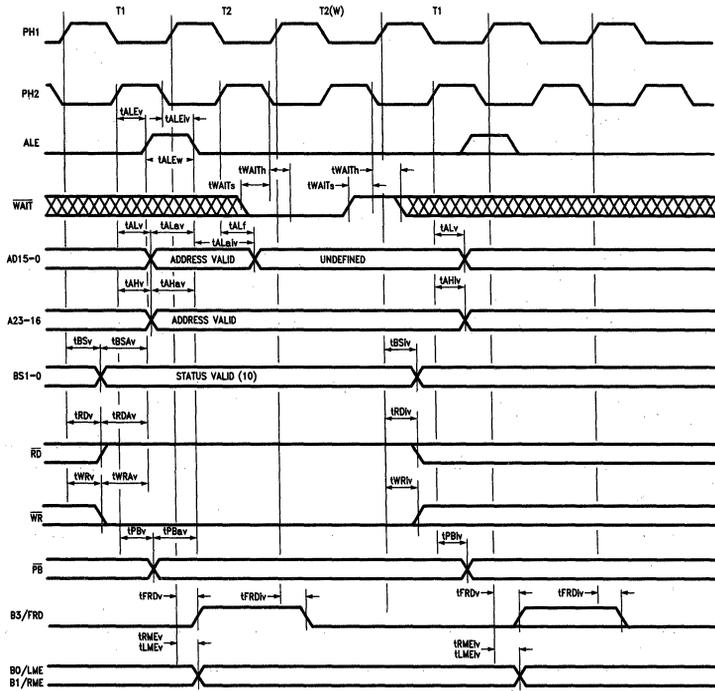


FIGURE 44. BITBLT Destination Write Cycle Timing

TL/F/9427-29

Timing Waveforms (Continued)

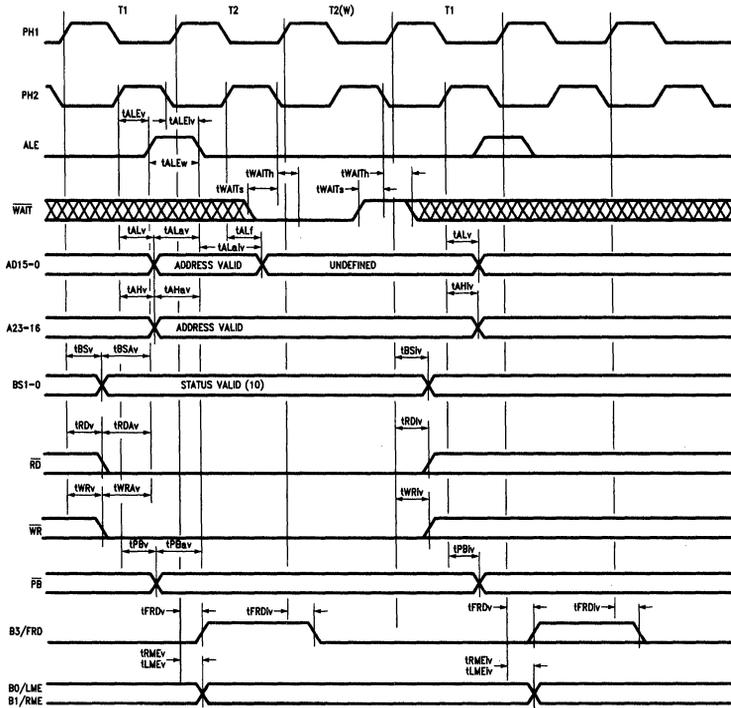


FIGURE 45. BITBLT Destination Read-Modify-Write Cycle Timing

TL/F/9427-30

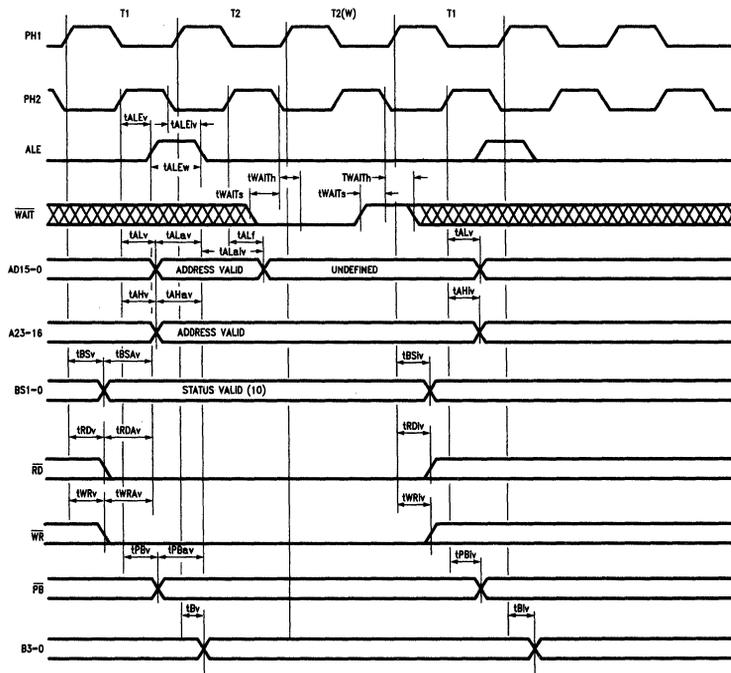


FIGURE 46. Line Drawing Read-Modify-Write Cycle

TL/F/9427-31

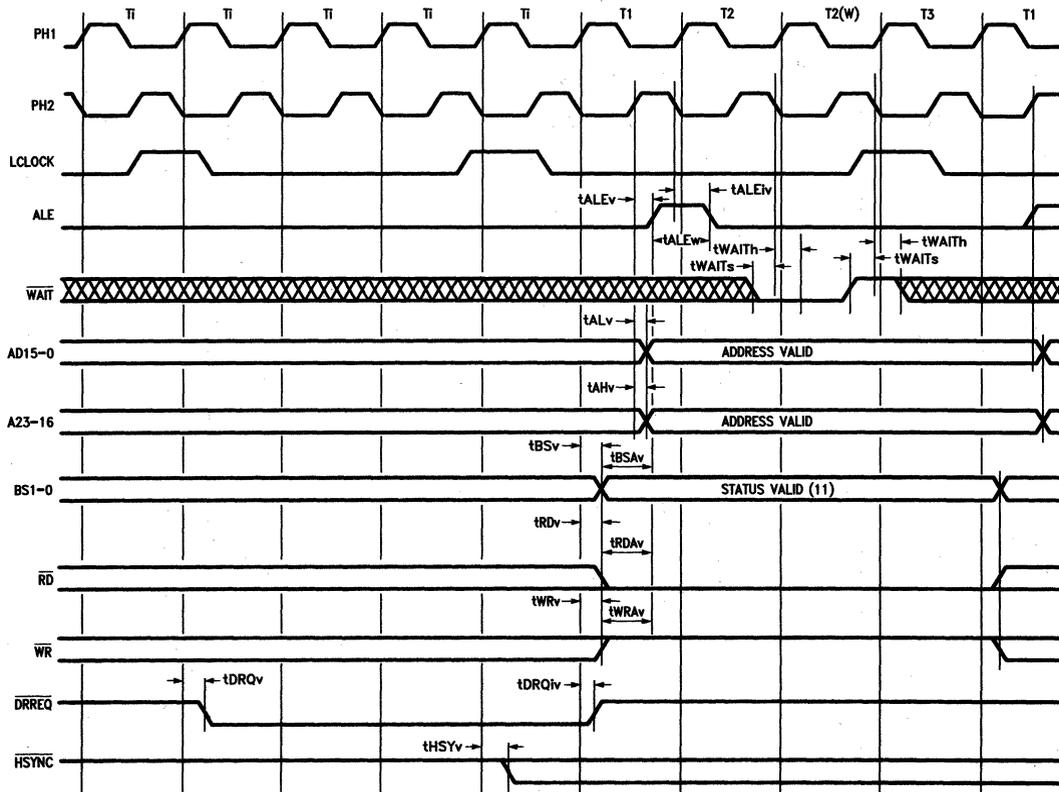


FIGURE 47. Mode 3 Video Refresh Cycle Timing (Bus Idle)

TL/F/9427-32

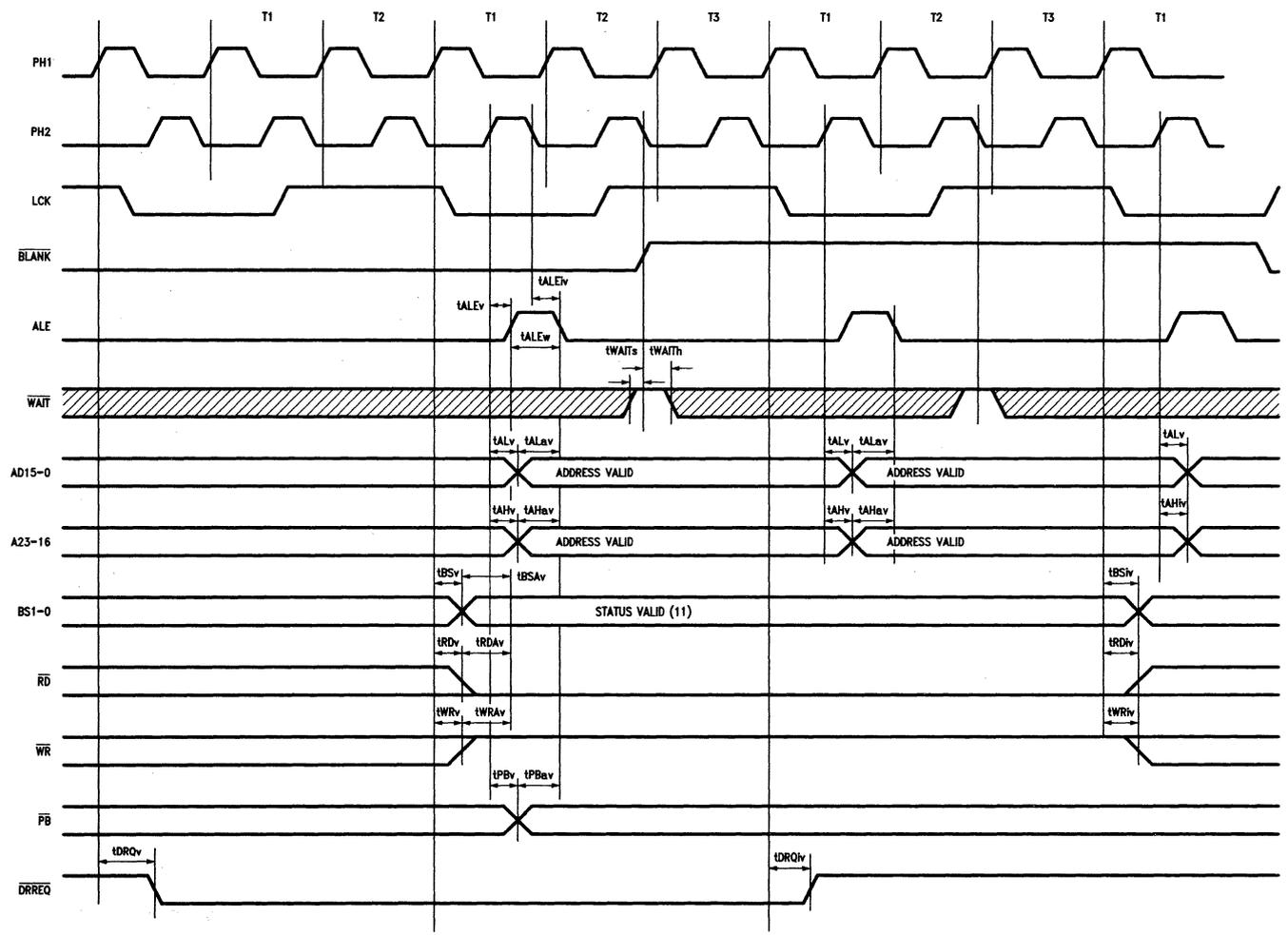


FIGURE 48. Mode 1 Display Refresh

TL/F/9427-35

1-57

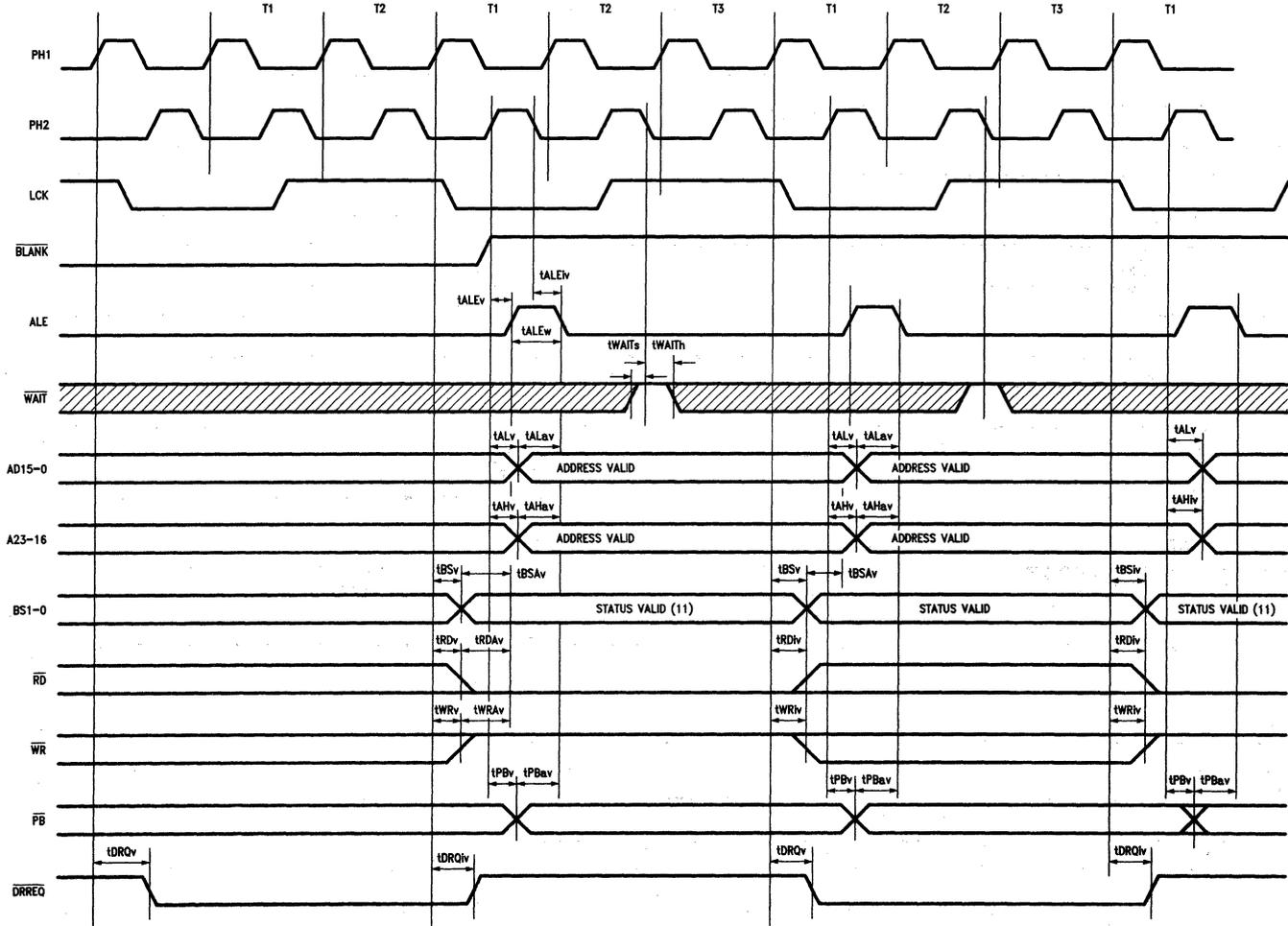


FIGURE 49. Mode 2 Display Refresh

Timing Waveforms (Continued)

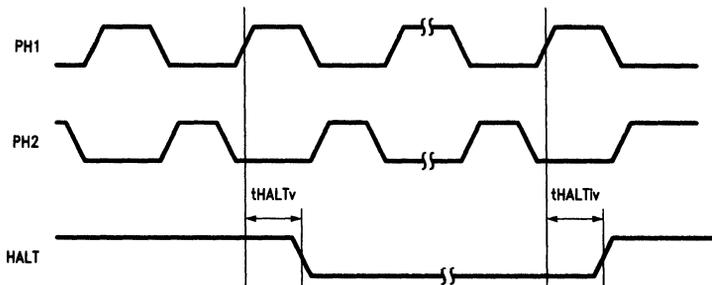


FIGURE 50. Halt Timing

TL/F/9427-37

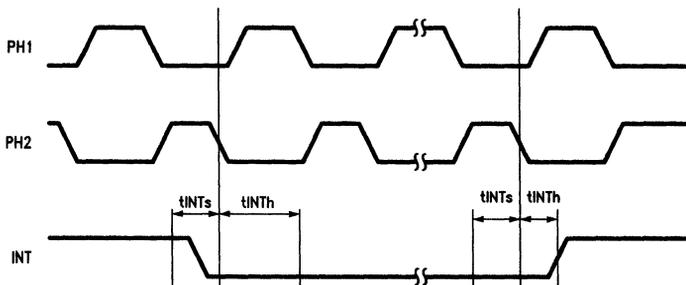


FIGURE 51. Interrupt Timing

TL/F/9427-38

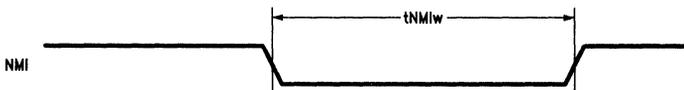


FIGURE 52. Non-Maskable Input Timing

TL/F/9427-39

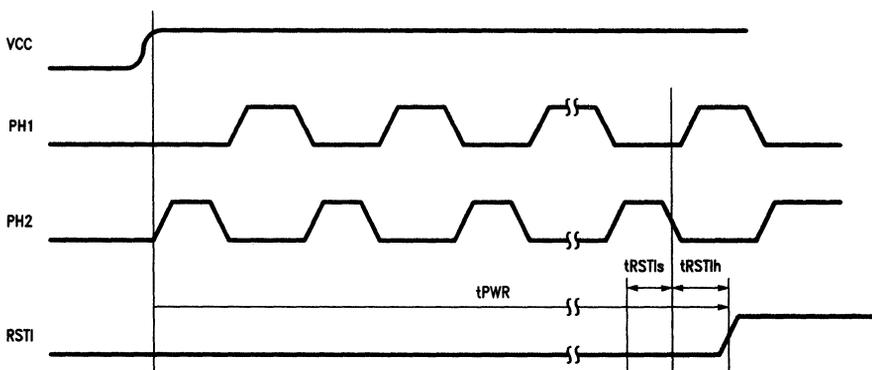


FIGURE 53. RESET Input Timing

TL/F/9427-40

Timing Waveforms (Continued)

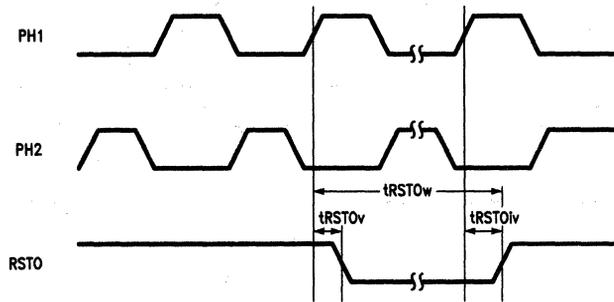


FIGURE 54. RESET Output Timing

TL/F/9427-41

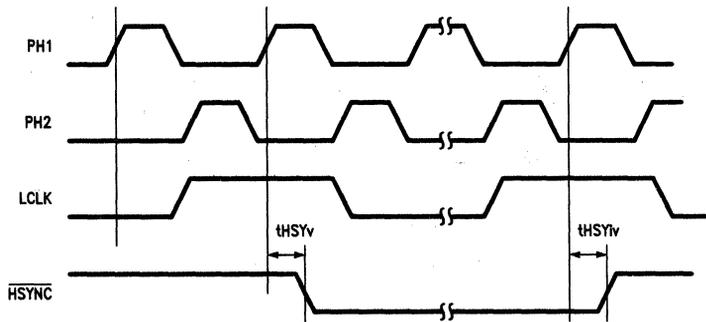


FIGURE 55. HSYNC Timing

TL/F/9427-42

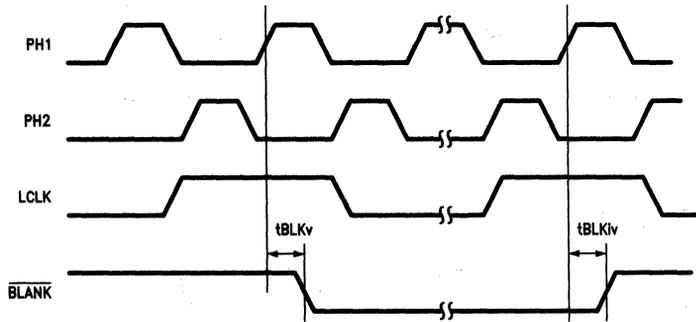


FIGURE 56. BLANK Timing

TL/F/9427-43

Timing Waveforms (Continued)

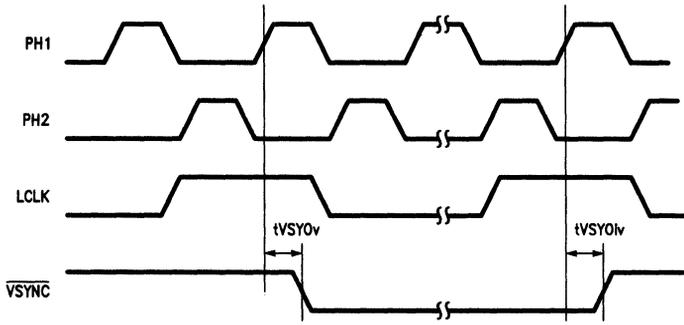


FIGURE 57. \overline{VSYNC} Output Timing

TL/F/9427-44

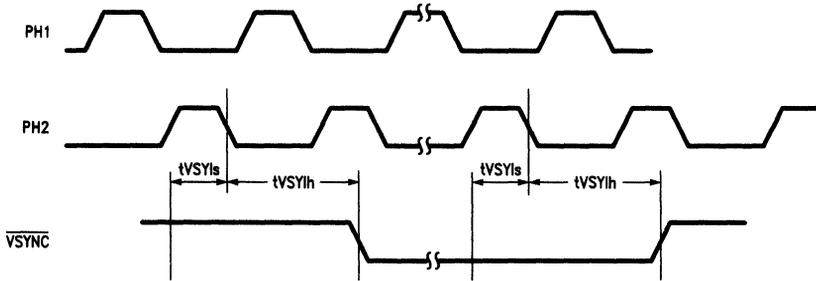


FIGURE 58. \overline{VSYNC} Input Timing

TL/F/9427-45

NS32CG16-10/NS32CG16-15 High-Performance Printer/Display Processor

General Description

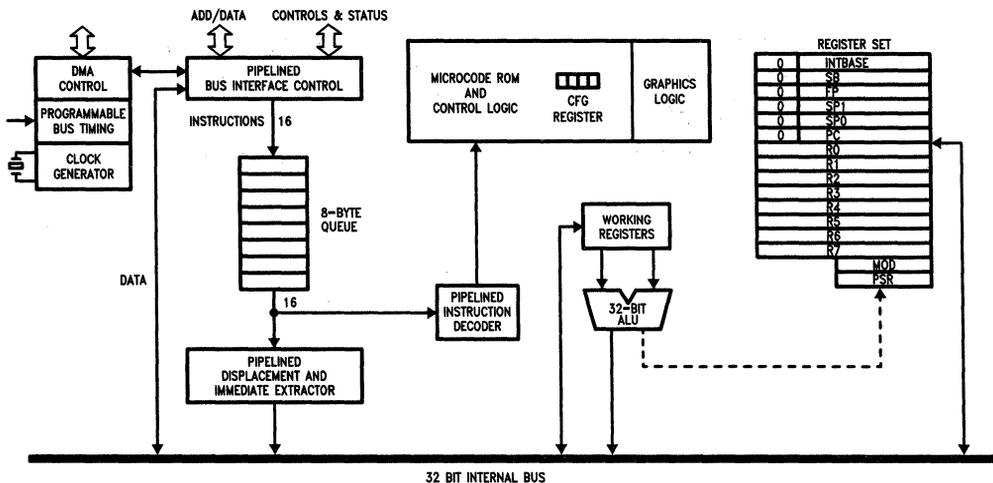
The NS32CG16 is a 32-bit microprocessor in the Series 32000® family that provides special features for graphics applications. It is specifically designed to support page oriented printing technologies such as Laser, LCS, LED, Ion-Deposition and InkJet.

The NS32CG16 provides a 16 Mbyte linear address space and a 16-bit external data bus. It also has a 32-bit ALU, an eight-byte prefetch queue, and a slave processor interface.

The capabilities of the NS32CG16 can be expanded by using an external floating point unit which interfaces to the NS32CG16 as a slave processor. This combination provides optimal support for outline character fonts.

The NS32CG16's highly efficient architecture, in addition to the built-in capabilities for supporting BITBLT (BIT-aligned BLock Transfer) operations and other special graphics functions, make the device the ideal choice to handle a variety of page description languages such as Postscript™ and PCL™.

Block Diagram



TL/EE/9424-1

Features

- Software compatible with the Series 32000 family
- 32-bit architecture and implementation
- 16 Mbyte linear address space
- Special support for imaging applications such as printers, faxes and scanners
 - 18 graphics instructions
 - Binary compression/expansion capability for font storage using RLL encoding
 - Pattern magnification for Epson and HP LaserJet™ emulations
 - 6 BITBLT instructions on chip
 - Interface to an external BITBLT processing unit for very fast BITBLT operations (optional)
- Floating point support via the NS32081 or the NS32381 for outline fonts, scaling and rotation
- On-chip clock generator
- Optimal interface to large memory arrays via the DP84xx family of DRAM controllers
- Power save mode
- High-speed CMOS technology
- 68-pin plastic PCC package



DP8510 BITBLT Processing Unit

General Description

The DP8510 BITBLT Processing Unit (BPU) is a high-performance microCMOS device designed for use in raster graphics applications. It implements, in high-speed pipelined logic, the data operations which are fundamental to BITBLT (BIT boundary Block Transfer) graphics: shifting, masking and bitwise logic operations. Under control of external hardware such as a state machine or a general-purpose microprocessor, it provides all necessary data path operations, easing the implementation of a wide variety of BITBLT systems. A number of input pins control the proper data flow in the BPU. A simple handshake scheme is used to interface the CPU, the BPU and the memory system.

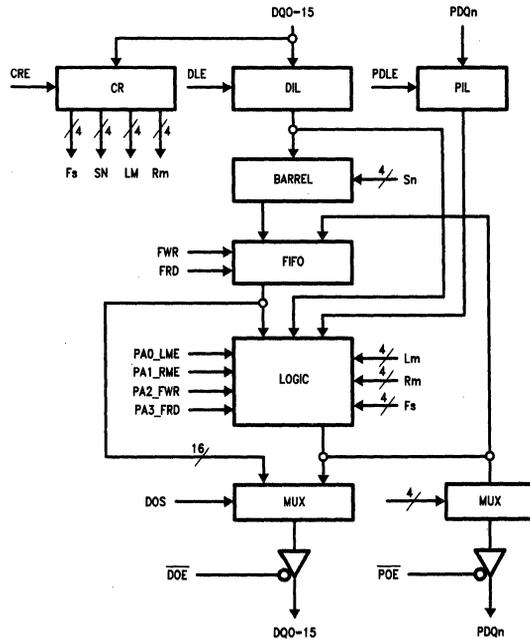
The BPU has two modes, BITBLT and line drawing. The mode is set by the \bar{B}/L pin. The line-drawing mode can be treated as a special case BITBLT with height and width equal to one.

In order to perform a BITBLT operation, the BPU's control register must first be loaded with four parameters: the shift number, left and right masks and the function select code, a total of 16 bits. BITBLT can then proceed, as directed by an external processor or state machine. It is the responsibility of the controller to generate appropriate addresses for the BITBLT, to interface with the frame buffer's memory control circuitry, and to control the BPU itself.

Features

- Supports all 16 classical BITBLT functions
- Pipelined data input for high system throughput
- Flexible architecture allows BPU to be used with a state machine or processor
- Multiple BPUs can be used for multiple bitplane/color applications
- Line drawing support
- Compatible with static or dynamic RAMs, including Video DRAMs
- Compatible with page mode, nibble mode and static column RAMs
- 32-bit to 16-bit barrel shifter
- 16-bit data port
- 16-word FIFO
- 16-bit logic operations
- 20 MHz operation
- Single +5 volt supply
- All inputs and outputs TTL-compatible
- Packaged in a 44-pin PCC (commercial) or 44-pin PGA (MIL)
- Single-bit pixel I/O port
- A member of National's Advanced Graphics Chip Set
- microCMOS technology

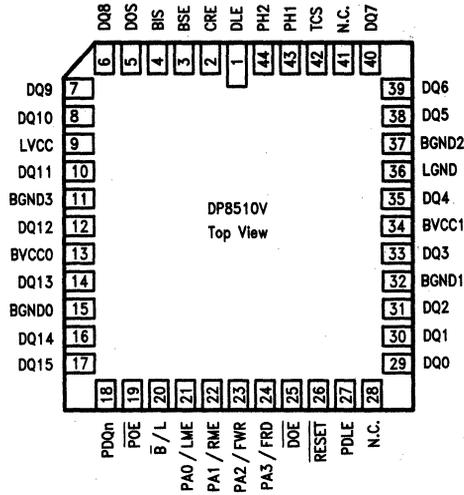
Block Diagram



TL/F/8672-22

Connection Diagrams

44-Pin Plastic Chip Carrier (PCC) Package



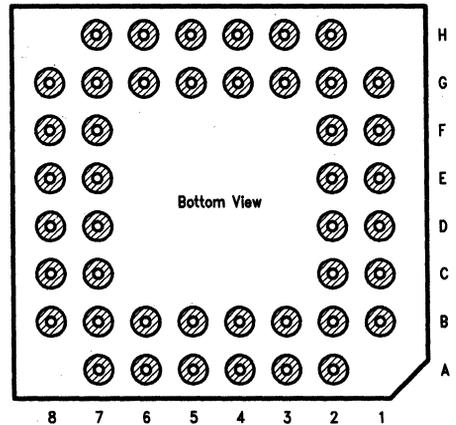
N.C. = No Connection

TL/F/8672-2

Order Number DP8510V
See NS Package Number V44A

44-Pin Grid Array (PGA) Package

Description	Pin	Description	Pin
DLE	A4	PA2/FWR	H5
CRE	A5	PA3/FRD	H4
BSE	B5	DOE	G4
BIS	A6	RESET	H3
DOS	B6	PDLE	G3
DQ8	A7	N.C.	H2
DQ9	B7	DQ0	G2
DQ10	C7	DQ1	F2
LVCC	B8	DQ2	G1
DQ11	C8	BGND1	F1
BGND3	D7	DQ3	E2
DQ12	D8	BVCC1	E1
BVCC0	E8	DQ4	D1
DQ13	E7	LGND	D2
BGND0	F8	BGND2	C1
DQ14	F7	DQ5	C2
DQ15	G8	DQ6	B1
PDQn	G7	DQ7	B2
POE	G6	N.C.	B3
B/L	H7	TCS	A2
PA0/LME	H6	PH1	A3
PA1/RME	G5	PH2	B4



TL/F/8672-25

Order Number DP8510U
See NS Package Number U44A

Pin Definitions

- DQ0-DQ15:** Data I/O Port, 16 bits wide. This is the main data port which is connected to the frame buffer. It serves as the 16-bit input to the Data Input Latch for both source and destination data. When \overline{DOE} is active (low), this port serves as data output.
- \overline{DOE} :** Data Output Buffer Enable. A low signal on this pin enables the data output buffers of DQ0-DQ15.
- PDQn:** Pixel Data I/O Port, 1 bit wide. This I/O port is used in the line drawing mode only. It serves as the single bit input to the Pixel Input Latch to provide a source bit for line drawing. When \overline{POE} is active (low), this port serves as an output for the pixel selected by PA0-PA3.
- \overline{POE} :** Pixel Output Buffer Enable. A low signal on this pin enables the pixel output buffer. This allows the CPU to read back the pixel data/value from the frame buffer in a multiple bit-plane system.
- \overline{B}/L :** BITBLT or Line Drawing. A low on this pin enables the BPU for BITBLT operation. A high on this pin sets the BPU to the line drawing mode.
- PA0/LME:** Pixel Address 0 or Left Mask Enable. When the BPU is in the line drawing mode, this pin inputs the least significant pixel address. When the BPU is in the BITBLT mode, this pin receives the Left Mask Enable, an active high signal. This input must be synchronized with respect to the falling edge of PH2.
- PA1/RME:** Pixel Address 1 or Right Mask Enable. When the BPU is in the line drawing mode, this pin inputs the second least significant pixel address. When the BPU is in the BITBLT mode, this pin receives the Right Mask Enable, an active high signal. This input must be synchronized with respect to the falling edge of PH2.
- PA2/FWR:** Pixel Address 2 or FIFO Write control. When the BPU is in the line drawing mode, this pin inputs the third least significant pixel address. When the BPU is in the BITBLT mode, this pin is the active high FIFO write control input. This input must be synchronized with respect to the falling edge of PH2.
- PA3/FRD:** Pixel Address 3 or FIFO Read control. When the BPU is in the line drawing mode, this pin inputs the most significant pixel address. When the BPU is in the BITBLT mode, this pin is the active high FIFO read control input. This input must be synchronized with respect to the falling edge of PH2.
- BIS:** Barrel Input Select. This signal controls the multiplexer prior to the BPU's barrel shifter. If this signal is high, a wordwise swap is performed between the two 16-bit inputs to the barrel shifter. If this signal is low, no swap is performed. Therefore, if this signal is low, the BIL register serves as the most-significant word to the barrel shifter, with the DIL-Source register serving as the least significant word. Conversely, when this signal is high, DIL-Source serves as the most significant input word to the barrel shifter, with BIL being the least significant word.
- BSE:** BITBLT Source Enable, enables the BITBLT source input data path and controls the latching function of the BITBLT source pipeline register. BSE should be held low (disabled) during the BITBLT destination data read/write cycles. This input must be synchronized with respect to the falling edge of PH2.
- TCS:** TTL Clock Select. This pin should tie to either V_{CC} or Ground. A high level on this pin selects the TTL level clock input. The use of a conventional TTL clock, permitted at clock frequencies up to 10 MHz, simplifies system design. When using TTL clock all references to the falling edge of PH2 must be changed to the rising edge of PH1. A low level selects PH1 and PH2 (MOS level clocks) as the BPU clock inputs.
- PH1:** Phase 1 clock input or the TTL clock input. When TCS is set low, this is the PH1 clock input, MOS level, maximum clock rate 20 MHz. When the TCS pin is set high, this is the TTL clock input with a 10 MHz maximum rate.
- PH2:** Phase 2 clock input. MOS level, maximum clock rate 20 MHz. When using PH1 as TTL clock input (TCS high), PH2 must be tied to ground.
- CRE:** BPU Control Register Enable. A high signal on this pin enables the BPU's Control Register. The data on DQ0-DQ15 is latched into the BPU Control Register on the falling edge of the next PH2 clock. CRE must be synchronized with respect to the valid data and must be removed before the rising edge of the subsequent PH2 clock.
- DLE:** Data Latch Enable. A high signal on this pin enables the BPU's data input latch. The data on DQ0-DQ15 is latched into the BPU data input latch on the falling edge of the next PH2 clock. DLE must be synchronized with respect to the valid data and must be removed before the rising edge of the subsequent PH2 clock.

Pin Definitions (Continued)

- PDLE:** Pixel port Data Latch Enable. A high signal on this pin enables the BPU's pixel port data input latch. The data bit on PDQn pin is latched into the one-bit pixel input data latch on the falling edge of the next PH2 clock. PDLE must be synchronized with respect to the valid pixel data and must be removed before the rising edge of the subsequent PH2 clock.
- DOS:** Data Output Select. DOS selects the data output from either the FIFO (DOS = 1) or the BITBLT logic block (DOS = 0).
- RESET:** FIFO control Reset. A low signal on this pin resets the BPU's FIFO read/write control circuitry. Data previously stored in the FIFO or on-chip latches are unchanged. This pin is controlled by the RGP's RESET line, an open-drain I/O pin. This input must be synchronized with respect to the falling edge of PH2. PA2/FWR and PA3/FRD inputs must be low 1 clock cycle prior to and 2 clock cycles after asserting RESET low in order to correctly reset the FIFO counters.
- LVCC:** Positive supply for on-chip logic circuits. 5V ± 10%
- LGND:** Ground for on-chip logic.
- BVCC0-BVCC1:** Positive supply for output buffers, two pins. 5V ± 10%.
- BGND0-BGND3:** Ground for output buffers, four pins.

BITBLT Fundamentals

BITBLT, BIT-aligned Block Transfer, is a general operator that provides a mechanism to move an arbitrary size rectangle of an image from one part of the frame buffer to another. During the data transfer process a bitwise logical operation can be performed between the source and the destination data. BITBLT is also called RasterOp: operations on rasters. It defines two rectangular areas, source and destination, and performs a logical operation (e.g., AND, OR XOR) between these two areas and stores the result back to the destination. It can be expressed in simple notation as:

Destination ← Source op Destination.
op: AND, OR, XOR, etc.

FRAME BUFFER ARCHITECTURE

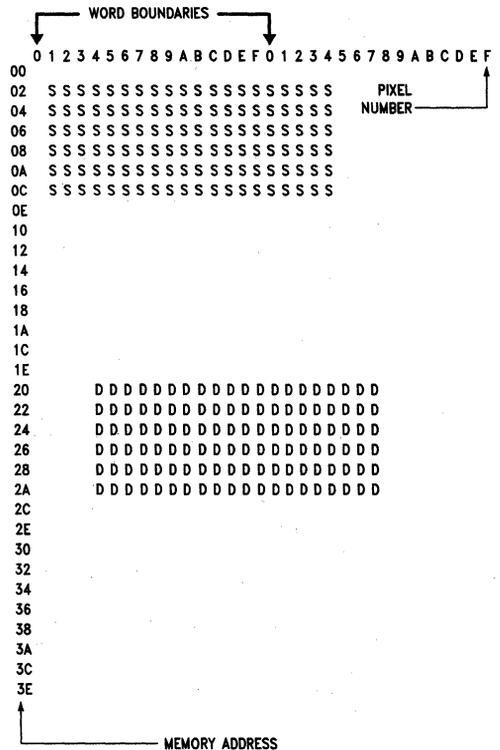
Generally, there are two kinds of frame buffer architectures: **PLANE-oriented** or **PIXEL-oriented**. BITBLT takes advantage of the plane-oriented frame buffer architecture's attribute of multiple, adjacent pixels-per-word, facilitating the movement of large blocks of data quickly in a frame buffer. However, the plane-oriented architecture has one inherent problem: the limit of resolution for memory addressing and access is the word, rather than the pixel. The BITBLT source starting address, the BITBLT destination starting address, the BITBLT width and the BITBLT height are all defined in pixels. The BITBLT source data block may start and end at any bit position of any word, and the destination data block also may start and end at any bit position of any word.

BIT ALIGNMENT

Before a logical operation can be performed between the source and the destination data, the source data must first be bit aligned to the destination data. In *Figure 1*, the source data need to be shifted three bits to the right in order to align the first pixel (that is, the pixel at the top left corner) in the source data block to the first pixel in the destination data block. For maximum performance, this alignment function must be implemented with a barrel shifter.

WORD BOUNDARIES AND DESTINATION MASKS

Each BITBLT destination scan line may start and end at any position in any data word. The neighboring bits (the bits sharing the same word address with any words in the destination data block, but not a part of the actual BITBLT rectangle) of the BITBLT destination scan line must remain unchanged after the BITBLT. Due to the plane-oriented frame buffer architecture, all memory operations must be word-aligned. In order to preserve the neighboring bits surrounding the BITBLT destination block, a left mask is needed for all the leftmost data words of the destination block, and a right mask is needed for all the rightmost data words of the destination data block. Both the left mask and the right mask remain the same throughout a given BITBLT operation.



TL/F/8672-3
FIGURE 1. A 32 by 32 Frame Buffer

BITBLT Fundamentals (Continued)

The following example illustrates the bit alignment requirement. In this example, the graphics controller has a 16-bit wide data bus. *Figure 1* shows a 32 pixel by 32 scan line frame buffer which is organized as a long bit stream from which wraps around every two words (32 bits). Further, the top left corner of the frame buffer starts from the lowest word in the memory, address 000hex. Each word in the memory contains 16 bits, DQ0–DQ15. The most significant bit of a memory word, DQ15, is defined as the first *displayed* pixel in a word. In other words, memory's DQ15 to DQ0 correspond to pixels 0 to 15 respectively. In this example, BITBLT addresses are expressed in terms of pixel number, starting (with 0) from the upper-leftmost pixel. The BITBLT source starting address is set to 021hex (the second pixel in the third word). The BITBLT destination starting address is set to 204hex (the fifth pixel in the 33rd word). The BITBLT width is set to 013hex (= 19 decimal, corresponding to a width of 20 pixels). The BITBLT height is set to 005hex (= 5 decimal, corresponding to 6 scan lines).

The left BITBLT mask for the above example is:

0000,1111,1111,1111

The right BITBLT mask for the above example is:

1111,1111,0000,0000

Note: Zeros in either the left mask or the right mask indicate the destination bits which will not be modified.

BITBLT DIRECTIONS

The BITBLT moves a rectangular block of data in a frame buffer. For a plane-oriented frame buffer, the BITBLT process can be considered a subroutine which has two nested loops. The loops are preceeded by the BITBLT setup computations. The outer loop is the BITBLT source and destination scan line pixel starting address calculation and line count test for completion. The innermost loop is the actual BITBLT data movement for a single BITBLT scan line and word count test for completion. The length of the innermost loop is the word count of the BITBLT width. The length of the second loop is equal to the BITBLT's height (number of scan lines involved in a BITBLT):

```
BITBLT: calculate BITBLT setup parameters ;once per BITBLT
        such as
        width, height
        bit misalignment (shift number)
        left, right masks
        horizontal, vertical directions
        etc
```

```
OUTERLOOP: calculate source, dest addresses ;once per scanline
```

```
    INNERLOOP: move data and increment addresses ;once per word
```

```
    UNTIL      done horizontally
```

```
UNTIL      done vertically
```

```
RETURN     (from BITBLT).
```

Each loop can be executed in one of two directions: the inner loop from left to right or right to left, the outer loop from top to bottom (down) or bottom to top (up).

The ability to move data starting from any corner of the BITBLT rectangle is necessary to avoid destroying the BITBLT source data as a result of destination writes when the source and destination are overlapped (i.e., when they share pixels). This situation is routinely encountered while panning or scrolling.

A determination of the correct execution directions of the BITBLT must be performed whenever the source and destination rectangles overlap. *Any* overlap will result in the destruction of source data (from a destination write) if the correct vertical direction is not used. Horizontal BITBLT direction is of concern only in certain cases of overlap, as will be explained below.

Figure 2 (a) and (b) illustrate two cases of overlap. Here, the BITBLT rectangles are three pixels wide by five scan lines high; they overlap by a single pixel in (a) and a single column of pixels in (b). For purposes of illustration, the BITBLT is assumed to be carried out pixel-by-pixel. This convention does not affect the conclusions.

In *Figure 2(a)*, if the BITBLT is performed in the UP direction (bottom-to-top) one of the transfers of the bottom scan line of the source will write to the circled pixel of the destination. Due to the overlap, this pixel is also part of the uppermost scan line of the source rectangle. Thus, data needed later is destroyed. Therefore, this BITBLT must be performed in the DOWN direction. Another example of this occurs any time the screen is moved in a purely vertical direction, as in scrolling text. It should be noted that, in both of these cases, the choice of horizontal BITBLT direction may be made arbitrarily.

Figure 2(b) demonstrates a case in which the horizontal BITBLT direction may not be chosen arbitrarily. This is an

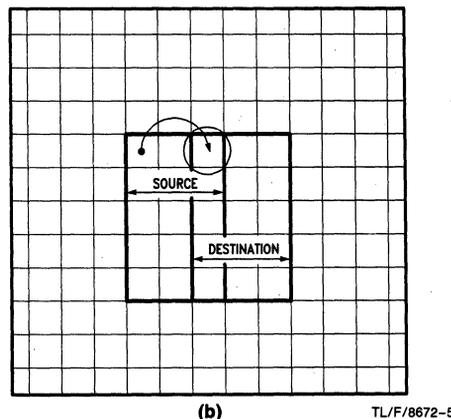
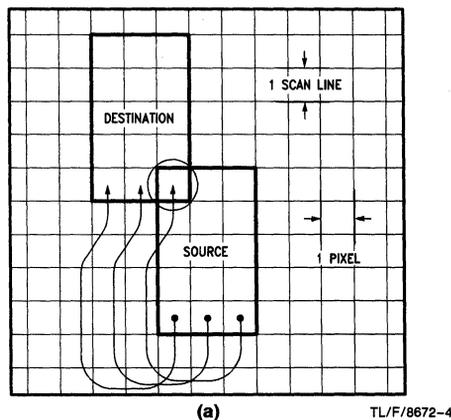


FIGURE 2. Overlapping BITBLT Operations

BITBLT Fundamentals (Continued)

instance of purely horizontal movement of data (panning). Because the movement from source to destination involves data within the same scan line, the incorrect direction of movement will overwrite data which will be needed later. In this example, the correct direction is from right to left.

BITBLT VARIATIONS

Some implementations of BITBLT are defined in terms of three operands: source, destination and mask/texture. This third operand is commonly used in monochrome systems to incorporate a stipple pattern into an area. These stipple patterns provide the appearance of multiple shades of gray in single-bit-per-pixel systems, in a manner similar to the 'half-tone' process used in printing.

Destination ← Texture op1 Source op2 Destination

While the BPU is essentially a two-operand device, three-operand BITBLT can be implemented quite flexibly and efficiently by performing the two operations serially. The BPU permits the use of any of its sixteen operations for each of the two operators shown above as 'op1' and 'op2'. Additionally, the on-chip FIFO can be used to store the intermediate result (the result of op1 later used as an operand of op2) thus minimizing the number of memory accesses required.

ENHANCING BITBLT PERFORMANCE

There are various ways to enhance BITBLT performance (speed). The simplest way is to try to get data in and out of the memory system quickly. Most of the bitmapped graphics systems utilize DRAMs for both cost and storage density reasons. Since the BITBLT data shows strong locality, the graphics system can take advantage of certain fast memory access modes available to the DRAMs, such as page mode access, static column access, etc. The BPU, by means of an internal FIFO, can pipe the BITBLT source data to reduce the frequency of switching out of the current page address space, thus maximizing the ability of the system to capitalize on the data's locality. This operation is described in the following section.

PIPING THE BITBLT SOURCE DATA

When the BITBLT width is more than a word, up to 16 source data words can be piped into the BPU's on-chip FIFO. At the end of each BITBLT scan line or at the end of 16 source data words, the controller switches from the BITBLT source address space to the BITBLT destination address space. When the BITBLT destination data word is fetched, two possible memory control sequences can be used. One is the modify-write sequence: write the BITBLT result back to the destination memory immediately after the logical function is executed. The second sequence involves storing the BITBLT result back to the BPU's on-chip FIFO, to a maximum of 16 words. Either at the end of each BITBLT destination scan line or at the 16th destination data word, the BITBLT resultant data is then read out from the FIFO and written to the BITBLT destination memory sequentially.

Summary of the BITBLT Memory Control Sequences

BITBLT MEMORY SEQUENCE I

- 0) Load the BPU Control Register with [FS, SN, LM, RM], via the data bus.

- 1) Read in the BITBLT source data up to 16 words (17 words in certain cases), barrel-shift, then write them into the on-chip FIFO. (Only 16 barrel-shifted data words can be stored.)
- 2) Read in the BITBLT destination data while the barrel-shifted source data is read out from the on-chip FIFO and the selected logical operation is executed, then write back to destination.
- 3) Go to step 1 until the end of the BITBLT scan line.
- 4) Go to step 1 for the remaining BITBLT scan lines.

BITBLT MEMORY SEQUENCE II

- 0) Load the BPU Control Register with [FS, SN, LM, RM] via the data bus.
- 1) Read in the BITBLT source data up to 16 words (17 words in certain cases), barrel-shift, then write them into the on-chip FIFO. (Only 16 barrel-shifted data words can be stored.)
- 2) Read in the BITBLT destination data in sequence, execute the selected logical operation and then write the result back to the on-chip FIFO, maximum 16 BITBLT data words.
- 3) Read BITBLT result from the FIFO and write them back to the BITBLT destination memory.
- 4) Go to step 1 until the end of the BITBLT scan line.
- 5) Go to step 1 for the remaining BITBLT scan lines.

DP8510 BIT ORDER

The DP8510's internal word bit-order is defined from D15 on the left to D0 on the right. The pixel address bits PA0-PA3 map directly to this left-to-right bit order convention. For example when PA(3:0) = 0 the rightmost bit of a displayed word will be addressed. When PA(3:0) = 0fh, the leftmost bit of a displayed word will be addressed (see *Figure 3b*).

It should be noted that when pixel data is viewed in the frame buffer the order of the bits will be consistent with most conventional microprocessor conventions that define D0 in the LSB bit position or in the rightmost bit location (see *Figure 3a*).

Block Diagram Description

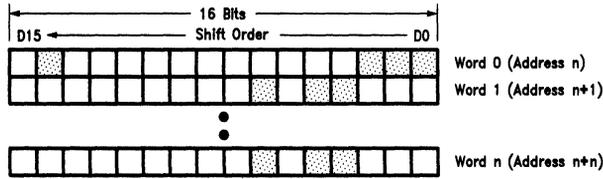
GENERAL

Figure 4 illustrates the block diagram of the BPU. It consists of several 16-bit latches and multiplexers connected via 16-bit data paths to the three major functional blocks: the Barrel Shifter, the FIFO and the BITBLT Logic Unit. Latches are provided for input data from the BITBLT source and destination as well as for control parameters.

Control parameters consist of 16-bit data which are written to the Control Register. This data is used, in conjunction with the BPU's external control pins, to route and modify the BITBLT data.

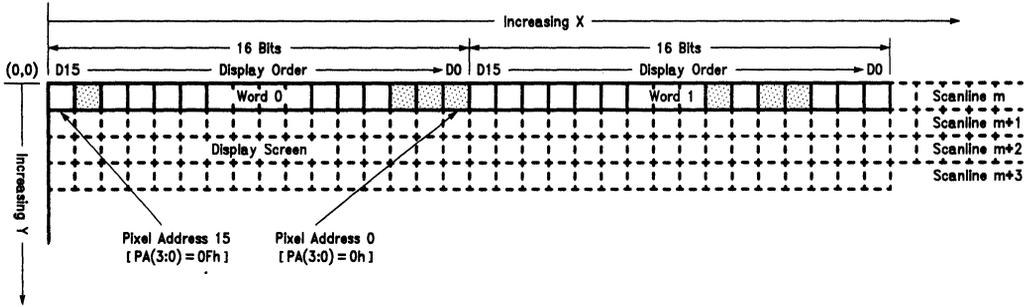
The primary flow of data during a BITBLT is from the DQ0-DQ15 pins, through the Master Data Input Latch (DIL-Master) to either the source path or destination path to the BITBLT Logic Unit. The output of the logic unit is then routed through tri-state buffers back to the DQ0-DQ15 pins.

Block Diagram Description (Continued)



TL/F/8672-26

**a) Data Representation In Frame Buffer
(as Seen by the CPU)**



TL/F/8672-27

b) Data Representation In DP8510-Based Display
**FIGURE 3. Frame Buffer vs. Display
Coordinate Bit-Order Representation**

The source path consists of the Source Data Input Latch (DIL-Source), the Barrel Input Latch (BIL), the Barrel Shifter and the FIFO. The destination path consists of only the Destination Data Input Latch (DIL-Destination). The destination path terminates directly at one input port to the logic unit. The logic unit's other input port receives either the output of the FIFO or the output of the Pixel port data Input Latch (PIL), as determined by the state of the \bar{B}/L pin, for BITBLTs and line drawing, respectively.

The output of the logic unit is routed through multiplexers to the output drivers for both DQ0-DQ15 and PDQn. In the case of DQ0-DQ15, the multiplexer permits the FIFO output, rather than the logic unit output, to be routed to the output drivers. The PDQn port, being a single bit port, is driven by one of the sixteen output bits of the logic unit, as selected by the multiplexer, according to the state of the pixel address lines, PA0-PA3.

Block Diagram Description (Continued)

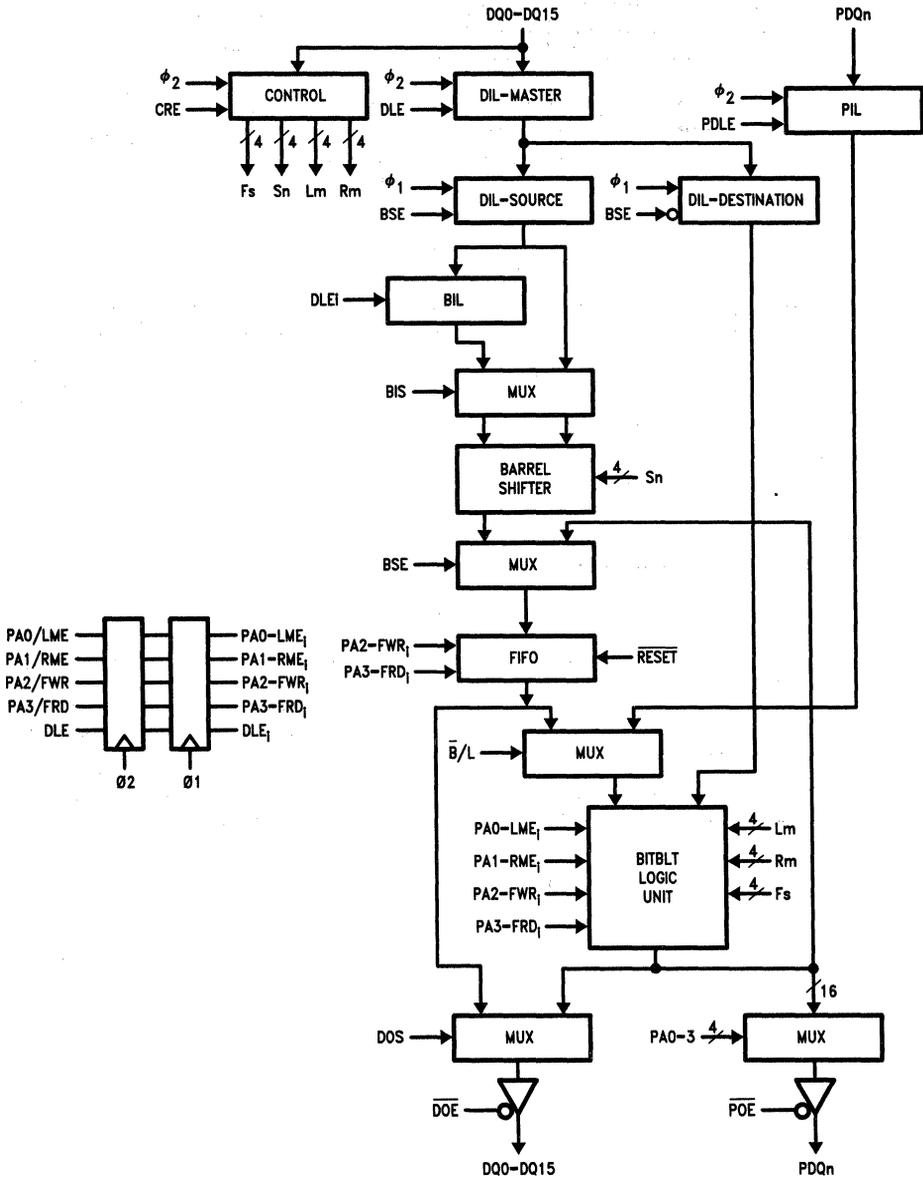


FIGURE 4. DP8510 BPU Detailed Block Diagram

TL/F/8672-6

Functional Block Description

BPU CONTROL REGISTER

The BPU Control Register consists of 4 fields, each field being four bits wide. The BITBLT Function Select field, FS, selects one of the 16 BITBLT operations. The barrel Shift Number field, SN, controls the number of bit positions shifted by the barrel shifter. The Left Mask field, LM, sets the BITBLT left mask pattern. The Right Mask field, RM, sets the BITBLT right mask pattern.

The SN will be explained in the barrel shifter section. The LM and the RM will be detailed in the BITBLT logic block section.

D15											D0																				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS				SN				LM				RM																			

FIGURE 5. Control Register Fields

The content of this register must be loaded prior to the first BITBLT operation by asserting CRE (Control Register Enable). Only the Function Select, FS, is valid when the BPU is in the line-drawing mode. The SN, RM and LM have no effect in the line-drawing mode.

FUNCTION SELECT

The 16 classical BITBLT functions supported by the BPU are described in Table I.

SHIFT NUMBER

There are 16 different barrel shift positions controlled by the SN of the Control Register.

Let $A = a_{15} \dots a_0$, $B = b_{15} \dots b_0$ and $C = c_{15} \dots c_0$, A and B are the input words, C is the output. An i-bit barrel-shift operation on A and B denoted by $C = S(i;A,B)$ is the operation of concatenating the most significant (i) bits of word B to the least significant (16-i) bits of word A.

When $A = B$, the $C = S(i;A, B)$ is equivalent to having i-bit circular left shift. The i-bit circular right shift is equivalent to the (16-i) bit circular left shift.

The truth table of the barrel-shift operation $S(i;A, B)$ is shown in Table II. (i = shift number)

LEFT AND RIGHT MASKS

The Truth Table for the left and right mask is shown in Table III. Zeroes in the mask patterns indicate bits in the destination data which will be preserved during the BITBLT operation.

Both the LM and the RM can be invoked simultaneously. This may be necessary where the BITBLT width is less than 16 pixels.

TABLE I. BITBLT Function Definitions

#	f3	f2	f1	f0	
0	0	0	0	0	0
1	0	0	0	1	-s AND -d
2	0	0	1	0	-s AND d
3	0	0	1	1	-s
4	0	1	0	0	s AND -d
5	0	1	0	1	-d
6	0	1	1	0	s XOR d
7	0	1	1	1	-s OR -d
8	1	0	0	0	s AND d
9	1	0	0	1	s XNOR d
10	1	0	1	0	d
11	1	0	1	1	-s OR d
12	1	1	0	0	s
13	1	1	0	1	s OR -d
14	1	1	1	0	s OR d
15	1	1	1	1	1

Note: d: destination
s: source

f3-0: function select code, which selects one of the 16 BITBLT functions.

TABLE II. Truth Table for Barrel-Shift Operation $S(i;A, B)$

i	c15	c14	c13	c12	c11	c10	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0
0	a15	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
1	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	b15
2	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	b15	b14
3	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	b15	b14	b13
4	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a1	b15	b14	b13	b12
5	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11
6	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10
7	a8	a7	a6	a5	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9
8	a7	a6	a5	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8
9	a6	a5	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7
10	a5	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6
11	a4	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5
12	a3	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4
13	a2	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3
14	a1	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2
15	a0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1

Functional Block Description (Continued)

TABLE III. Truth Table for Left and Right Mask

	D15	D0
LM = 0, L-mask = 1111	1111	1111
LM = 1, L-mask = 0111	1111	1111
LM = 2, L-mask = 0011	1111	1111
LM = 3, L-mask = 0001	1111	1111
LM = 4, L-mask = 0000	1111	1111
LM = 5, L-mask = 0000	0111	1111
LM = 6, L-mask = 0000	0011	1111
LM = 7, L-mask = 0000	0001	1111
LM = 8, L-mask = 0000	0000	1111
LM = 9, L-mask = 0000	0000	0111
LM = 10, L-mask = 0000	0000	0011
LM = 11, L-mask = 0000	0000	0001
LM = 12, L-mask = 0000	0000	0000
LM = 13, L-mask = 0000	0000	0000
LM = 14, L-mask = 0000	0000	0000
LM = 15, L-mask = 0000	0000	0000
RM = 0, R-mask = 1000	0000	0000
RM = 1, R-mask = 1100	0000	0000
RM = 2, R-mask = 1110	0000	0000
RM = 3, R-mask = 1111	0000	0000
RM = 4, R-mask = 1111	1000	0000
RM = 5, R-mask = 1111	1100	0000
RM = 6, R-mask = 1111	1110	0000
RM = 7, R-mask = 1111	1111	0000
RM = 8, R-mask = 1111	1111	1000
RM = 9, R-mask = 1111	1111	1100
RM = 10, R-mask = 1111	1111	1110
RM = 11, R-mask = 1111	1111	1111
RM = 12, R-mask = 1111	1111	1111
RM = 13, R-mask = 1111	1111	1111
RM = 14, R-mask = 1111	1111	1111
RM = 15, R-mask = 1111	1111	1111

1 = Enable Logic OP
0 = Disable Logic OP

BARREL-SHIFTER

The function of the barrel shifter is to align the BITBLT source data to the destination data. Bit alignment may cross word boundaries.

The barrel shifter in the BPU is implemented as a 32- to 16-bit multiplexer. Depending upon the type of the BITBLT and the BITBLT length, the necessary source data word(s) are fetched into the BPU forming a 32-bit input to the barrel shifter. Barrel Input Latch (BIL) stores the first source data word fetched and the Source Data Input Latch (DIL-source) stores the subsequent source data word.

A multiplexer precedes the barrel shifter to swap the input words if necessary. That is, Barrel Input Select (BIS) causes BIL (and DIL-Source) to be routed to the left or right (and right or left) half of the barrel shifter. Note that this is a word-level swap; it does not affect the ordering of the bits within the words. This swap mechanism facilitates the fetching of BITBLT data from left-to-right or right-to-left.

When the BITBLT direction is set from the left to the right (the source data words are fetched starting from the left-hand side of the BITBLT rectangle), the BIS pin should be set low. If the BITBLT direction is set from the right to the left, BIS must be set high to exchange the source data sequence.

Figure 6 depicts the data path in the BPU's barrel-shifter block.

The Barrel Input Latch (BIL) is the BITBLT source data pipeline register. It is loaded (from DIL-Source) one clock cycle after DIL-Source is loaded. Therefore, when the system fetches a BITBLT source word from memory (BSE asserted), this word will be first loaded into DIL-Source on the rising edge of a PH1; it will then be transferred to BIL on the rising edge of the following PH1. As a result, if the two words have been fetched in temporal sequence, the first word fetched will be in BIL; the second word will be in DIL-Source. It should be noted that this condition (different words in BIL and DIL-Source) will be true for only one clock period, since the next rising edge of PH1 will again cause BIL to be loaded from DIL-Source. Therefore, the shifted result (the output of the barrel shifter) must be written to the FIFO on the rising edge of PH1 immediately following the PH1 used to load DIL-Source.

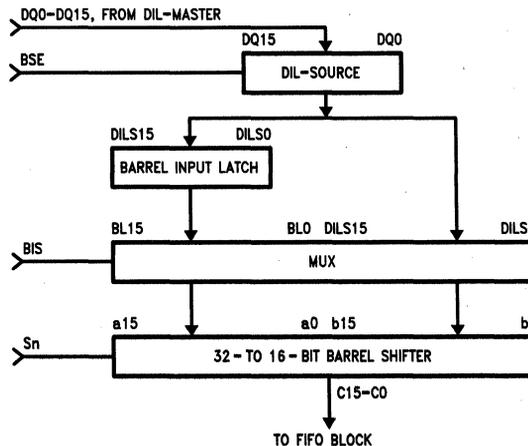


FIGURE 6. BPU Barrel-Shifter Block

TL/F/8672-7

Functional Block Description (Continued)

Source Data Multiplexing, an Example:

Figure 7 shows an example of the BPU Barrel Shifter operation. Let Word A = aF..a0, Word B = bF..b0, SN = S3..S0 and BSE = 1. In both cases below, Word B is fetched after Word A.

FIFO

The FIFO, 16 bits by 16 words, constitutes the BPU's on-chip storage and is implemented with dual-port RAM. The FIFO has separate READ/WRITE controls, the FIFO Write

(FWR) and the FIFO Read (FRD). The use of faster, localized memory access modes (e.g., page, static column) is supported by the BPU via the FIFO, since the FIFO provides storage for multiple, barrel-shifted source words.

Figure 8 depicts the BPU FIFO structure.

Case I. If BIS = 0,

Word A	B
F E D C B A 9 8 7 6 5 4 3 2 1 0	F E D C B A 9 8 7 6 5 4 3 2 1 0
SN	
0	aFaEaDaCaBaAa9a8a7a6a5a4a3a2a1a0
1	aEaDaCaBaAa9a8a7a6a5a4a3a2a1a0bF
2	aDaCaBaAa9a8a7a6a5a4a3a2a1a0bFbE
3	aCaBaAa9a8a7a6a5a4a3a2a1a0bFbEbD
4	aBaAa9a8a7a6a5a4a3a2a1a0bFbEbDbC
5	aAa9a8a7a6a5a4a3a2a1a0bFbEbDbCbB
6	a9a8a7a6a5a4a3a2a1a0bFbEbDbCbBbA
7	a8a7a6a5a4a3a2a1a0bFbEbDbCbBbAb9
8	a7a6a5a4a3a2a1a0bFbEbDbCbBbAb9b8
9	a6a5a4a3a2a1a0bFbEbDbCbBbAb9b8b7
A	a5a4a3a2a1a0bFbEbDbCbBbAb9b8b7b6
B	a4a3a2a1a0bFbEbDbCbBbAb9b8b7b6b5
C	a3a2a1a0bFbEbDbCbBbAb9b8b7b6b5b4
D	a2a1a0bFbEbDbCbBbAb9b8b7b6b5b4b3
E	a1a0bFbEbDbCbBbAb9b8b7b6b5b4b3b2
F	a0bFbEbDbCbBbAb9b8b7b6b5b4b3b2b1

Case II. If BIS = 1 (The BIS controls the multiplexing before the barrel shifter),

Word B	A
F E D C B A 9 8 7 6 5 4 3 2 1 0	F E D C B A 9 8 7 6 5 4 3 2 1 0
SN	
0	bFbEbDbCbBbAb9b8b7b6b5b4b3b2b1b0
1	bEbDbCbBbAb9b8b7b6b5b4b3b2b1b0aF
2	bDbCbBbAb9b8b7b6b5b4b3b2b1b0aFaE
3	bCbBbAb9b8b7b6b5b4b3b2b1b0aFaEaD
4	bBbAb9b8b7b6b5b4b3b2b1b0aFaEaDaC
5	bAb9b8b7b6b5b4b3b2b1b0aFaEaDaCaB
6	b9b8b7b6b5b4b3b2b1b0aFaEaDaCaBaA
7	b8b7b6b5b4b3b2b1b0aFaEaDaCaBaAa9
8	b7b6b5b4b3b2b1b0aFaEaDaCaBaAa9a8
9	b6b5b4b3b2b1b0aFaEaDaCaBaAa9a8a7
A	b5b4b3b2b1b0aFaEaDaCaBaAa9a8a7a6
B	b4b3b2b1b0aFaEaDaCaBaAa9a8a7a6a5
C	b3b2b1b0aFaEaDaCaBaAa9a8a7a6a5a4
D	b2b1b0aFaEaDaCaBaAa9a8a7a6a5a4a3
E	b1b0aFaEaDaCaBaAa9a8a7a6a5a4a3a2
F	b0aFaEaDaCaBaAa9a8a7a6a5a4a3a2a1

FIGURE 7

Functional Block Description (Continued)

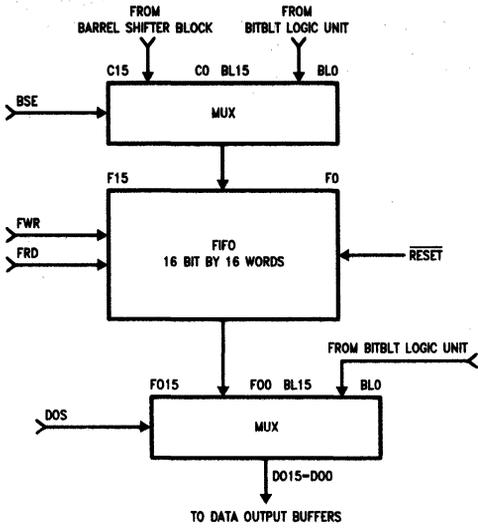


FIGURE 8. BPU FIFO Block

TL/F/8672-8

BITBLT LOGIC CONTROL UNIT

This block performs the selected BITBLT logical operation and the BITBLT destination data masking.

The 16 BITBLT functions can be expressed as:

$$\begin{aligned}
 f3 & * (s * d) + \\
 f2 & * (s * -d) + \\
 f1 & * (-s * d) + \\
 f0 & * (-s * -d)
 \end{aligned}$$

where f0-f3 are the bits of the Function Select field, FS.

LINE DRAWING MODE ($\bar{B}/L = 1$)

Line drawing mode differs from BITBLT mode in the following respects:

- 1) the FIFO is not used in line drawing mode.
- 2) the Shift Number parameter is ignored in the line drawing mode
- 3) the Left Mask and Right Mask parameters are ignored in the line drawing mode. Instead, masks are generated in matched pairs, as needed, according to the current state of the Pixel Address lines PA0-PA3.

For example, if the pixel address (the binary value on PA0-3) is 12 decimal (PA3 = PA2 = 1, PA1 = PA0 = 0), then the following masks will be generated:

	MS bit	LS bit
	↓	↓
	LM = 0000 0000 0000 1111	
and	RM = 1111 1111 1111 1000	

Since these masks are used simultaneously when the destination word is generated (during line drawing mode), and since zeroes in the mask indicate destination bits that are unaffected (are preserved) by the BITBLT operation, the AND of the two masks indicates the bit that will be affected by the BITBLT. In this example, bit 3 of the memory word is selected.

It is important to note this mapping convention of the BPU: the logical pixel (indicated by PA0-3) associated with the current line drawing BITBLT operation corresponds to a physical bit (indicated by the ones-complement of PA0-3) in the current memory word. That is, logical pixel 0 corresponds to DQ15 in any word. Logical pixel 1 corresponds to DQ14 in any word, etc.

- 4) Conceptually, during line drawing, there is no "source", in the conventional BITBLT sense of replicating an item from one set of storage locations to another set of storage locations. Rather, the line is "created" bit-by-bit by an address-generating algorithm. However, a mechanism is provided in the BPU to establish a single bit which can be logically combined (in the BITBLT Logic Unit) with each bit of the destination. This mechanism is embodied in the pixel data port and its associated input latch (PIL).
- 5) the PIL bit is replicated into all 16 bits of the logic unit's source input port in line drawing mode.
- 6) a single bit of the logic unit's output port is selected to drive the PDQn output buffer, according to PA0-PA3, in line drawing mode. In BITBLT mode, this multiplexer is at TRI-STATE®.

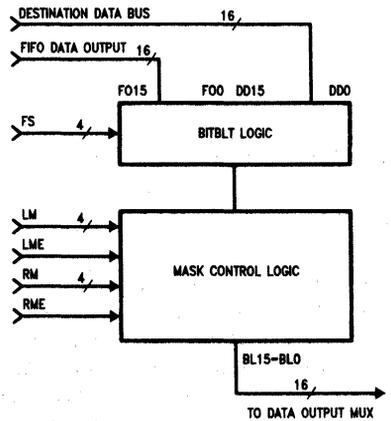


FIGURE 9. BPU Logical Control Unit

TL/F/8672-9

Applications of the DP8510 BPU

SYSTEM CONTROL SEQUENCE

Figure 10 illustrates a typical control and memory access sequence that might be used to accomplish BITBLT with the BPU. In this example, memory access time is sufficiently fast to avoid wait states (the BPU is capable of accepting a new operand at each clock). All control strobes are assumed to be generated by the CPU/state machine which is serving as the BPU's controller.

In this example, a single operation is performed per clock cycle for the sake of clarity. Overlap of memory access with internal BPU operations would reduce the number of clock cycles.

Clock	Action
1	address for source operand 1 is generated, first RAM access is started, data becomes valid at end of cycle
2	source operand 1 is latched in data input latch, address for source operand 2 is generated, second RAM access is started, data becomes valid at end of cycle
3	source operand 2 is latched in data input latch (forming the 32-bit input to the barrel shifter), data propagates through the barrel shift logic and becomes valid at the input to the FIFO
4	the barrel-shifted source word is written to the FIFO, the BPU's source path is disabled, the BPU's destination path is enabled
5	the address for the destination word is generated, the destination RAM access is started, data becomes valid at end of cycle
6	destination data is latched into data input latch, source word is read from internal FIFO, BITBLT logic unit produces resultant word
7	output buffers are enabled destination write to RAM occurs.

FIFO STRUCTURE, OPERATION AND RESTRICTIONS

The DP8510 BPU's FIFO is implemented as a file of sixteen 16-bit registers. The file has separate ports for read and write operations. Register selection is accomplished via a pair of shift-register ring counters. One of these is used during FIFO write cycles and is incremented by FIFO Write (FWR); the other is used during FIFO read cycles and is incremented by FIFO Read (FRD). At RESET, both ring counters are set to enable FIFO register 0. Both ring counters will wrap around; that is, they will select the FIFO registers in the sequence:

0, 1, 2, . . . , 13, 14, 15, 0, 1, . . .

During a FIFO write cycle, the FIFO register currently selected by the WRITE ring counter will be written with the appropriate internally-sourced data (barrel shifter output or logic unit output, according to the state of BSE); the WRITE ring counter will then be incremented to select the next FIFO register.

The FIFO read operation is implemented in a read-ahead manner as follows: the contents of the FIFO register selected by the FIFO READ ring counter is copied to a master

latch (of a master-slave pair) during each PH2 regardless of the state of FRD; however, the contents of this latch are not passed on to the slave (output) latch until the next PH1 when FRD is active. This FRD also causes the FIFO READ ring counter to be incremented to the next FIFO register, initiating the read-ahead of that register in anticipation of the next FIFO read cycle.

In general, a given clock cycle can accomplish a simultaneous FIFO read and FIFO write. However, two restrictions apply to FIFO operations. First, a simultaneous FIFO read and FIFO write should apply to *different* FIFO registers; that is, the SAME register should NOT be read and written in the same cycle. Failure to meet this requirement is not destructive, but the result may be contrary to what the user intends: the output value will be the "old" contents of the currently-selected FIFO register, not the "new" value being written to that register. This behavior is a result of the read-ahead operation in conjunction with the two-phase master-slave output latch.

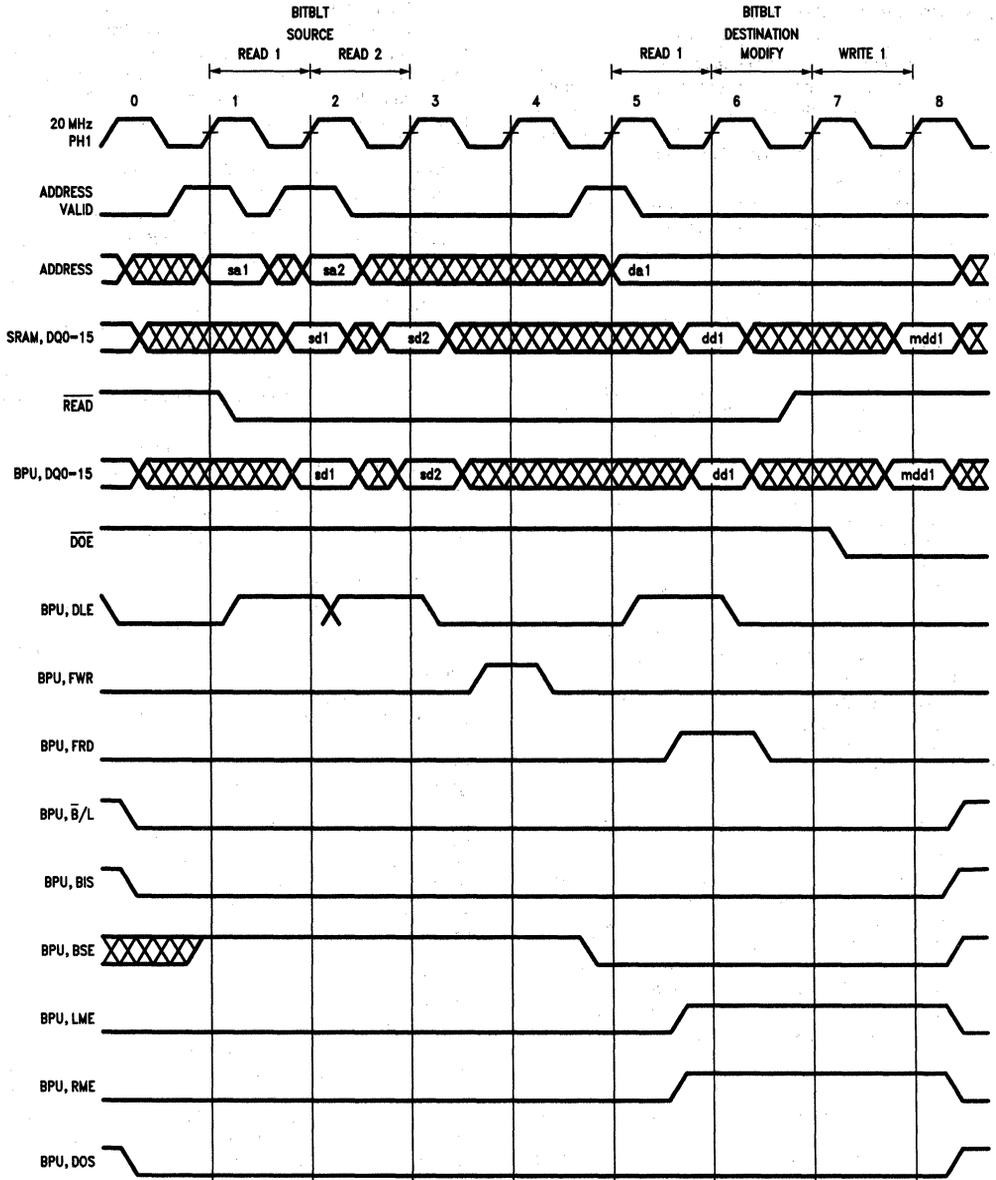
The second restriction is a speed-dependent one; it does not apply at clock frequencies of 10 MHz and below. At higher clock speeds, a two clock-cycle minimum is imposed between the FIFO write to a given location and the corresponding FIFO read of that location. Failure to meet this restriction may result in incorrect data being read from the FIFO. Consider the case of the FIFO at RESET: if the first FIFO write is performed during clock cycle (m), the first FIFO read (which is the read of that location) may not take place until clock cycle (m+2). An equivalent "FIFO-initially-empty" case occurs whenever the (n)th FIFO location is written and the (n-1)th FIFO location is read during the same clock cycle. This restriction does not impair the ability of the BPU to perform consecutive FIFO reads and/or FIFO writes at one-clock intervals.

BPU USED WITH A CPU OR STATE MACHINE

The BPU places no restrictions on the practical number of planes in a system; rather, systems can be expanded from one to any number of planes without compromising BITBLT performance. Data movement within each plane is carried out in parallel by a dedicated BPU for each plane; as a result, a BITBLT can be performed for any number of planes in the same period of time as in a single-plane monochrome system.

Systems currently implementing BITBLT completely in software can realize a major performance increase through the use of the BPU. In this case, the BPU can be thought of as a hardware accelerator for BITBLT. Two examples of the BPU in this application are shown in Figures 11 and 12. In both examples, the CPU is responsible for BITBLT preprocessing, such as the determination of direction, initial BITBLT addresses, etc. These examples illustrate the variety of cost/complexity versus performance tradeoffs that may be made. While Figures 11 and 12 illustrate a single plane application, extension to multiple planes can be achieved by implementing a BPU on each bitplane which will result in virtually the same performance while delivering a greater number of shades or colors.

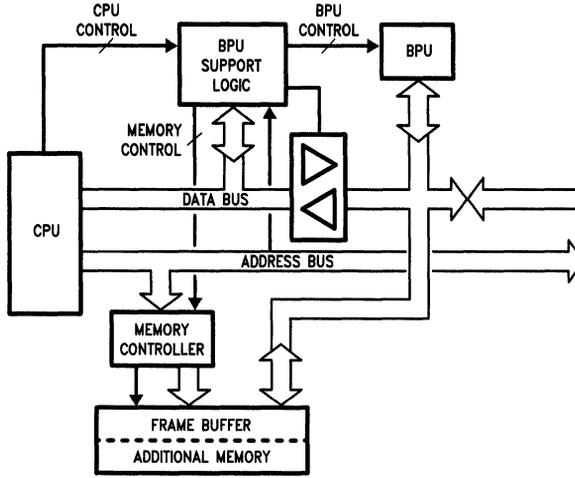
Applications of the DP8510 BPU (Continued)



TL/F/8672-10

**FIGURE 10. Typical BITBLT Timing
(Using Fast Static RAM)**

Applications of the DP8510 BPU (Continued)

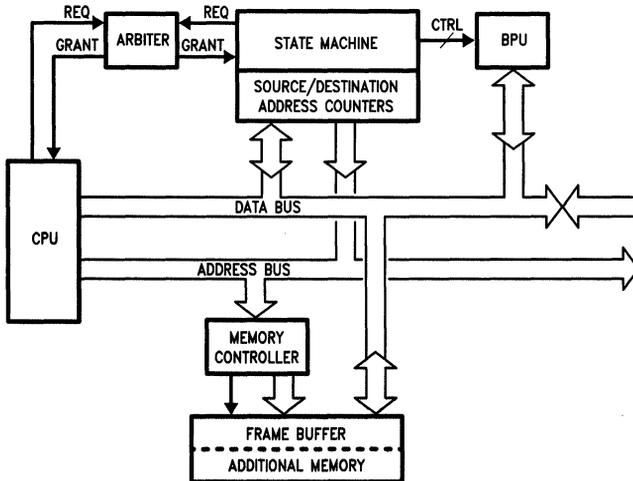


TL/F/8672-12

FIGURE 11. BPU with CPU using String/Block Move instructions

Figure 11 presents a solution requiring minimal hardware but whose performance may be limited by the CPU's bus cycle time. In this example, the CPU's block move or string move instruction is used to produce the source and destination addresses for the BITBLT. The BPU, in conjunction with its support logic, intercepts the source data during the read cycle. The support logic is also responsible for turning the string/block instruction's write cycles into read-modify-writes to allow the BPU to receive the destination data, logically combine it with the source data, and return it to the destination address. The result is a relatively simple hardware/software implementation of a single scan line of the BITBLT.

Performance can be further enhanced, at the cost of greater hardware complexity, through an implementation as described in Figure 12. As in the previous example, this solution performs a single scan line of the BITBLT; however, in this case, the BITBLT proceeds without any involvement on the CPU's part. All intermediate addresses within the scan line are generated by up-down counters; all bus cycles are effected by the state machine's hardware. Within this scheme, further tradeoffs may be made. For example, the designer has the option of using the BPU's source pipeline to take advantage of faster memory access modes, such as page mode.



TL/F/8672-13

FIGURE 12. BPU with CPU and state machine

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature under bias

0°C to +70°C (Comm.)

-55°C to +125°C (Mil.)

Storage Temperature Range

-65°C to +150°C

All input or output voltage with respect to GND

-0.5V to +7V

Power Dissipation @ 20 MHz

0.5W

ESD rating is to be determined.

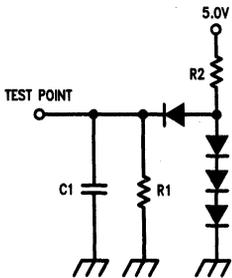
Note: Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.

DC Electrical Characteristics

T_A = 0° to 70° (Comm.), -55°C to +125°C (Mil.), V_{CC} = 5V ± 10%, GND = 0V

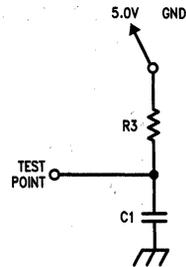
Symbol	Characteristics	Conditions	Min	Typ	Max	Unit
V _{IH}			2.0		V _{CC} + 0.5	V
V _{IL}			-0.5		0.8	V
V _{CH}	MOS Clock High	PH1, PH2 Pins Only, MOS	V _{CC} - 0.5		V _{CC} + 0.5	V
V _{CL}	MOS Clock Low	PH1, PH2 Pins Only, MOS	-0.5		0.3	V
V _{CLT}	MOS Clock Ringing	PH1, PH2 Pins Only, MOS	-0.5		0.5	V
V _{TCH}	TTL Clock High	TTL Clock/PH1 Pin Only	2.5			V
V _{TCL}	TTL Clock Low	TTL Clock/PH1 Pin Only			0.8	V
V _{OH}		I _{OH} = -3 mA	2.4			V
V _{OL}		I _{OL} = 3 mA			0.5	V
I _{IN}	Leakage Current	V _{IN} = V _{IH} or V _{IL}			± 10	μA
I _{OZ}	TRI-STATE® Leakage	V _O = V _{CC} or GND			± 10	μA
I _{CC1}	Quiescent Current	PH1, PH2 at 20 MHz			12	mA
I _{CC2}	Supply Current	PH1, PH2 at 100 kHz			12	mA
I _{CC3}	Supply Current	PH1, PH2 at 20 MHz			25	mA
C _{IN}	Input Capacitance	f _{in} at 1 MHz			10	pF
Φ _{JA}	Thermal Resistance—Ceramic PGA Package Junction to Ambient				59	°C/W
Φ _{JC}	Junction to Case				35	°C/W

Note: All output test conditions are 50 pF plus one TTL load.



TL/F/8672-14

FIGURE 13. BPU Output Test Load Circuitry



TL/F/8672-15

FIGURE 14. BPU Output TRI-STATE Test Load Circuitry

Note 1: C1 = 50 pF

R1 = 6 kΩ

R2 = 1.3 kΩ

R3 = 1.8 kΩ

Note 2: Connect SW to +5V for tpLZ and tpZL measurements.

Note 3: Connect SW to GND for tpHZ and tpZH measurements.

DP8510 AC Electrical Characteristics*

Name	Figure	Description	Conditions	Commercial		Military**	
				Min	Max	Min	Max
fmos		PH1, PH2 MOS Clock Frequency	TCS = 0		20 MHz		20 MHz
tc[mos]		PH1, PH2 MOS Clock Period	TCS = 0	50			50
fttl		PH1 TTL Clock Frequency	TCS = 1		17 MHz		17 MHz
tc[ttl]		PH1 TTL Clock Period	TCS = 1	58.8			58.8
tph1	16	PH1 High Time (TCS=0)	RE 50% to Next FE 50%	19		19	
tph1H	16	PH1 High Time (TCS=1)	RE 1.5V to Next FE 1.5V	17		17	
tph1L	16	PH1 Low Time (TCS=1)	FE 1.5V to Next RE 1.5V	17		17	
tph2	16	PH2 High Time (TCS=0)	RE 50% to Next FE 50%	19		19	
tck1	16	50% PH1 RE to 50% PH2 RE		25		25	
tck2	16	50% PH2 RE to 50% PH1 RE		22		22	
tnov1	16	Non-Overlap Time	PH2-to-PH1 50%	3		3	
tnov2	16	Non-Overlap Time	PH1-to-PH2 50%	3		3	
tcl	19	Data Switching Time	From L/ \bar{B} or DOS 1.5V	43		43	
tles2f	17	DLE, PDLE, CRE Setup Time (TCS=0)	Before PH2 FE 50%	5		5	
tles1r	17	DLE, PDLE, CRE Setup Time (TCS=1)	Before PH1 RE 1.5V	5		5	
tleh2f	17	DLE, PDLE, CRE Hold Time (TCS=0)	After PH2 FE 50%	5		5	
tleh1r	17	DLE, PDLE, CRE Hold Time (TCS=1)	After PH1 RE 1.5V	8		8	
tleis	17	DLE, PDLE, CRE Invalid	Before PH2 RE 50%	5		5	
tds2f	17	DIL, CR Data Setup Time (TCS=0)	Before PH2 FE 50%	5		5	
tdh2f	17	DIL, CR Data Hold Time (TCS=0)	After PH2 FE 50%	8		8	
tds1r	17	DIL, CR Data Setup Time (TCS=1)	Before PH1 RE 1.5V	5		5	
tdh1r	17	DIL, CR Data Hold Time (TCS=1)	After PH1 RE 1.5V	10		10	
tctls2f	17	$\overline{\text{RESET}}$ Setup Time (TCS=0)	Before PH2 FE 50%	5		5	
tctlh2f	17	$\overline{\text{RESET}}$ Hold Time (TCS=0)	After PH2 FE 50%	8		8	
tctls1r	17	$\overline{\text{RESET}}$ Setup Time (TCS=1)	Before PH1 RE 1.5V	5		5	
tctlh1r	17	$\overline{\text{RESET}}$ Hold Time (TCS=1)	After PH1 RE 1.5V	10		10	
tbseh2f	20	BSE Hold Time (TCS=0)	After PH2 FE 50%	5		5	
tbses1r	20	BSE Setup Time (TCS=1)	Before PH1 RE 1.5V	10		10	
tbseh1r	20	BSE Hold Time (TCS=1)	After PH1 RE 1.5V	5		5	
tbish2f	20	BIS Hold Time (TCS=0)	After PH2 FE 50%	5		5	
tbiss1r	20	BIS Setup Time (TCS=1)	Before PH1 RE 1.5V	15		15	
tbish1r	20	BIS Hold Time (TCS=1)	After PH1 RE 1.5V	5		5	
tpdn1r	21	Valid Pixel Data	After PH1 RE 50%		37		43
tdpZH	18	Data TRI-STATE to Active High	After $\overline{\text{DOE}}$, $\overline{\text{POE}}$ FE 1.5V		25		30
tdpZL	18	Data TRI-STATE to Active Low	After $\overline{\text{DOE}}$, $\overline{\text{POE}}$ FE 1.5V		25		30
tdpHZ	18	Data Active High to TRI-STATE	After $\overline{\text{DOE}}$, $\overline{\text{POE}}$ RE 1.5V		25		30
tdpLZ	18	Data Active Low to TRI-STATE	After $\overline{\text{DOE}}$, $\overline{\text{POE}}$ RE 1.5V		25		30
tdq1r	22	Valid Output Data (DOS=0)	After PH1 RE 50%		40		45
tdq1r	22	Valid Output Data (DOS=1)	After PH1 RE 50%		30		35
tvfwr	23	Valid FIFO Write (FWR)†	From DLE	1tc	1tc	1tc	1tc
tvfrd	23	Valid FIFO Read (FRD)††	From Valid FWR	2tc		2tc	

*Conditions

Comm.: $T_A = 0^\circ$ to $+70^\circ\text{C}$, $V_{CC} = 5V \pm 10\%$, GND = 0VMil.: $T_A = -55^\circ$ to $+125^\circ\text{C}$, $V_{CC} = 5V \pm 10\%$, GND = 0V

All values are expressed in nanoseconds unless otherwise specified.

**Military Specifications are preliminary. Please contact your National Semiconductor Sales Office or Distributor for availability and specifications.

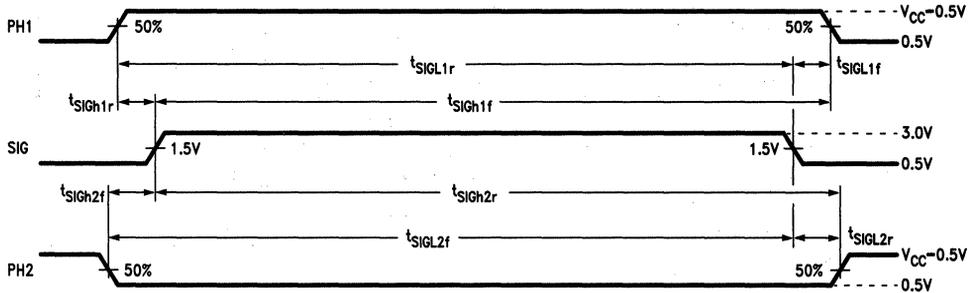
†To insure that valid source data is written to the FIFO, FWR must occur one PH1 cycle time after DLE.

††To ensure that valid data is read from the FIFO, there must be at least two PH1 periods between FIFO write and FIFO read cycles when the FIFO read and write counters are equal. Please consult additional criteria in applications section of this datasheet.

Timing Diagrams

DEFINITIONS

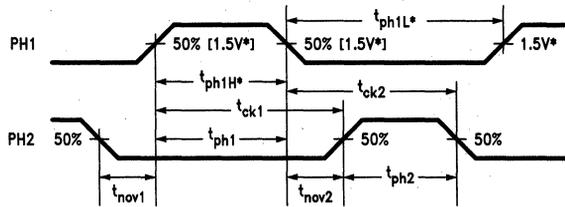
All the timing specifications given in this section refer to 50% of the leading or trailing edges of the appropriate clock phase and 0.5V or 3.0V on the appropriate signal as illustrated in the following figures, unless specifically stated otherwise.



TL/F/8672-16

Note: Data is measured at 1.5V to 50% points of PH1 and PH2.

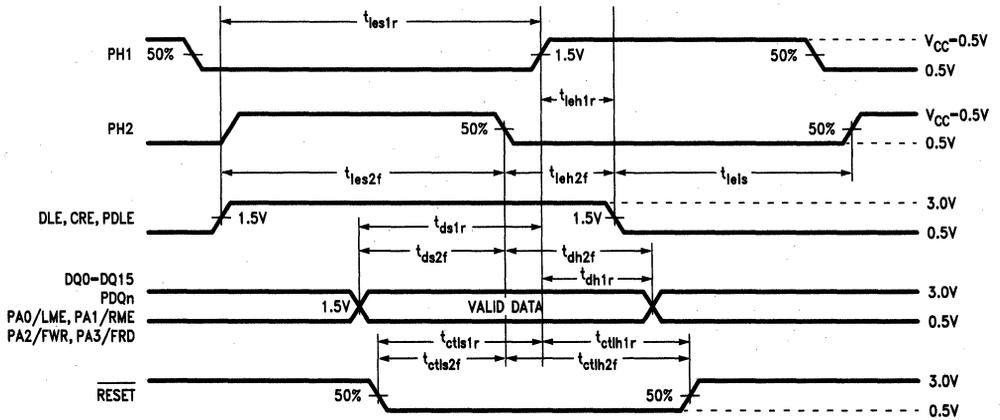
FIGURE 15. Timing Specification Standard



TL/F/8672-17

*TTL Clock Specification

FIGURE 16. Two-Phase MOS/Single-Phase TTL Clock Timing Specification



TL/F/8672-18

FIGURE 17. Input Data Timing Specification

Timing Diagrams (Continued)

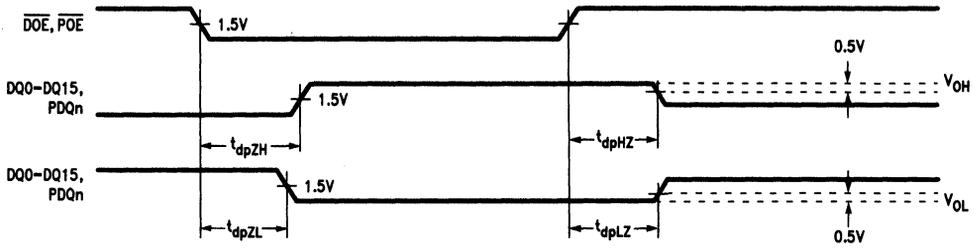


FIGURE 18. Output Enable Timing Specification

TL/F/8672-19

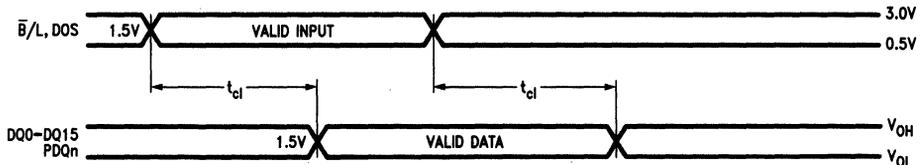


FIGURE 19. Mode Control Timing Specification

TL/F/8672-20

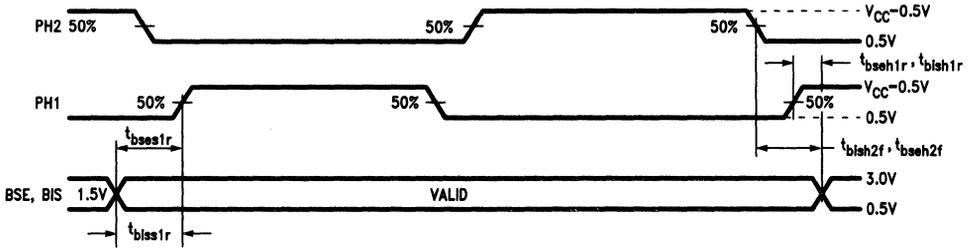


FIGURE 20. Control Signal Timing Specification

TL/F/8672-21

Timing Diagrams (Continued)

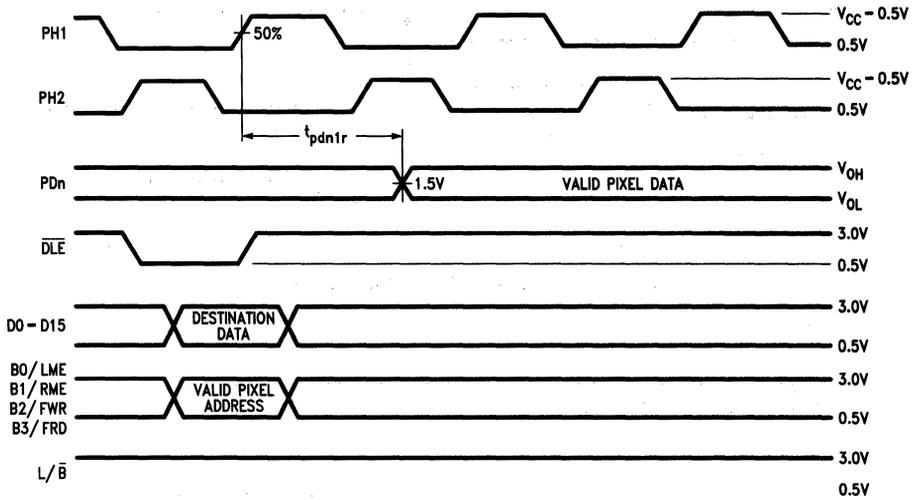


FIGURE 21. Pixel Read Timing Specification

TL/F/8672-23

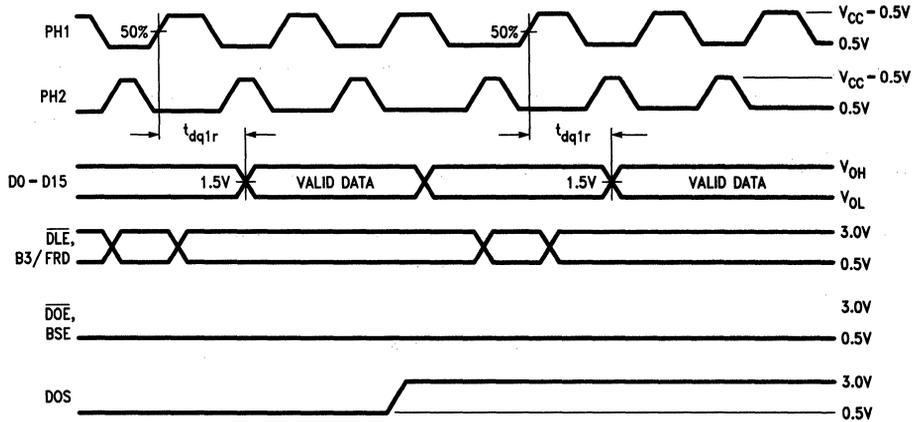
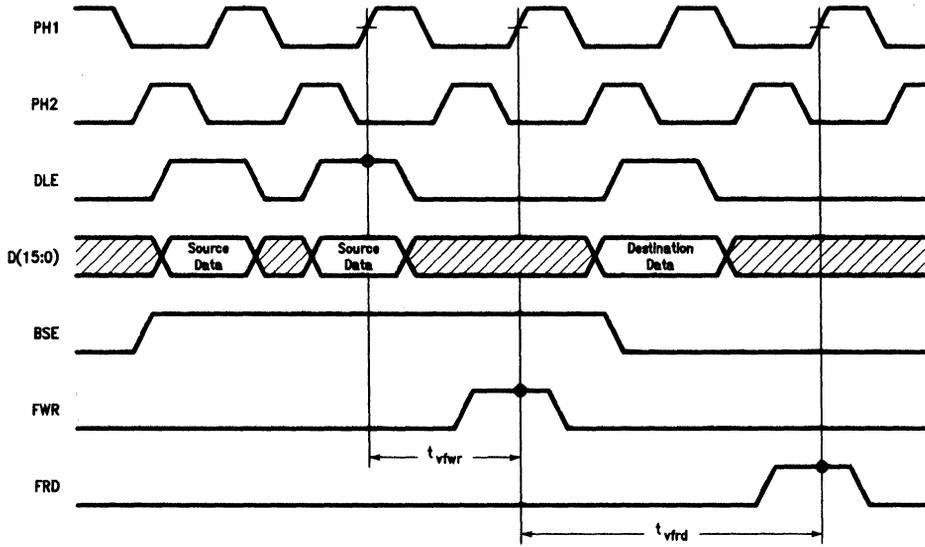


FIGURE 22. FIFO/DIL-Dest Data Read Timing Specification

TL/F/8672-24

Timing Diagrams (Continued)



TL/F/8672-28

FIGURE 23. DLE & FWR Timing Relationships



DP8511 BITBLT Processing Unit (BPU)

General Description

The DP8511 BITBLT Processing Unit (BPU), a member of National Semiconductor's Advanced Graphics Chip Set (AGCS), is a high performance microCMOS device intended for use in raster graphics applications. Specifically designed to complement the DP8500 Raster Graphics Processor (RGP), the BPU performs data operations that are elementary to BITBLT (BIT boundary Block Transfer) graphics: Shift, mask, and bitwise logical manipulation of memory. Under the control of the RGP, the BPU performs the necessary BITBLT data path operations at pipelined hardware speeds. A simple set of control lines interfaces the BPU to the RGP and to the system memory.

The BPU has two modes of operation: BITBLT and Line Drawing. BITBLT performs shift and logical operations on blocks of 16-bit data words. Line drawing performs similar operations on single-bit pixel data by utilizing a single bit pixel port (PDn). This port allows data read and read-modify-write operations on single pixels across a number of bit-planes, giving access to pixel depth. The BPU provides both pixel level processing commonly used in image processing applications and extremely fast planar operations used most frequently in color graphics.

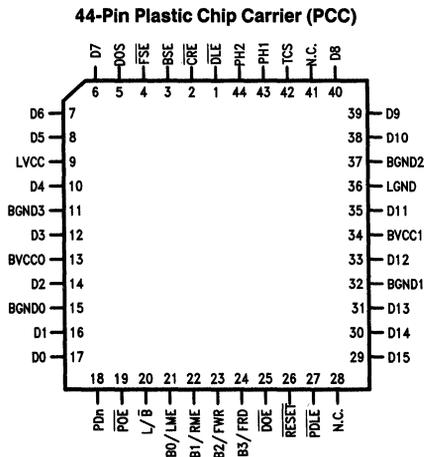
The BPU's operation is controlled by the values loaded to the Control Register (CR) and the Function Select Register (FSR). This dual register configuration of the DP8511 allows for high throughput in multi-plane systems that incorporate a BPU per plane. This performance advantage is achieved by allowing the flexibility of changing the FSR's contents inde-

pendent of the CR, so that multiple bitplanes can be updated simultaneously while each BPU performs different logical operations on its own destination data.

Features

- Interfaces directly to the DP8500 Raster Graphics Processor or any general purpose controller
- 20 MHz operation
- Supports all 16 classical BITBLT functions
- Pipelined data input for high system throughput
- Provides performance independent of the number of bitplanes
- Line Drawing support
- Compatible with static, dynamic RAMs, and Video RAMs
- Compatible with page mode, nibble mode and static column RAMs
- 32-bit to 16-bit barrel shifter
- 16-bit data port, single bit pixel port
- 16-word FIFO
- 16-bit logic operations
- Single +5V supply
- All inputs and outputs TTL compatible
- 2 micron microCMOS technology
- Packaged in a 44-pin PCC (commercial) or 44-pin PGA (MIL)

Connection Diagrams



TL/F/9337-1

N.C. = No Connection

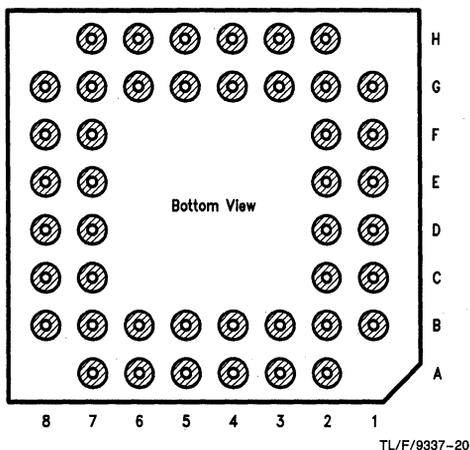
Top View

Order Number DP8511V
See NS Package Number V44A

Connection Diagrams (Continued)

44-Pin Grid Array (PGA) Package

Description	Pin	Description	Pin
DLE	A4	B2/FWR	H5
CRE	A5	B3/FRD	H4
BSE	B5	DOE	G4
FSE	A6	RESET	H3
DOS	B6	PDLE	G3
D7	A7	N.C.	H2
D6	B7	D15	G2
D5	C7	D14	F2
LVCC	B8	D13	G1
D4	C8	BGND1	F1
BGND3	D7	D12	E2
D3	D8	BVCC1	E1
BVCC0	E8	D11	D1
D2	E7	LGND	D2
BGND0	F8	BGND2	C1
D1	F7	D10	C2
D0	G8	D9	B1
PDQn	G7	D8	B2
POE	G6	N.C.	B3
L/B	H7	TCS	A2
B0/LME	H6	PH1	A3
B1/RME	G5	PH2	B4



Order Number DP8511U
See NS Package Number U44A

Pin Descriptions

INPUTS

- DOE:** Data Output Buffer Enable. When asserted low, this pin enables the output buffers on D0–D15. This input is generally driven by memory control logic.
- POE:** Pixel Output Buffer Enable. Enables the Pixel Output Buffer when active (low). This pins allows for reading pixel data from the frame buffer in a multiple bitplane system. This signal is generally provided by the memory control logic.
- L/B:** Line Drawing or BITBLT. A high on this input enables the BPU for line drawing mode. A low enables the BITBLT mode. This input is normally driven by the RGP.
- B0/LME:** Bit Select 0/Left Mask Enable. In the line drawing mode, this pin serves as the least significant pixel address bit. When in the BITBLT mode, this pin is used as the active high Left Mask Enable input. This input must be synchronized with respect to the falling edge of PH2. This pin is normally driven by the RGP.
- B1/RME:** Bit Select 1/Right Mask Enable. In the line drawing mode, this pin serves as the second least significant pixel address bit. When in the BITBLT mode, this pin is used as the active high Right Mask Enable input. This input must be synchronized with respect to the falling edge of PH2. This pin is normally driven by the RGP.

- B2/FWR:** Bit Select 2/FIFO Write Control. In the line drawing mode, this pin serves as the second most significant pixel address bit. When in the BITBLT mode, this pin is used as the active high FIFO Write control input. This input must be synchronized with respect to the falling edge of PH2. This pin is normally driven by the RGP.
- B3/FRD:** Bit Select 3/FIFO Read Control. In the line drawing mode, this pin serves as the most significant pixel address bit. When in the BITBLT mode, this pin is used as the active high FIFO Read input. This input must be synchronized with respect to the falling edge of PH2. This pin is normally driven by the RGP.
- BSE:** BITBLT Source Enable. This pin enables the BITBLT source input data path and controls the latching function of the BITBLT source pipeline register. BSE is sampled and latched on the falling edge of PH2 clock. The level on BSE must be asserted one clock cycle prior to loading data to the Data Input Latch (DIL), and must remain constant during the subsequent data load cycle to ensure data integrity. This input is driven by the RGP.
- TCS:** TTL Clock Select. This pin is tied to either V_{CC} or GND. A high level on this pin selects the TTL level clock input mode which uses a conventional TTL level clock of up to 17 MHz, simplifying

Pin Descriptions (Continued)

- TCS:** the system design. When using TTL clock mode all references to the falling edge of PH2 must be changed to the rising edge of PH1. A low level selects a 2 phase MOS clock mode which uses PH1 and PH2 as clock inputs.
- PH1:** Phase 1 clock input/TTL clock input. When TCS is low, this input serves as the Phase 1 MOS clock input with a maximum clock rate of 20 MHz. When TCS is high, this input is the TTL clock input with a maximum rate of 17 MHz.
- PH2:** Phase 2 clock input. This pin is the Phase 2 MOS clock input with a maximum rate of 20 MHz. When using a TTL clock, this input must be tied to ground.
- CRE:** BPU Control Register Enable. The data on D0–D15 is latched into the Control Register on the rising edge of CRE. This input is driven by memory decode logic.
- DLE:** Data Latch Enable. A low on this pin enables the BPU's Data Input Latch (DIL). The data on D0–D15 is latched into the DIL on the falling edge of the PH2 clock. DLE must be synchronized with respect to the valid data and must be removed before the rising edge of the subsequent PH2 clock. This input is generally driven by memory control logic.
- FSE:** BPU Function Select Register Enable. The data on D0–D15 is latched into the Function Select Register on the rising edge of FSE. This input generally is driven by memory decode logic.
- PDLE:** Pixel port Data Latch Enable. A low level on this pin enables the BPU's pixel port data input latch. The data on the PDn pin is latched into the one-bit Pixel Input Latch on the falling edge of the PH2 clock. PDLE must be synchronized with respect to the valid pixel data and must be removed before the rising edge of the subsequent PH2 clock.
- DOS:** Data Output Select: DOS selects the data output from either the FIFO (DOS = 1) or the BITBLT logic unit (DOS = 0). This input should be grounded when the BPU is controlled by an RGP.
- RESET:** FIFO control Reset. A low on this pin resets the BPU's FIFO read/write control circuitry. Data previously stored in the FIFO or on-chip latches is unchanged. This input must be synchronized with respect to the falling edge of PH2. PA2/FWR and PA3/FRD inputs must be low 1 clock cycle prior to and 2 clock cycles after asserting RESET. This pin is controlled by either the RGP's RSTO line or a TTL level reset signal.
- INPUTS/OUTPUTS**
- D0–D15:** 16-bit Bidirectional Data Port. This port serves as the input to the Data Input Latch, Control Register, and to the Function Select Register. When DOE is active (low), this port serves as a 16-bit output buffer.

- PDn:** Pixel Data Port, a single bit bidirectional port. This I/O port is used in the line drawing mode ($L/\bar{B} = 1$). It serves as the input to the Pixel Input Latch, providing a source bit for line drawing. When POE is active (low), this port serves as an output for the pixel previously latched in the Pixel Output Latch (POL).

SUPPLIES

- LVCC:** Positive supply for on-chip logic. $5 V_{DC} \pm 10\%$.
- LGND:** Ground for on-chip logic.
- BVCC0–**
- BVCC1:** Positive supply for output buffers, two pins total. $5 V_{DC} \pm 10\%$.
- BGND0–**
- BGND3:** Ground for output buffers, four pins total.

BITBLT Fundamentals

BITBLT, BIT-aligned Block Transfer, is a general operator that provides a mechanism to move an arbitrary size rectangle of an image from one part of the frame buffer to another. During the data transfer process a bitwise logical operation can be performed between the source and the destination data. BITBLT is also called RasterOp: operations on rasters. It defines two rectangular areas, source and destination, and performs a logical operation (e.g., AND, OR XOR) between these two areas and stores the result back to the destination. It can be expressed in simple notation as:

Destination ← Source op Destination.

op: AND, OR, XOR, etc.

FRAME BUFFER ARCHITECTURE

Generally, there are two kinds of frame buffer architectures: PLANE-oriented or PIXEL-oriented. BITBLT takes advantage of the plane-oriented frame buffer architecture's attribute of multiple, adjacent pixels-per-word, facilitating the movement of large blocks of data quickly in a frame buffer. However, the plane-oriented architecture has one inherent problem: the limit of resolution for memory addressing and access is the word, rather than the pixel. The BITBLT source starting address, the BITBLT destination starting address, the BITBLT width and the BITBLT height are all defined in pixels. The BITBLT source data block may start and end at any bit position of any word, and the destination data block also may start and end at any bit position of any word.

BIT ALIGNMENT

Before a logical operation can be performed between the source and the destination data, the source data must first be bit aligned to the destination data. In *Figure 1*, the source data need to be shifted three bits to the right in order to align the first pixel (that is, the pixel at the top left corner) in the source data block to the first pixel in the destination data block. For maximum performance, this alignment function must be implemented with a barrel shifter.

WORD BOUNDARIES AND DESTINATION MASKS

Each BITBLT destination scan line may start and end at any position in any data word. The neighboring bits (the bits sharing the same word address with any words in the destination data block, but not a part of the actual BITBLT rectangle) of the BITBLT destination scan line must remain un-

BITBLT Fundamentals (Continued)

changed after the BITBLT. Due to the plane-oriented frame buffer architecture, all memory operations must be word-aligned. In order to preserve the neighboring bits surrounding the BITBLT destination block, a left mask is needed for all the leftmost data words of the destination block, and a right mask is needed for all the rightmost data words of the destination data block. Both the left mask and the right mask remain the same throughout a given BITBLT operation.

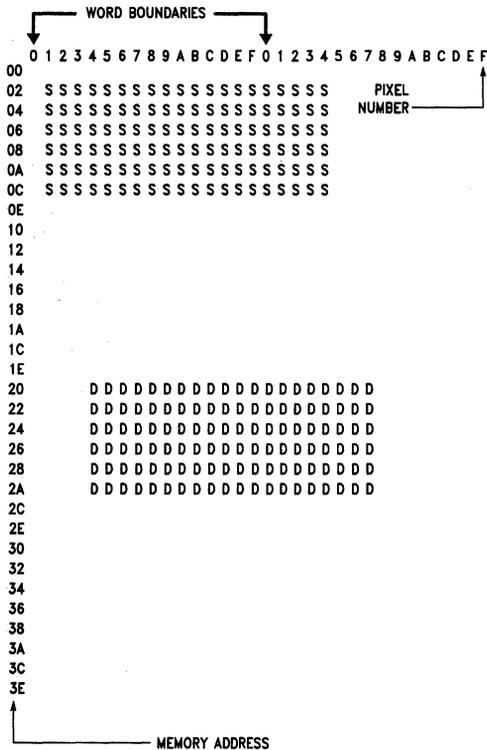


FIGURE 1. A 32 by 32 Frame Buffer

TL/F/9337-17

The following example illustrates the bit alignment requirement. In this example, the graphics controller has a 16-bit wide data bus. *Figure 1* shows a 32 pixel by 32 scan line frame buffer which is organized as a long bit stream which wraps around every two words (32 bits). Further, the top left corner of the frame buffer starts from the lowest word in the memory, address 000hex. Each word in the memory contains 16 bits, DQ0-DQ15. The most significant bit of a memory word, DQ0, is defined as the first *displayed* pixel in a word. In other words, memory's DQ0 to DQ15 correspond to pixels 0 to 15 respectively. In this example, BITBLT addresses are expressed in terms of pixel number, starting (with 0) from the upper-leftmost pixel. The BITBLT source starting address is set to 021hex (the second pixel in the third word). The BITBLT destination starting address is set

to 204hex (the fifth pixel in the 33rd word). The BITBLT width is set to 013hex (=19 decimal, corresponding to a width of 20 pixels). The BITBLT height is set to 005hex (=5 decimal, corresponding to 6 scan lines).

The left BITBLT mask for the above example is:

0000,1111,1111,1111

The right BITBLT mask for the above example is:

1111,1111,0000,0000

Note: Zeros in either the left mask or the right mask indicate the destination bits which will not be modified.

BITBLT DIRECTIONS

The BITBLT moves a rectangular block of data in a frame buffer. For a plane-oriented frame buffer, the BITBLT process can be considered a subroutine which has two nested loops. The loops are preceded by the BITBLT setup computations. The outer loop is the BITBLT source and destination scan line pixel starting address calculation and line count test for completion. The innermost loop is the actual BITBLT data movement for a single BITBLT scan line and word count test for completion. The length of the innermost loop is the word count of the BITBLT width. The length of the second loop is equal to the BITBLT's height (number of scan lines involved in a BITBLT):

```
BITBLT: calculate BITBLT setup parameters ;once per BITBLT
        such as
        width, height
        bit misalignment (shift number)
        left, right masks
        horizontal, vertical directions
        etc
```

```
OUTERLOOP: calculate source, dest addresses ;once per scanline
```

```
    INNERLOOP: move data and increment addresses ;once per word
    UNTIL      done horizontally
```

```
UNTIL      done vertically
```

```
RETURN     (from BITBLT).
```

Each loop can be executed in one of two directions: the inner loop from left to right or right to left, the outer loop from top to bottom (down) or bottom to top (up).

The ability to move data starting from any corner of the BITBLT rectangle is necessary to avoid destroying the BITBLT source data as a result of destination writes when the source and destination are overlapped (i.e., when they share pixels). This situation is routinely encountered while panning or scrolling.

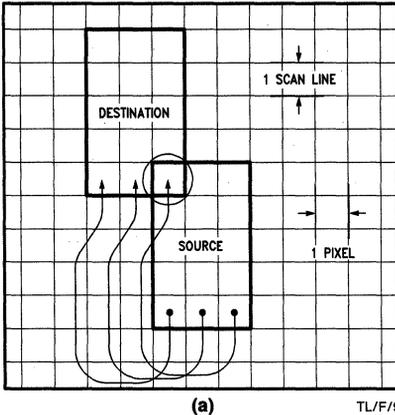
A determination of the correct execution directions of the BITBLT must be performed whenever the source and destination rectangles overlap. *Any* overlap will result in the destruction of source data (from a destination write) if the correct vertical direction is not used. Horizontal BITBLT direction is of concern only in certain cases of overlap, as will be explained below.

Figure 2 (a) and (b) illustrate two cases of overlap. Here, the BITBLT rectangles are three pixels wide by five scan lines high; they overlap by a single pixel in (a) and a single col-

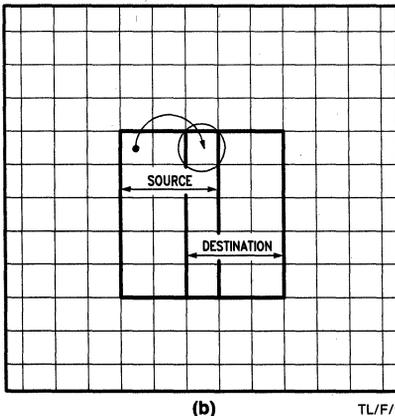
BITBLT Fundamentals (Continued)

umn of pixels in (b). For purposes of illustration, the BITBLT is assumed to be carried out pixel-by-pixel. This convention does not affect the conclusions.

In Figure 2(a), if the BITBLT is performed in the UP direction (bottom-to-top) one of the transfers of the bottom scan line of the source will write to the circled pixel of the destination. Due to the overlap, this pixel is also part of the uppermost scan line of the source rectangle. Thus, data needed later is destroyed. Therefore, this BITBLT must be performed in the DOWN direction. Another example of this occurs any time the screen is moved in a purely vertical direction, as in scrolling text. It should be noted that, in both of these cases, the choice of horizontal BITBLT direction may be made arbitrarily.



TL/F/9337-18



TL/F/9337-19

FIGURE 2. Overlapping BITBLT Operations

Figure 2(b) demonstrates a case in which the horizontal BITBLT direction may not be chosen arbitrarily. This is an instance of purely horizontal movement of data (panning). Because the movement from source to destination involves data within the same scan line, the incorrect direction of movement will overwrite data which will be needed later. In this example, the correct direction is from right to left.

BITBLT VARIATIONS

Some implementations of BITBLT are defined in terms of three operands: source, destination and mask/texture. This

third operand is commonly used in monochrome systems to incorporate a stipple pattern into an area. These stipple patterns provide the appearance of multiple shades of gray in single-bit-per-pixel systems, in a manner similar to the 'half-tone' process used in printing.

Destination ← Texture op1 Source op2 Destination

While the BPU is essentially a two-operand device, three-operand BITBLT can be implemented quite flexibly and efficiently by performing the two operations serially. The BPU permits the use of any of its sixteen operations for each of the two operators shown above as 'op1' and 'op2'. Additionally, the on-chip FIFO can be used to store the intermediate result (the result of op1 later used as an operand of op2) thus minimizing the number of memory accesses required.

ENHANCING BITBLT PERFORMANCE

There are various ways to enhance BITBLT performance (speed). The simplest way is to try to get data in and out of the memory system quickly. Most of the bitmapped graphics systems utilize DRAMs for both cost and storage density reasons. Since the BITBLT data shows strong locality, the graphics system can take advantage of certain fast memory access modes available to the DRAMs, such as page mode access, static column access, etc. The BPU, by means of an internal FIFO, can pipe the BITBLT source data to reduce the frequency of switching out of the current page address space, thus maximizing the ability of the system to capitalize on the data's locality. This operation is described in the following section.

PIPING THE BITBLT SOURCE DATA

When the BITBLT width is more than a word, up to 16 source data words can be piped into the BPU's on-chip FIFO. At the end of each BITBLT scan line or at the end of 16 source data words, the controller switches from the BITBLT source address space to the BITBLT destination address space. When the BITBLT destination data word is fetched, two possible memory control sequences can be used. One is the modify-write sequence: write the BITBLT result back to the destination memory immediately after the logical function is executed. The second sequence involves storing the BITBLT result back to the BPU's on-chip FIFO, to a maximum of 16 words. Either at the end of each BITBLT destination scan line or at the 16th destination data word, the BITBLT resultant data is then read out from the FIFO and written to the BITBLT destination memory sequentially.

Summary of the BITBLT Memory Control Sequences

BITBLT MEMORY SEQUENCE I

- 1) Load the BPU Control Register with [BIS, SN, LM, RM], via the data bus.
- 2) Load the BFU FS register with function select code.
- 3) Read in the BITBLT source data up to 16 words (17 words in certain cases), barrel-shift, then write them into the on-chip FIFO. (Only 16 barrel-shifted data words can be stored.)
- 4) Read in the BITBLT destination data while the barrel-shifted source data is read out from the on-chip FIFO and the selected logical operation is executed, then write back to destination.

Summary of the BITBLT Memory Control Sequences (Continued)

- 5) Go to step 3 until the end of the BITBLT scan line.
- 6) Go to step 3 for the remaining BITBLT scan lines.

BITBLT MEMORY SEQUENCE II

- 1) Load the BPU Control Register with [BIS, SN, LM, RM] via the data bus.
- 2) Load BPU FS register with function select code.
- 3) Read in the BITBLT source data up to 16 words (17 words in certain cases), barrel-shift, then write them into the on-chip FIFO. (Only 16 barrel-shifted data words can be stored.)
- 4) Read in the BITBLT destination data in sequence, execute the selected logical operation and then write the result back to the on-chip FIFO, maximum 16 BITBLT data words.
- 5) Send BITBLT result from the FIFO and write them back to the BITBLT destination memory.
- 6) Go to step 3 until the end of the BITBLT scan line.
- 7) Go to step 3 for the remaining BITBLT scan lines.

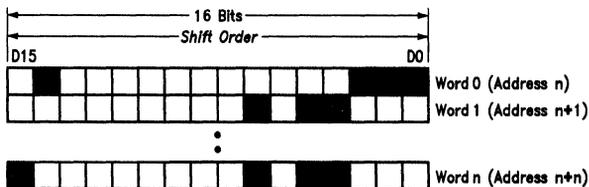
DP8511 BIT ORDER

The DP8511's bit order was defined so it reflects the sequence in which pixel data is viewed in a downward facing

Cartesian Coordinate display system (see *Figure 3b*). In this type of display configuration, the pixel address increases as one scans from left-to-right as well as from the top to the bottom of the display. When addressing individual pixels within a given 16-bit word, a 4-bit pixel address value is assigned to each pixel, or bit, of the word and the address value increases from the leftmost pixel in the word to the right. This allows for the formation of a uniform pixel address space simply by concatenating the pixel address (4 bits) to the LSB of the word address.

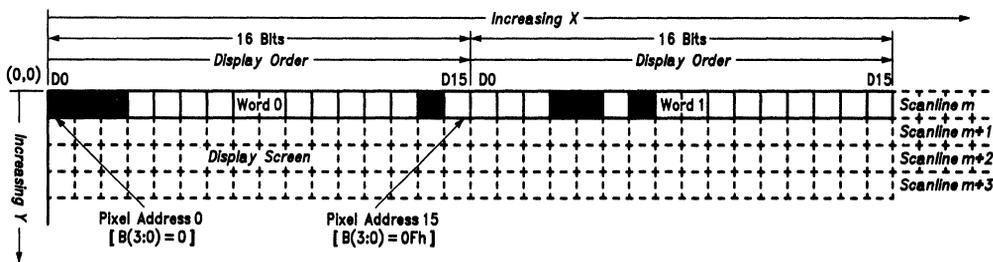
The DP8511's internal word bit-order is defined to be consistent with the order in which pixel data is displayed on the screen (i.e., from D0 on the left to D15 on the right). This convention provides logical continuity when observing how data is manipulated inside the DP8511 and establishes the relationship to the display pixel addressing scheme. The pixel address bits B0-B3 map directly to this left-to-right bit order convention.

It should be noted that when data is viewed in the frame buffer the order of the pixel bits will be consistent with most conventional microprocessor conventions that define D0 in the LSB bit position or in the rightmost bit location (see *Figure 3a*).



TL/F/9337-21

a) Data Representation in Frame Buffer (as Seen by the CPU)



TL/F/9337-22

b) Data Representation in DP8511-Based Display (Downward Facing Cartesian Coordinate System)

FIGURE 3. Frame Buffer vs Display Coordinate Bit-Order Representation

Functional Block Description

BPU CONTROL REGISTER

The BPU Control Register (CR) is a 13-bit register consisting of three 4-bit fields and one single-bit field. The Right Mask field, RM, controls the BITBLT right mask pattern. The Left Mask, LM, controls the BITBLT left mask pattern. The barrel Shift Number, SN, determines the number of bit positions shifted by the Barrel Shifter. The BIS bit controls the organization of two 16-bit words at the input to the barrel shifter.

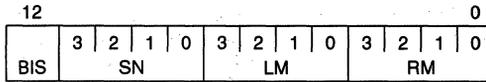


FIGURE 4. Control Register Fields

The CR is loaded by presenting the appropriate data at the Data Port (D0–D15) and then asserting a low level on the CRE (Control Register Enable) input. Data is latched into the register on the rising edge of CRE.

When the DP8511 is controlled by the RGP, the CR is treated as a memory mapped write only register and resides in the RGP's memory space. The address of this register is determined by the RGP's BPUB register (more information can be found in the RGP Programmer's Reference Manual). The CR Register contents are not affected when the BPU is in the line drawing mode ($L/\bar{B} = 1$).

Left and Right Masks

The Left and Right mask fields are used to determine which portion of the destination data word to preserve during a BITBLT operation. Masks are used when a destination starting or ending address falls at a pixel address that is not on a word boundary. Since data is accessed and modified in 16-bit word quantities, neighboring bits on both sides of the destination data block must be protected or they risk being corrupted when the destination data is logically combined with the source data. The masks isolate the logic operation to only those bits that are enabled with a one in the mask fields (of course, RME and/or LME must be asserted concurrently). The truth table for the left and right mask patterns is shown in Table I.

The RM and LM values are automatically determined by the RGP during BITBLT and are based on the starting address of the destination data block specified in the DSAD register.

Shift Number

There are 16 different barrel shift operations controlled by the SN of the Control Register. Let $A = a_{15} \dots a_0$, $B =$

$b_{15} \dots b_0$ and $C = c_{15} \dots c_0$, A and B are the input words, C is the output. An i-bit barrel-shift operation on A and B denoted by $C = S(i; A, B)$ is the operation of concatenating the least significant (l) bits of word B to the most significant (16-i) bits of word A.

When $A = B$, the $C = S(i; A, B)$ is equivalent to having an i-bit circular left shift. The i-bit circular right shift is equivalent to the (16-i) bit circular left shift.

The truth table of the barrel-shift operation $S(i, A, B)$ is shown in Table III. The shift number is represented by "i", A0–A15 corresponds to the most significant input word of the barrel shifter, and B0–B15 corresponds to the least significant input word of the barrel shifter. C0–C15 is the barrel shifter output.

Barrel Input Select (BIS)

The Barrel Input Select bit is used to route two 16-bit words, one in the Barrel Input Latch (BIL) and the other in the Data Input Latch (DIL), to either the most significant or the least significant word position of the Barrel Shifter. When the BIS bit is set high, the DIL source serves as the most significant input word to the barrel shifter with BIL data going to the least significant word position. When the BIS bit is set low, the DIL source input is routed to the least significant position of the barrel shifter and BIL data goes to the most significant position.

The criteria for setting the BIS bit can be determined by the direction of the BITBLT source data read operation. When transferring data words from memory to the BPU in a left-to-right sequence, the BIS bit should be set to zero so that the first source word fetched into the source pipe will be routed to the most significant word position of the barrel shifter while the second will go into the least significant position. In the case where data is being fetched in a right-to-left fashion, it becomes necessary to swap the two source words at the input to the barrel shifter so the bit order of the two words will remain contiguous. This is done by setting the BIS bit high.

This bit is automatically set by the RGP during BITBLT operations.

FUNCTION SELECT REGISTER

The Function Select Register (FSR) is a 4-bit register used to set one of 16 BITBLT logic operations to be performed between the source and a destination data blocks. The function select truth table can be found in Table II.

Functional Block Description (Continued)

TABLE I. Left and Right Mask Truth Table

		D0	D15
LM = 0	L_mask =	1111	11111111 1111
LM = 1	L_mask =	0111	11111111 1111
LM = 2	L_mask =	0011	11111111 1111
LM = 3	L_mask =	0001	11111111 1111
LM = 4	L_mask =	0000	11111111 1111
LM = 5	L_mask =	0000	01111111 1111
LM = 6	L_mask =	0000	00111111 1111
LM = 7	L_mask =	0000	00011111 1111
LM = 8	L_mask =	0000	00001111 1111
LM = 9	L_mask =	0000	00000111 1111
LM = 10	L_mask =	0000	00000011 1111
LM = 11	L_mask =	0000	00000001 1111
LM = 12	L_mask =	0000	00000000 1111
LM = 13	L_mask =	0000	00000000 0111
LM = 14	L_mask =	0000	00000000 0011
LM = 15	L_mask =	0000	00000000 0001
RM = 0	R_mask =	1000	00000000 0000
RM = 1	R_mask =	1100	00000000 0000
RM = 2	R_mask =	1110	00000000 0000
RM = 3	R_mask =	1111	00000000 0000
RM = 4	R_mask =	1111	10000000 0000
RM = 5	R_mask =	1111	11000000 0000
RM = 6	R_mask =	1111	11100000 0000
RM = 7	R_mask =	1111	11110000 0000
RM = 8	R_mask =	1111	11111000 0000
RM = 9	R_mask =	1111	11111100 0000
RM = 10	R_mask =	1111	11111110 0000
RM = 11	R_mask =	1111	11111111 0000
RM = 12	R_mask =	1111	11111111 1000
RM = 13	R_mask =	1111	11111111 1100
RM = 14	R_mask =	1111	11111111 1110
RM = 15	R_mask =	1111	11111111 1111

“1” = Enable Logic Op
“0” = Disable Logic Op

The 16 BITBLT functions can be expressed as:

$$\begin{aligned} & f3 * (-s * -d) + \\ & f2 * (-s * d) + \\ & f1 * (s * -d) + \\ & f0 * (s * d) \end{aligned}$$

where f3–f0 are the bits of the Function Select Registers, 's' is the source data and 'd' is the destination data.

The FSR is loaded by asserting the Function Select Enable (FSE) pin low while valid data is made available at the data port (D0–D15). Data is latched into the register on the rising edge of FSE.

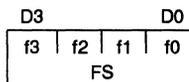
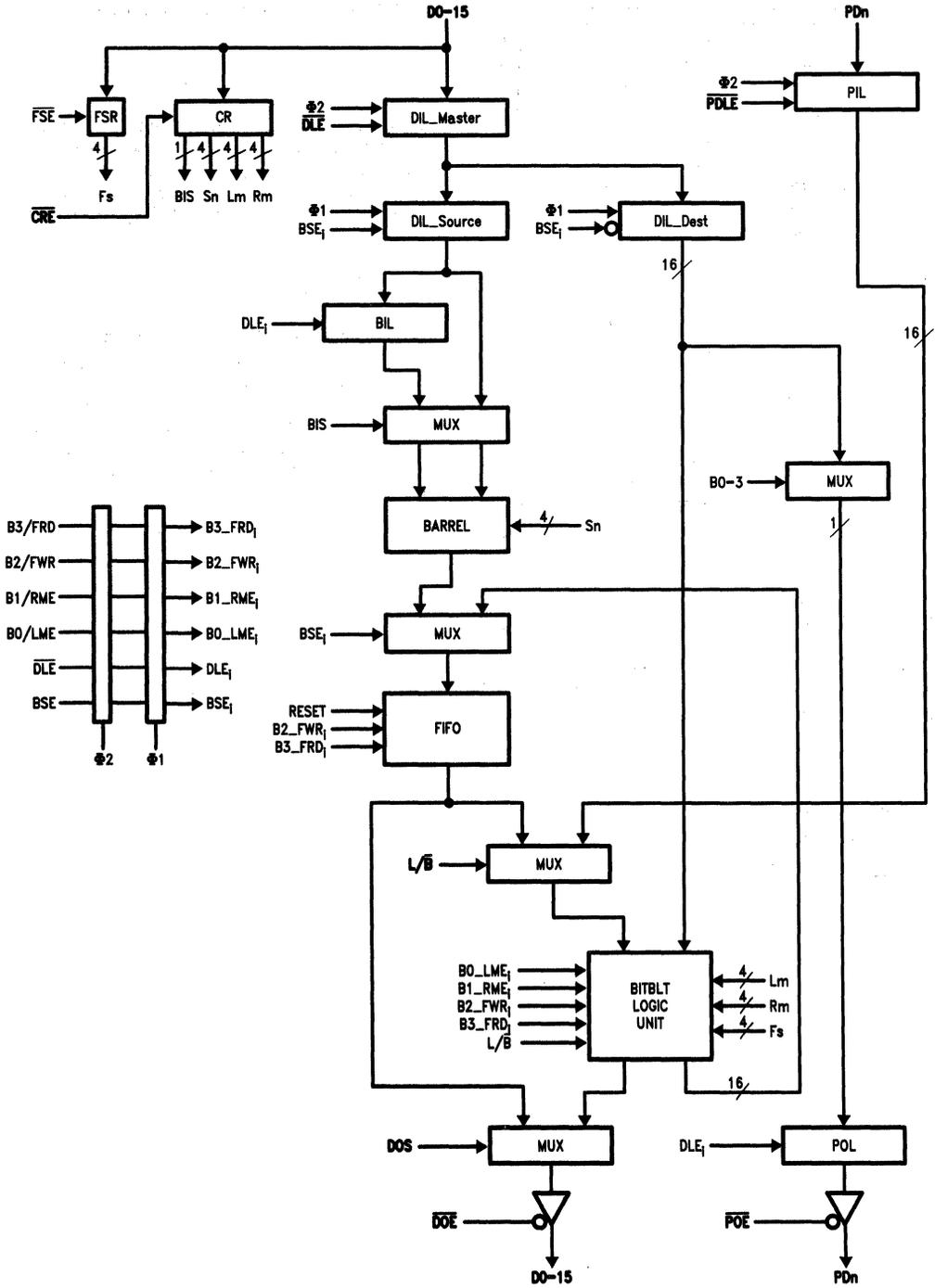


FIGURE 5. Function Select Register (FSR)

The contents of the FSR are defined by the user prior to BPU operations. In multi-plane systems that use a BPU per plane, each BPU can have its logic function programmed independently. This feature allows the RGP to BITBLT data to all bitplanes concurrently while maintaining the flexibility of having unique logical operations being performed on each plane.

Block Diagram



Functional Block Description (Continued)

TABLE II. BITBIT Function Definitions

Function Select Bits				Operation Performed
f3	f2	f1	f0	
0	0	0	0	0
0	0	0	1	s and d
0	0	1	0	s and -d
0	0	1	1	s
0	1	0	0	-s and d
0	1	0	1	d
0	1	1	0	s xor d
0	1	1	1	s or d
1	0	0	0	-s and -d
1	0	0	1	s xor d
1	0	1	0	-d
1	0	1	1	s or -d
1	1	0	0	-s
1	1	0	1	-s or d
1	1	1	0	-s or -d
1	1	1	1	1

Note:
d: destination data
s: source data
f3-f0: function select code, which selects one of the 16 BITBLT functions.

DATA INPUT LATCH

The Data Input Latch (DIL) consists of a master latch, DIL-Master, and two slave latches, DIL-Source and DIL-Dest (see Figure 6).

Data is latched into the DIL-Master latch on the falling edge of PH2 when DLE is asserted low. Data is then transferred into either the DIL-Source slave latch or the DIL-Destination slave latch on the next rising edge of PH1 depending on the state of the BSE input. Data is then latched when PH1 returns low.

Each of the slave latches is used to direct the flow of data to either the source or destination pipe. When performing BITBLT, data from the source area is read and latched into the source pipe via the DIL-Source latch, aligned and stored in the FIFO. Once the source data operation is complete, data from the destination area is then directed into the destination pipe via the DIL-Dest latch to be combined, word-by-word, with the data stored in the source pipe FIFO.

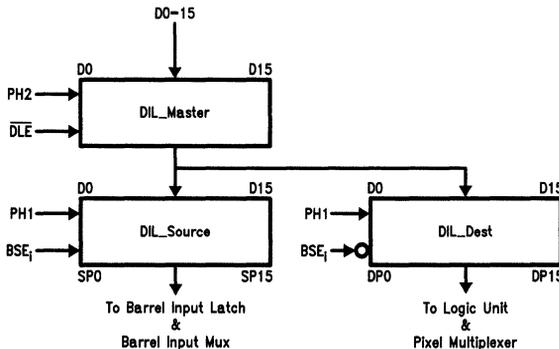


FIGURE 6. Data Input Latch Block Diagram

BARREL INPUT LATCH

The Barrel Input Latch (BIL) serves as a pipeline register for source data (see Figure 7) during a BITBLT source read cycle. Data, which is transferred into the DIL-Source register on the rising edge of PH1, is subsequently transferred into the BIL on the next rising edge of PH1. As a result, if two words are fetched and latched into the BPU on two subsequent clock cycles (assuming BSE = 1), the first data word will be in the BIL and the second in the DIL-Source Latch. The BIL and DIL-Source outputs are then concatenated via a crossbar multiplexer to form the 32-bit input to the Barrel Shifter.

It is important to note that valid data will remain in the BIL for only ONE clock cycle as the next PH1 cycle will again transfer data from the DIL-Source Latch to the BIL. This condition implies that the result from the Barrel Shifter must be latched into the FIFO on the rising edge of PH1 immediately following the PH1 transition used to latch the second word into the DIL-Source latch.

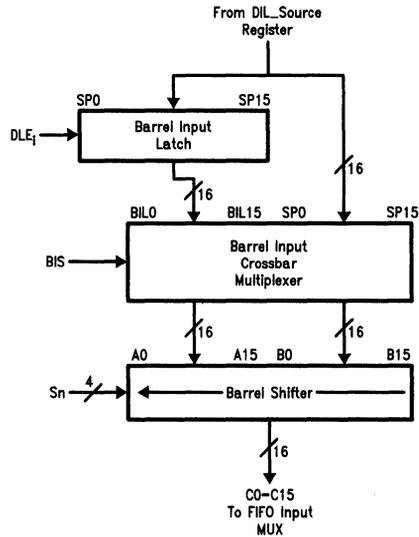


FIGURE 7. Barrel Shifter/Barrel Input Latch Block Diagram

TL/F/9337-23

TL/F/9337-24

Functional Block Description (Continued)

BARREL SHIFTER

The function of the Barrel Shifter is to align the BITBLT source data to the destination data. Bit alignment may cross word boundaries.

The Barrel Shifter is implemented as a 32-to-16-bit multiplexer (see *Figure 7*). Depending on the type and length of a BITBLT operation the necessary source data words are fetched from memory into the BPU to form a 32-bit input to the Barrel Shifter. The Barrel Input Latch (BIL) stores the first source data word fetched and the Source Data Input Latch (DIL-Source) stores the subsequent source data word.

A multiplexer precedes the Barrel Shifter to facilitate swapping the input words if necessary. This word swapping is controlled by the state of the BIS bit in the Control Register and does not affect the ordering of the data bits of the input words. This swap mechanism facilitates the fetching of BITBLT data from left-to-right or from right-to-left.

FIFO

The FIFO, 16 bits by 16 words, constitutes the BPU's on chip storage (see *Figure 8*). Implemented as a dual port register file, the FIFO has separate READ/WRITE control inputs, FIFO Read (FRD) and FIFO Write (FWR). The registers are selected via two shift register ring counters which

are incremented via the FRD and FWR inputs. These counters are set to zero with the RESET pin.

The FIFO facilitates the use of fast localized memory access modes such as page and static column modes by providing storage for multiple words of barrel shifted words.

BITBLT LOGIC CONTROL UNIT

The BITBLT Logic Control Unit (LCU) is responsible for the mask and bitwise logical operations performed on BITBLT and line drawing data (see *Figure 9*).

In the BITBLT mode ($L/\bar{B} = 0$), the LCU performs a bitwise logical operation between a source data word read from the FIFO and the Destination Data Input Latch contents. The logic operation performed is specified in the FS Register. The LCU is also responsible for masking the appropriate destination data bits from change during the logic operation. The masking function is determined by the values set in the LM and RM fields of the Control Register and the state of both LME and RME control inputs.

During Linedrawing ($L/\bar{B} = 1$), the LCU performs a bitwise logical operation between the contents of the Pixel Input Latch and a single bit of the Destination Data Input Latch contents. This bit is selected by the state of Pixel Address bits B0-B3. The logical operation function is determined by the FS Register contents.

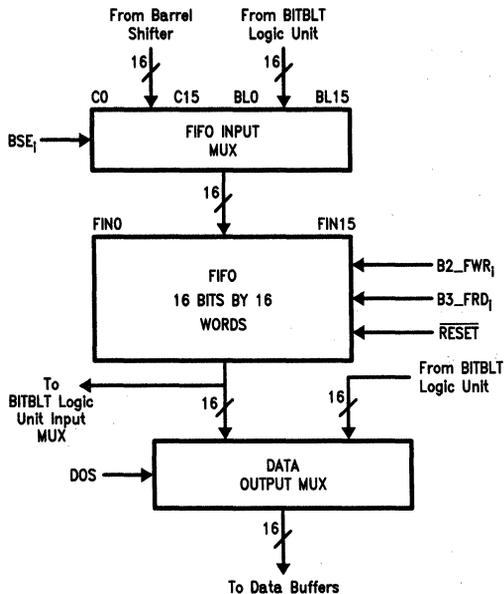


FIGURE 8. FIFO Block Diagram

TL/F/9337-25

Functional Block Description (Continued)

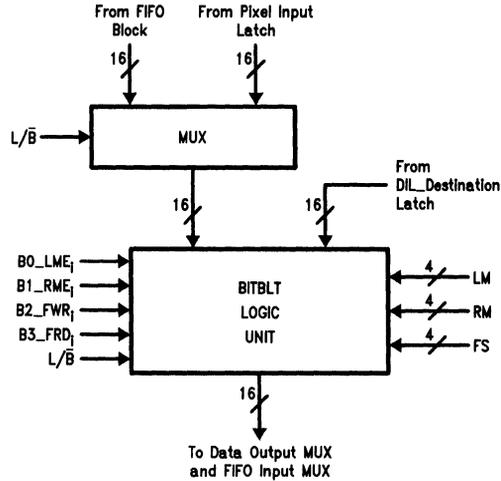


FIGURE 9. BITBLT Logic Unit Block Diagram

TL/F/9337-26

TABLE III. Barrel Shifter Truth Table*

i	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15
0	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15
1	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0
2	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1
3	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2
4	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3
5	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4
6	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5
7	a7	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6
8	a8	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7
9	a9	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8
10	a10	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9
11	a11	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
12	a12	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11
13	a13	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12
14	a14	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
15	a15	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14

*i = shift number

A0-A15 = barrel shifter most significant word

B0-B15 = barrel shifter least significant word

C0-C15 = barrel shifter output

Functional Block Description

(Continued)

PIXEL PORT

The Pixel Port consists of a single-bit I/O port dedicated to pixel level operations. It consists of a Pixel Input Latch and a Pixel Output Latch (see *Figure 10*). Data is written to the Pixel Input Latch by asserting \overline{PDLE} . Data is read from the Pixel Output Latch by asserting \overline{POE} .

The Pixel Input Latch (PIL) is used to store a pixel value which is used as the source operand during linedrawing Read-Modify-Write cycles. Internally this bit is replicated into a 16-bit word and routed to the BITBLT Logic Unit where it is masked and logically combined with a word read from the destination space. The Pixel Address inputs (B0-3) determine which bit of the destination word will be combined with the pixel port data. The PIL must be loaded prior to performing any linedrawing or pointwise drawing operations and L/B must be set high for proper operation.

The Pixel Output Latch (POL) is used to read a single-bit pixel from the frame buffer. This is accomplished by loading the appropriate destination word to the DIL-Dest latch (BSE must be asserted low while using this function). A single bit of this word is selected based on the value of the Pixel Address inputs (B0-3) and routed to the Pixel Output Latch. This bit is unmodified when read from the frame buffer.

Operational Description

The BPU's primary function is to perform two types of data manipulation tasks: 1) execute shift, mask, and logic operations between source and destination data blocks (BITBLT operation); 2) logically combine line drawing data with destination data (Line Drawing operations).

These operations are carried out in the BPU via 3 major functional blocks: The Barrel Shifter, the FIFO, and the BITBLT logic Unit. These blocks in combination with a variety of latches and multiplexers synchronize, modify and route the data through and around the device.

Dataflow through the BPU is dictated by control parameters previously set up in the Control and Function Select Registers and by the state of various control inputs on the device.

BITBLT OPERATION

The mechanics of a BITBLT operation are relatively straightforward. Following *Figure 11* and *Figure 12*, a source data word is fetched from memory and latched into the Data Input Latch (DIL-Source) via the DIL-Master latch. A second source data word is subsequently fetched and latched into the DIL-Source latch while the first source data word is simultaneously transferred to the Barrel Input Latch. This makes both the first and second source data words available to the Barrel Shifter as a 32-bit data word. This 32-bit quantity is routed through a crossbar mux and into the Barrel Shifter which produces an aligned 16-bit source word. This data is latched into the FIFO on the next clock cycle.

The data is then read from the FIFO on a subsequent clock cycle and routed to the BITBLT Logic Unit where it is masked and logically combined with a destination data word previously fetched and latched into the DIL-Destination latch. The data is then routed back to the Data Port (D0-D15) so it can be written back to the destination data memory.

LINE DRAWING OPERATION

During a line drawing operation (see *Figures 13* and *14*), the 'source' data, referred to as the 'Pixel Input Data', actually comes from the Pixel Input Latch (PIL). The PIL input data is latched into the BPU's PIL prior to the line-drawing operation. The destination data is then fetched from memory and written to the DIL-Destination latch (BSE must be asserted low). The pixel input data bit is internally replicated into all 16-bit positions of a data word and routed to the BITBLT Logic Unit where it is logically combined with the value in the DIL-Destination. Since only one bit of the destination word is to be modified with pixel data, a mask is

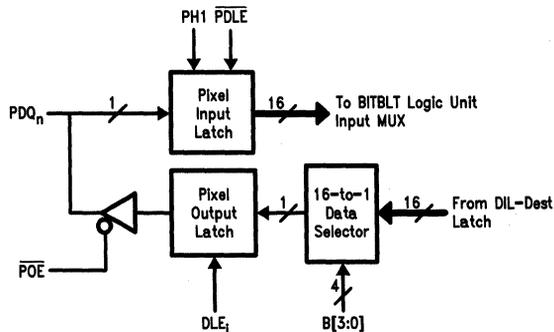


FIGURE 10. Pixel Port Block Diagram

TL/F/9337-27

Operational Description (Continued)

generated based on the state of the pixel address lines (B0–B3) so that all other bits of the destination word are protected from change. The resulting data word is then routed back to the Data Port (D0–D15).

In multiple-plane color display systems that utilize a BPU per bitplane, the Pixel Input Latch is commonly used to determine color value during linedrawing. This is accomplished by loading the color value across the depth of the planes. This technique is also useful for rapidly downloading image data that is stored in a word-per-pixel format into a frame buffer utilizing a planar architecture.

The Pixel Output Latch reflects a single pixel data value derived from a 16-bit word previously loaded to the DIL-Destination latch. The Output Pixel Data value is one-of-16 bits of the destination data word selected by the pixel address lines B0–B3 and is purely unmodified destination data. This function is useful for reading pixel color data in multiple-plane systems that support image processing capabilities.

TABLE IV. DP8500 RGP Drawing Instructions that utilize the DP8511 BPU

INITB	Asserts RST0 (resets FIFO pointers)
BTulsd	Block Transfer
DCbuld	Draw Character
DRLN[A]	Draw Line
DRLNS	Draw Line Steps
DRPGN[A]	Draw Polygon
DRPLN[A]	Draw Polygonal Line
DRPT	Draw Point
FILLAd	Fill Polygon
FILLTd	Fill Trapezoid
MOV s,FSE	Load Function Select Register (FSE = Address of FSE Register)
MOV s,PIL	Load Pixel Input Latch (PIL = Address of Pixel I/O Port)
MOV POL,d	Read Pixel Output Latch (POL = Address of Pixel I/O Port)
RDPT	Read Point

Operational Description (Continued)

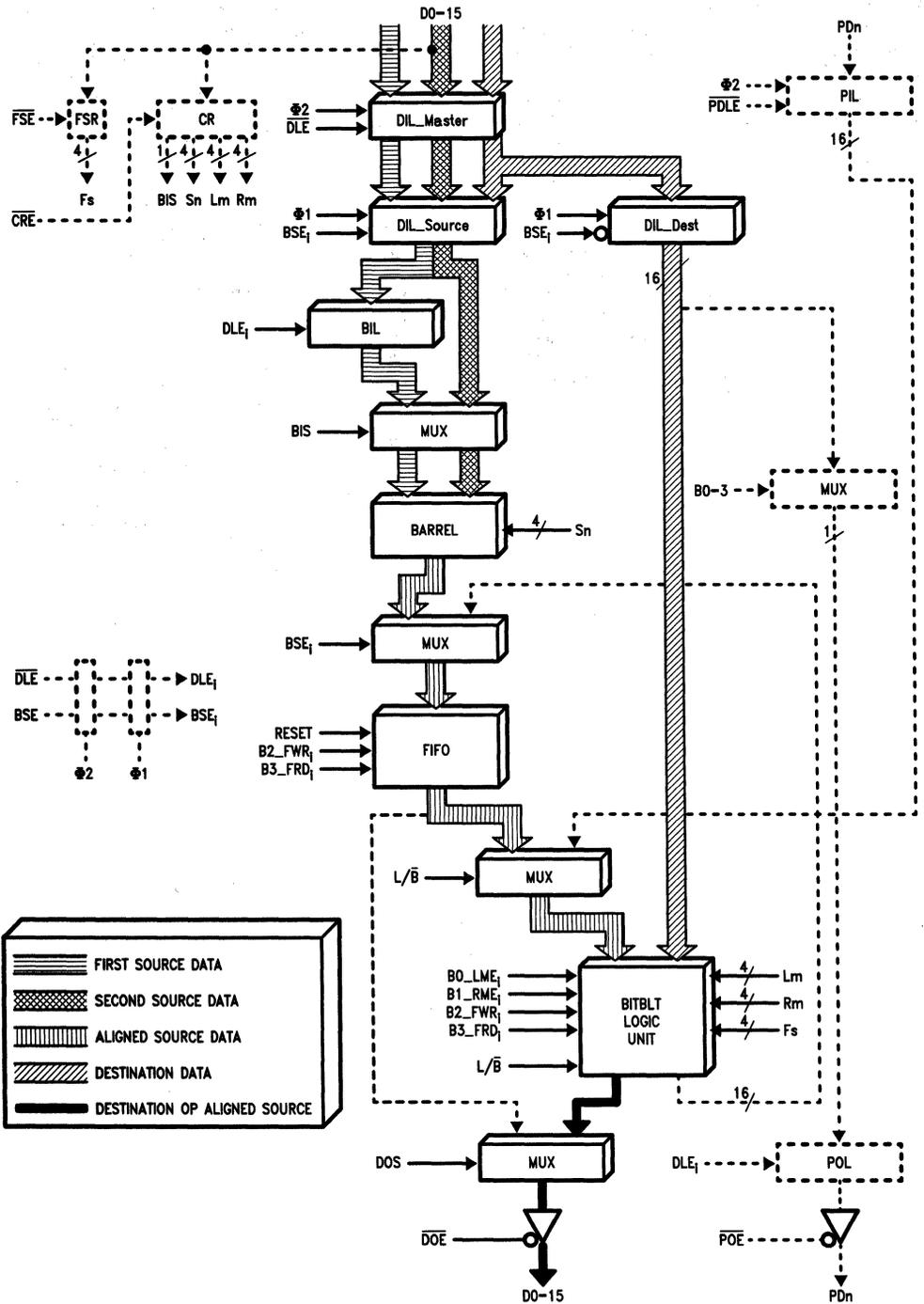


FIGURE 11. Dataflow during a Typical BITBLT Operation. (RGP Instructions: BT, FILLA, DC.)

TL/F/9337-3

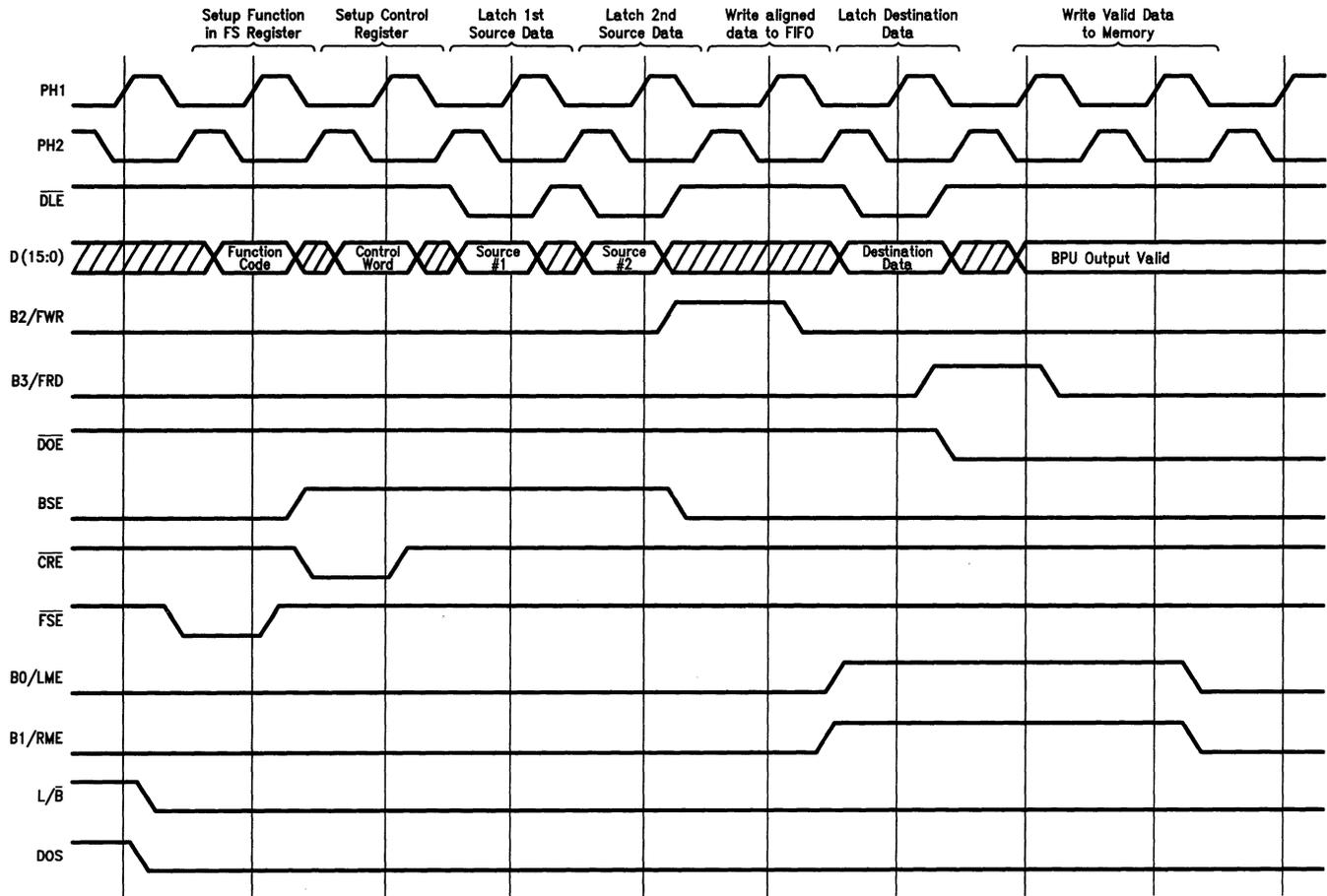


FIGURE 12. Typical Single BITBLT Source Read/Destination Read-Modify-Write Cycle

TL/F/9337-28

Operational Description (Continued)

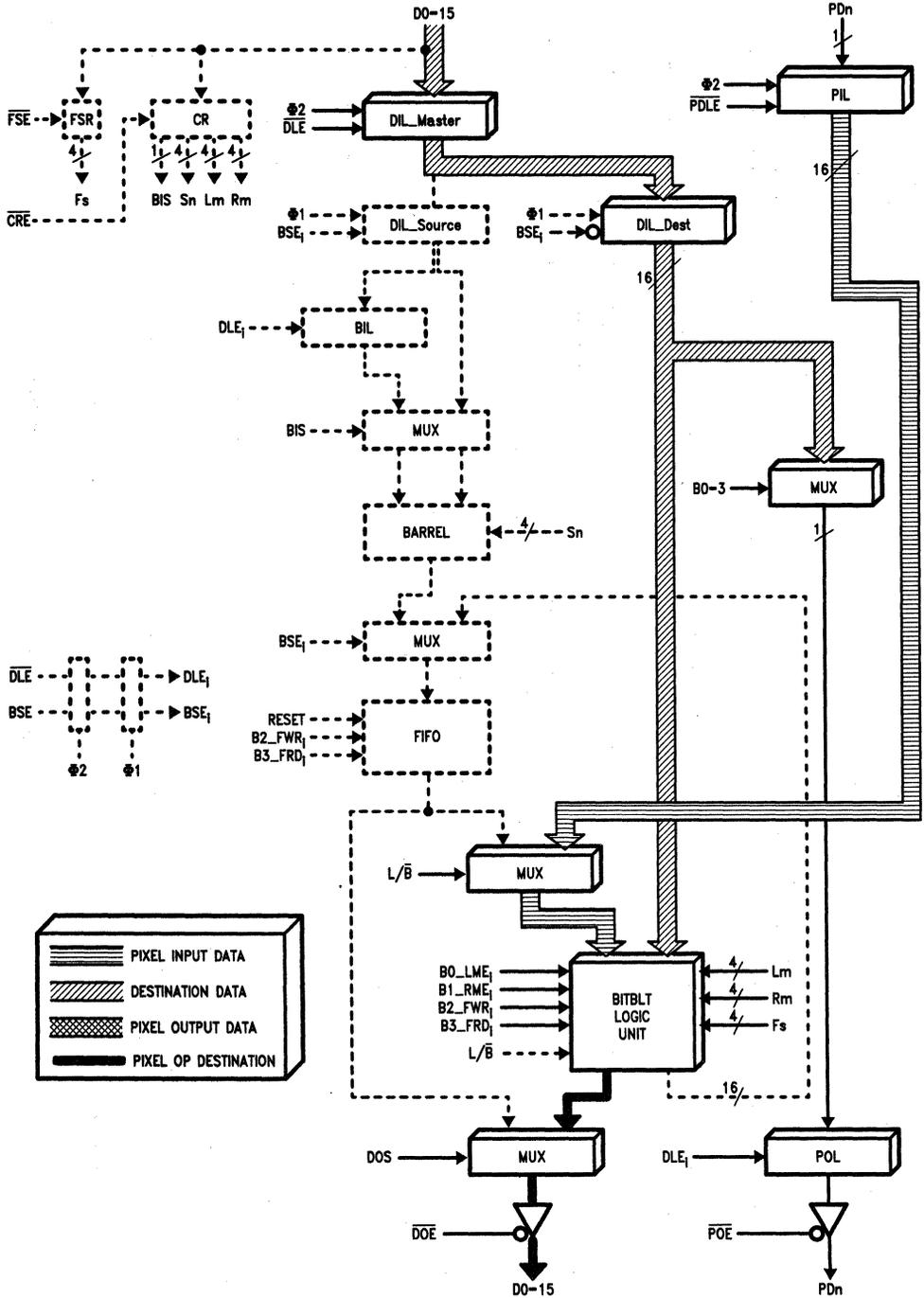


FIGURE 13. Dataflow during a Typical Line Drawing Operation. (RGP Instructions: RDPT, DRPT, DRLN, DRLNS, DRPGN, DRPLN.)

TL/F/9337-4

Operational Description (Continued)

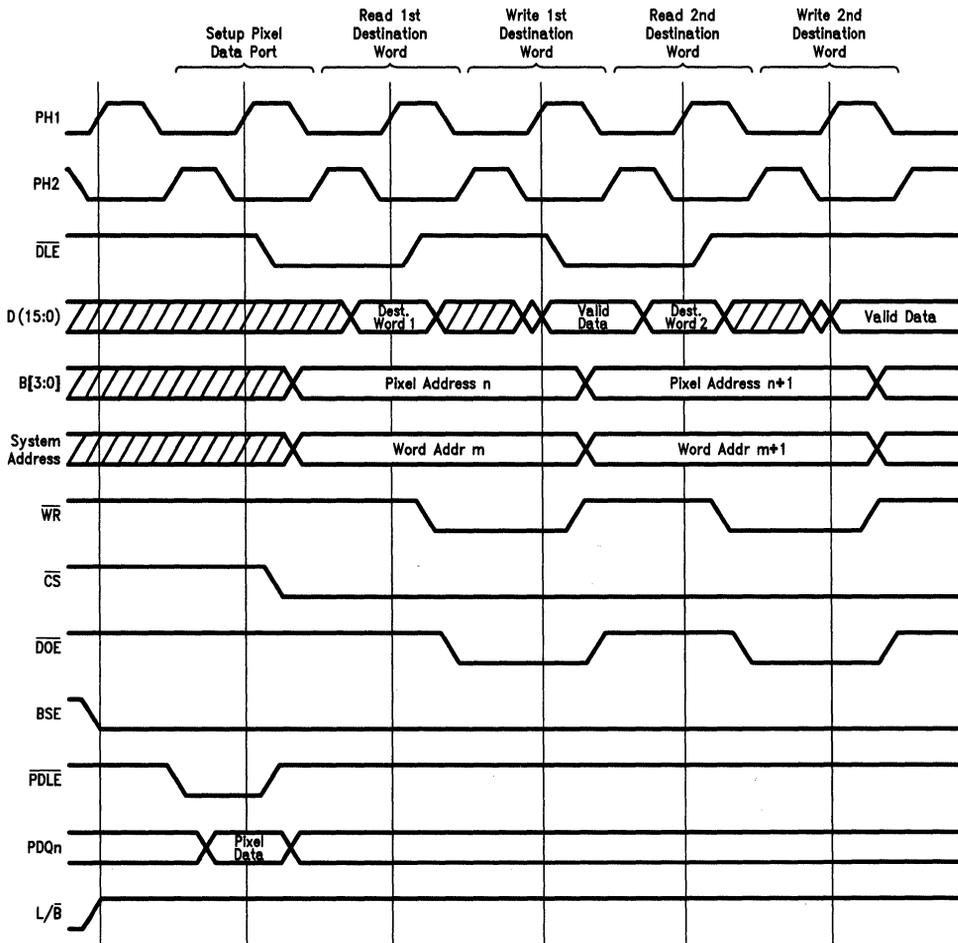


FIGURE 14. Typical Linedrawing Read-Modify-Write Timing

TL/F/9337-29

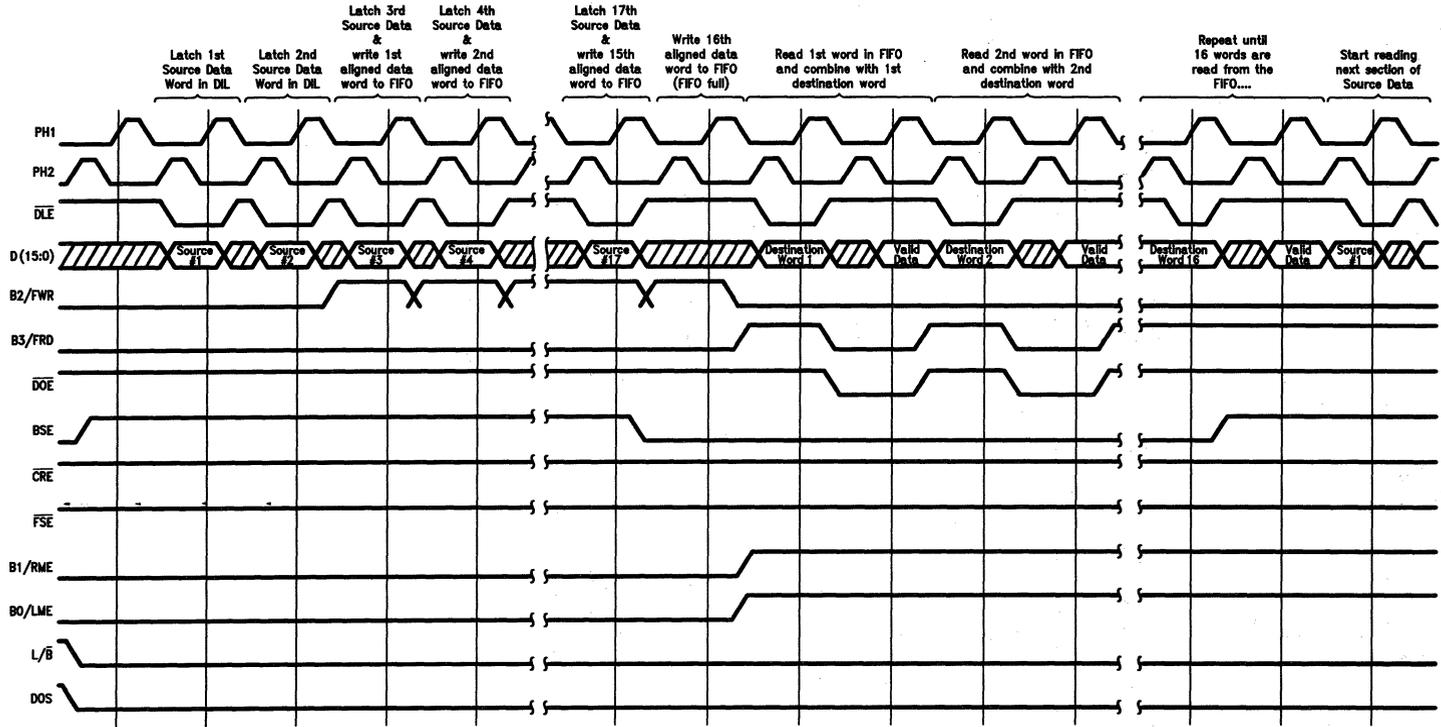


FIGURE 15. Pipelined BITBLT Example for a Transfer of a Single-Width, Non-Aligned Horizontal Line of at Least 16 Words in Length

TL/F/9337-30

Systems Applications

A typical system example illustrating the BPU's interface to the RGP and system elements is shown in *Figure 16*. Included in the example is National's DP8500 Raster Graphics Processor, the DP8520 Video Ram Controller/Driver and the DP8511 BPU. A user defined state machine is used to control the dataflow between the RGP, BPU and Frame Buffer.

Since the BPU and RGP are members of an integrated family of devices, they have been designed to minimize the need for external glue logic between them. The user is left the task of defining the interface from the RGP/BPU to the frame buffer, tailoring it to the system needs.

STATE MACHINE TASKS

Referring to *Figure 16*, the state machine performs the following tasks:

1. Determines the start of an RGP memory cycle.
2. Determines the type of access to be performed.
3. Generates the necessary timing required to carry out the access including wait state generation.
4. Generates refresh timing signals.
5. Generates BPU Control Register strobe (\overline{CRE}).
6. Generates BPU Function Select Register strobe (\overline{FSE}).
7. Generates the \overline{DOE} , \overline{DLE} , \overline{POE} , and \overline{PDLE} strobes to the BPU.

The memory address of the Control Register and Function Select Register are user defined locations in the RGP memory space. It is therefore necessary to generate the \overline{CRE}

and \overline{FSE} strobes in order to select these registers. The \overline{DOE} and \overline{DLE} strobes must also be generated by the user since the timing characteristics of these signals are determined by the memory access timing requirements.

MULTIPLE BITPLANE ARCHITECTURE

The Advanced Graphics Chip Set (AGCS) architectural relationship, in particular that of the RGP/BPU devices, is such that multiple plane memory systems can be developed with relative ease and flexibility. The purpose of segregating the BITBLT and Line Drawing functions into a separate device (the BPU) becomes evident when studying the implementation of a multi-plane display system. *Figure 17* illustrates an approach to designing a multi-plane system.

In any BITBLT based display system, it is necessary to be able to move data rapidly between two areas of memory without restrictions imposed by memory organization. In multiplane systems it is necessary to allow for data operations to occur between planes of memory as well as locally within the bit-plane. The ability is governed by the logic in the video plane control block.

If a BPU is incorporated in each bit-plane of memory and the plane controller is designed correctly, data transfers can be performed in parallel across n-planes of memory, between specific bitplanes, as well as locally on each bit-plane. As an added bonus, this design approach allows the system to be expanded to any practical number of bit-planes while maintaining the performance and throughput characteristics of a single plane system.

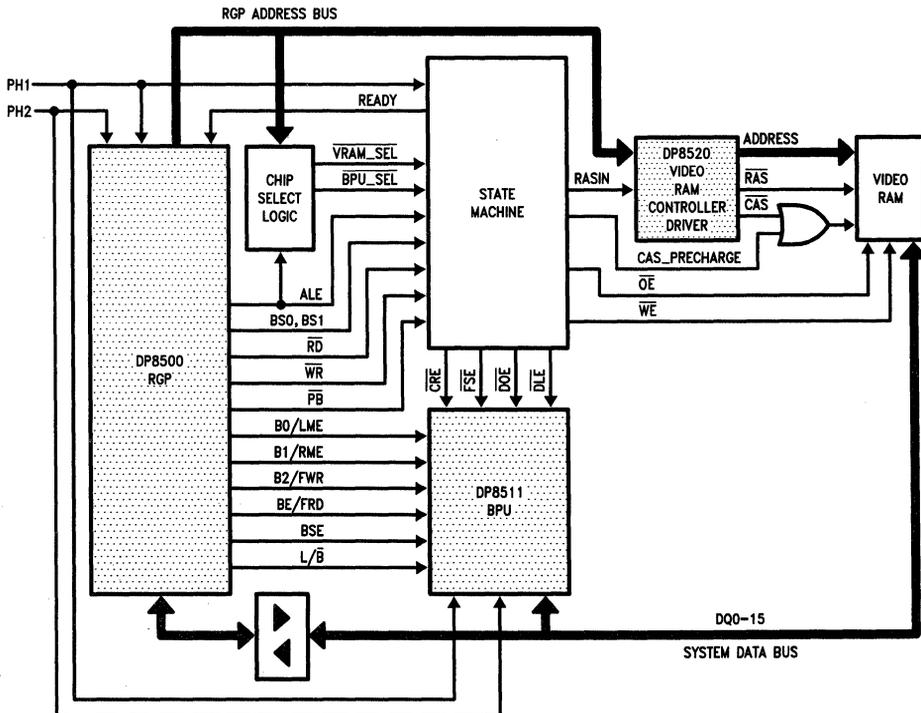


FIGURE 16. DP8500 System Interface

TL/F/9337-5

System Applications (Continued)

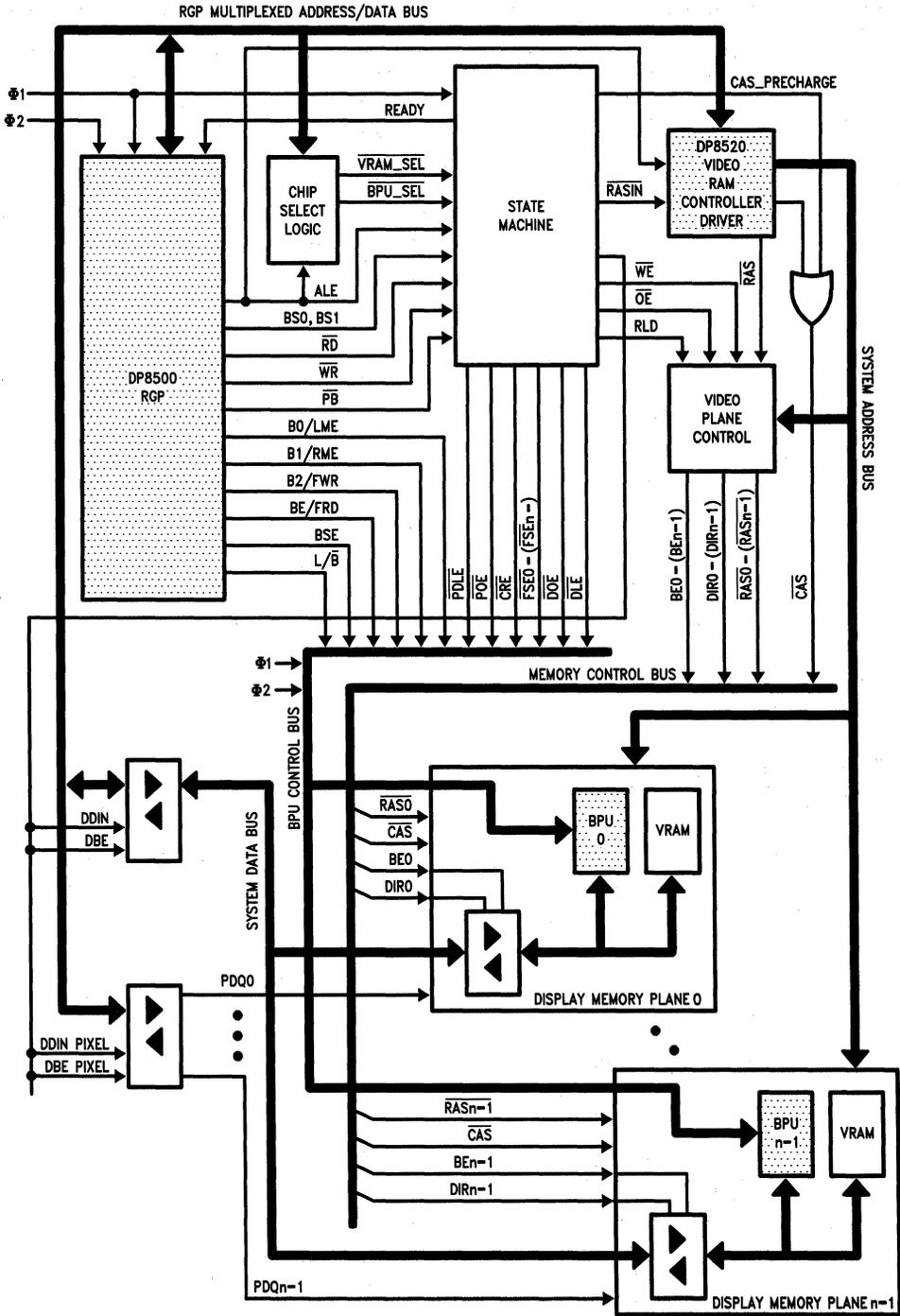
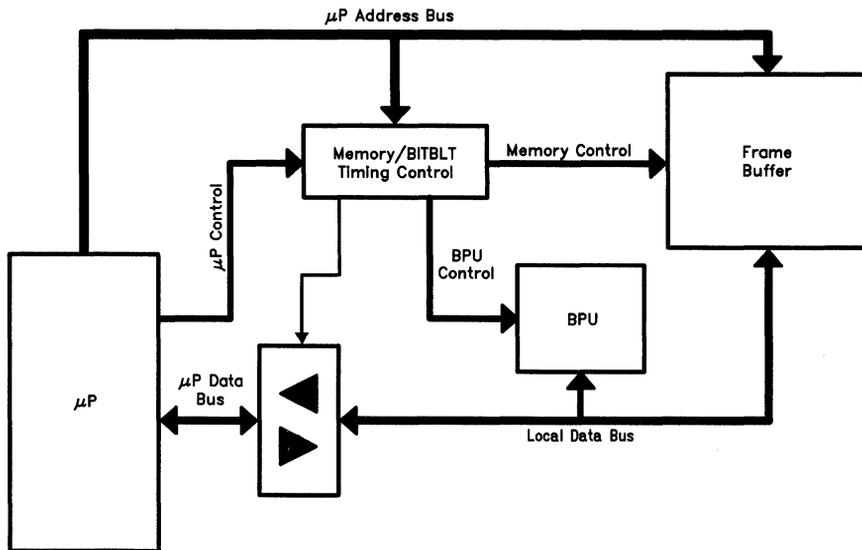


FIGURE 17. Typical RGP-Based Multiple Bitplane System

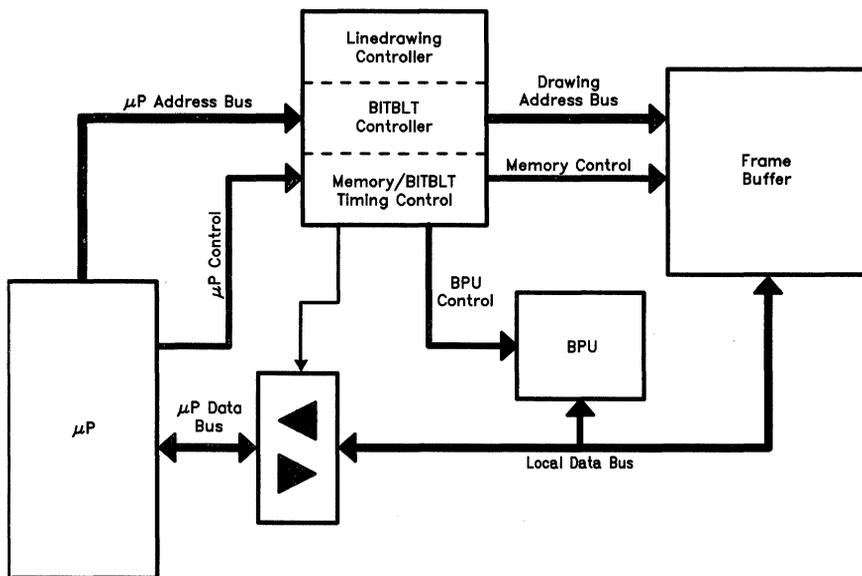
TL/F/9337-6

System Applications (Continued)



a) Simple Microprocessor Interface Utilizing Software for Address Generation

TL/F/9337-31



b) Microprocessor Interface with Dedicated Hardware Controller for BITBLT and Linedrawing Operations

TL/F/9337-41

FIGURE 18. Design Examples for Microprocessor-Based System

System Applications (Continued)

BPU APPLICATIONS UTILIZING A CONVENTIONAL MICROPROCESSOR

If desired, the DP8511 can be used in conjunction with a conventional microprocessor. In this application, the BPU is operated under the control of the microprocessor and a small dedicated timing state controller which produces the basic memory and BPU timing control. The microprocessor would be responsible for interpreting the BITBLT or linedrawing setup parameters and generating the appropriate address and timing control while the BPU would serve as the BITBLT or linedrawing data processor. Implementation can vary as to which level of hardware integration is chosen versus doing the drawing and BITBLT functions in software. This partitioning is ultimately based on the performance level targeted for the application.

Two examples of this partitioning is illustrated in *Figure 18*. In *Figure 18a* the drawing engine consists of a microprocessor, a small memory/BPU timing generator, the frame buffer and the BPU. In this application, the microprocessor is responsible for calculating the address, mask, and other related setup parameters as well as generating the addresses and mask enables used for the memory operations between the frame buffer and the BPU. The timing controller would decode the microprocessor's memory read/write operations and generate the appropriate memory and BPU control timing. As suspected, the drawing performance of this solution would be based on the microprocessor's I/O performance.

Another approach to this design (see *Figure 18b*) would be similar in concept except would partition the functionality so the address and mask enable generation would be part of the memory/BPU timing generation hardware. In this case, the microprocessor would calculate the setup information only and transfer it to a dedicated hardware address controller which would perform the drawing memory cycles in conjunction with the memory/BPU timing controller. In this way the hardware would be able to take advantage of performance features like page mode memory access and interleaved setup and drawing execution.

If a dedicated drawing machine is implemented, the logic necessary would most likely be divided into two parts: a BITBLT controller and a linedrawing controller. Each of these would be responsible for generating the address and control for the memory cycles to be performed. The logic for a BITBLT controller is relatively easy to design since it consists mostly of counters and comparators that control the inner and outer loop for the block transfer. A linedrawing controller, on the other hand, involves the design of a dedicated machine based on a linedrawing algorithm (i.e., Bresenham's linedrawing algorithm). Since this functionality is easier to realize using a software solution, a hybrid solution could be implemented which would assign any linedrawing tasks to the microprocessor and the BITBLT tasks to a hardware controller.

DETERMINING BITBLT PARAMETERS

Before a BITBLT operation can take place, it is necessary to calculate a number of parameters that will be used to control the BPU's activity. These parameters include:

- 1) BIS value,
- 2) The Shift Number (SN),
- 3) The Left Mask value (LM),
- 4) The Right Mask value (RM).

These values can be calculated by using the following parameters normally specified prior to starting the transfer cycle.

These are:

- 1) BITBLT Source Starting Address (BSS),
- 2) BITBLT Destination Starting Address (BDS),
- 3) BITBLT Width (BLW),
- 4) Source Warp (vertical word address offset),
- 5) Destination Warp (vertical word address offset),
- 6) and the BITBLT direction.

Many of these parameters are used to calculate the inner and outer loop values for the BITBLT control logic. These are, for example, used as comparator values for the horizontal and vertical counters that control the BITBLT operation. In addition, these parameters are used to calculate the Control Register values for the BPU.

When using the BPU in conjunction with the DP8500 Raster Graphics Processor, all of the setup calculation, Control Register loading, and BITBLT control logic used to generate the drawing cycles remains transparent to the user.

When using the BPU with a microprocessor or hardware controller, it will be necessary to calculate all variables which are used to program the hardware for appropriate operation. The following sections will help in determining these values.

A Simple Inner/Outer Loop Example

In a non-RGP based system, a mechanism must be defined that is responsible for generating the timing and control for the BITBLT operation. This mechanism may be as simple as a microprocessor functioning in tandem with a small state machine that generates the BPU and memory control signals. The microprocessor would be responsible for generating the address information.

A more elaborate solution could be a dedicated state machine that generates the BPU and memory control signals in addition to performing the address generation.

In either case, some intelligent processor will be required to calculate the BPU's control parameters. Determining the level of hardware sophistication for the BITBLT controller versus doing some portion of the control with a microprocessor is solely dependent on the targeted performance level.

Below is an example of a simple inner/outer loop that describes how a BITBLT controller might be implemented. The Starting Addresses and width values are specified in pixel values while the warps are specified in word values.

System Applications (Continued)

```

/* AN INNER/OUTER BITBLT LOOP EXAMPLE          */
/* to show how a BITBLT is performed.         */
/*                                             */

static int
    BSS,          /*BITBLT Source Starting Bit Address */
    BSW,          /*BITBLT Source Word Address (BSS/16) */
    csc,          /*current source address */
    BDS,          /*BITBLT Destination Start Bit Address */
    BDW,          /*BITBLT Dest. Word Address *(BDS/16) */
    cdc,          /*current source address */
    BLW,          /*BITBLT Width -1 */
    BW,           /*BITBLT Width in words */
    BLH,          /*BITBLT Height -1 */
    WC,           /*down counter for horiz. width (in words) */
    VC,           /*down counter for vert. height (in words) */
    start,        /*flag that starts the BITBLT operation */
    stop_bitblt, /*This concludes the whole thing */
    BSWRP,        /*BITBLT Source Warp */
    BDWRP,        /*BITBLT Destination Warp */
    ldst,         /*horiz end-of-line flag */
    fdst,         /*horiz start-of-line flag */
    rme,          /*right mask enable */
    lme,          /*left mask enable */
    dir,          /*Direction of BITBLT */
                /*0 = left to right, 1 = right to left */
    Eob;          /*end of horizontal line flag */

main()
{
/* Calculate BITBLT parameters such as:
    Height
    Width
    Shift Number
    Left and Right Masks
    Direction of BITBLT
    etc.
and download to BITBLT control logic.
*/
do_bitblt_outer_loop();
}
/*****
/* BITBLT function routines */
*****/
do_bitblt_outer_loop()
{
    if (start)
        { /*Calc. the initial source starting word address */
            BSW = BSS>>4; /* BSS/16 */
            csc = BSW;
            /*Calc. the initial destination starting word address */
            BDW = BDS>>4; /* BDS/16*/
            cdc = BDW;
            /*Calculate the initial width of transfer value */

```

System Applications (Continued)

```

    BW = BLW >> 4;
    if (BUN & 0xf)
        BW = BW + 1;
}

while (stop_bitblt == 0)
{
    /* do this until the height is finished */
    if (start)
        VC = BLH; /* set initial vertical count */
    fdst = 1;
    do_bitblt_inner_loop();
    if (Eob) /*if at end of horiz. line then reset values */
    {
        VC = VC - 1;
        WC = BW; /*reset horizontal counter */
        BSW = BSW + BSWRP;
        cdc = BSW;
        BDW = BDW + BDWRP;
        cdc = BDW;
        Eob = 0;
        ldst = 0;
    }
    if (VC == 0)
        stop_bitblt = 0xffff;
}
}

do_bitblt_inner_loop()
{
while (Eob == 0)
{
    /* BITBLT horizontal down-counter */
    if (start)
    {
        WC = BW; /*set initial horizontal cnt */
        start = 0; /*reset flag */
    }
    if (WC == 1)
        ldst = 1;
        do_source_cycle();
        set_mask();
        do_dest_cycle();
        fdst = 0;
    /* Keep track of horiz. count */
    if (Eob == 0)
        WC = WC - 1;
    /* Flag to indicate end-of-horizontal line */
    if (WC == 0)
        Eob = 1;
}
}
}

```

System Applications (Continued)

```

do_source_cycle()
{
/* This routine is the hardware control logic
that reads the source data from memory and
strokes it into the BPU and controls the transfer
into the FIFO.
*/
    csc = csc + 1;
}
do_dest_cycle()
{
/* This routine is the hardware control logic that generates
the appropriate address and control signals to read
the data for the destination cycle.
*/
    cdc = cdc + 1;
}
set_mask()
{
/*
1) if the BLT is from right-to-left and the last word of the
horizontal line is being transferred, then enable the left mask.
2) if the BLT is from left-to-right and the 1st word of the
horizontal line is being processed, then enable the left mask.
3) if the BLT is from right-to-left and the 1st word of the
horizontal line is being processed, then enable the right mask.
4) if the BLT is from left-to-right and the last word of the
horizontal line is being processed, then enable the right mask.
*/
lme = (dir & ldst)|(~dir & fdst);
lme = (dir & fdst)|(~dir & ldst);
} /*end of inner-outer example */

```

Barrel Input Select (BIS)

One item to determine is the BIS value. This is done by selecting the direction that the BITBLT is to take place in. Then the following procedure should be used to resolve the BIS value:

```

Calculate_BIS_Value()
{
static short
BTDR, /*BITBLT down and to the right */
BTDL, /*BITBLT down and to the left */
BTUR, /*BITBLT up and to the right */
BTUL, /*BITBLT up and to the left */
dir, /*direction data is read from memory */
/*0=left-to-right, 1=right-to-left */
BIS, /*BIS value to be used */
set_bis, /*flag that forces BIS high */
LM, /*left mask value */
RM, /*right mask value */
SN, /*4-bit shift number */
/*when BITBLTing to-the-right... */
if (BTDR|BTUR)
    dir = 0;
/*when BITBLTing to-the-left... */
if (BTDL|BTUL)
    dir = 1;

```

```

/*if the shift number is zero then always
set BIS to one. This is done to save reading
the source data twice */
if (SN = 0)
    set_bis = 1;
BIS = dir|set_bis;
}

```

Shift Number (SN)

The shift number is calculated by subtracting the 4 least-significant bits of the Destination Address from the 4 least-significant bits of the Source Address.

```

Calculate_Shift_Number()
{
static short
BSS, /*Source Starting Address */
BDS, /*Destination Starting Address */
SN, /*Resultant Shift Number */
SN = (BSS - BDS) & 0xf;
}

```

System Applications (Continued)

Left and Right Masks (LM, RM)

The masks are calculated by using the destination pixel address in conjunction with the BITBLT Width value and the direction of the BITBLT. Generally speaking, the masks are enabled only when operating on either of the two words that bound a scanline of BITBLT data. For example, when BITBLTing from right-to-left, the right mask is enabled by detecting the start of a new BITBLT destination scanline while the left mask is enabled when the leftmost word of the BITBLT destination scanline is being processed.

```
Calculate_Masks()
{
  static short
  RM,          /*Right Mask Value */
  LM,          /*Left Mask Value
  BDS,         /*BITBLT Destination Starting Addr */
  BLW,        /*BITBLT Width - 1 */
  if (dir = 0) /* left-to-right BITBLT */
    {LM = BDS & 0xf;
     RM = (BDS + (BLW-1)) & 0xf;
    }
  if (dir = 1) /* right-to-left BITBLT */
    {LM = (BDS - (BLW-1)) & 0xf;
     RM = (~BDS & 0xf) + BLW;
    }
}
```

If additional masking is being generated, for example, a clipping mask, then this must be taken into consideration when establishing the boundary mask values since the clipping mask may affect the boundary masks values.

Additional Considerations

Another item that must be determined in order to achieve the maximum BITBLT performance is whether or not two source read cycles are necessary during the source operation to align the source area to the destination area. This decision can be based on the following criteria:

- 1) If the pixel starting address of the source is less than the pixel starting address of the destination and the BITBLT direction is from right-to-left, then it will be necessary to read two adjacent source words during the source read cycle. This is because the alignment process requires a portion of the adjacent word in order to makeup a full 16-bit value for the destination RMW cycle.

- 2) If the pixel starting address of the source is greater than or equal to the pixel address of the destination and the BITBLT direction is from left-to-right and the shift number is greater than zero, then it will be necessary to read two adjacent source words during the source read cycle. Again, this is because the alignment process requires a portion of the adjacent word to complete the alignment to the destination.

Upon evaluation of the above statements, it is safe to deduct that when shifting the source from right-to-left, two source words must be read in order to align the source to the destination properly. Conversely, when shifting the data from left-to-right, one source word must be latched into both word positions of the Barrel Shifter so that an effective rotate right can be executed. This can be done by allowing an additional clock cycle to occur after latching the data into the Data Input Latch which will transfer a copy of the DIL's contents to the Barrel Input Latch. The shift value in this later case is (BSS - BDS) & 0xf.

BITBLT APPLICATIONS

The following section gives a detailed analysis of how a BITBLT operation is executed using the DP8511. The objective is to copy a 20-bit wide single height source block from pixel address 0002h to a non-aligned destination pixel address of 0805h using an "AND" function code and in a left-to-right BITBLT direction. Since both the source and the destination data areas are not aligned to a word boundary, it becomes necessary to shift the source data so its left-most boundary is aligned to the bit address of the targeted destination area. Also, since the width of the BITBLT block is 20 bits, the operation will involve two horizontally adjacent words. Because both words are considered to be boundary words it will be necessary to perform Read-Modify-Write cycles for each word during the destination portion of the operation.

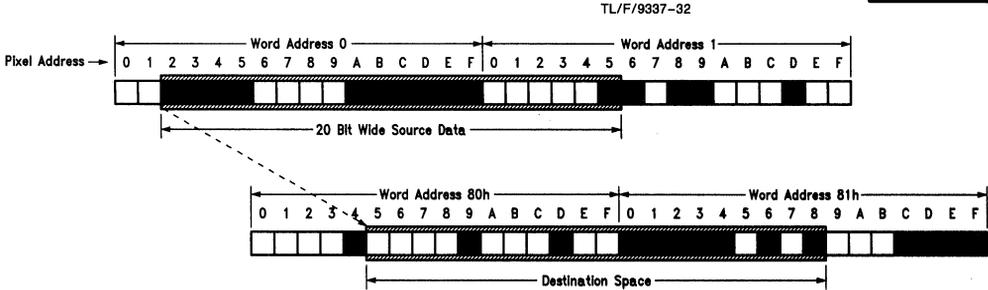
Calculating how to align the source to the destination area is essential to determining the shift, left, and right mask values. See the section, *Determining BITBLT Parameters*, for more information on how to calculate these parameters.

System Applications (Continued)

LEGEND

■ = logic 0

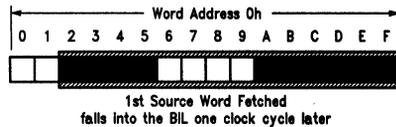
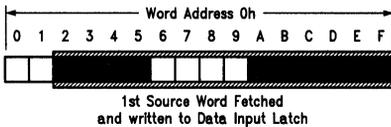
□ = logic 1



TL/F/9337-33

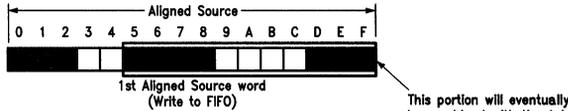
Procedure:

- 1) Load Function Code to FS Register, value equal to 1 ("AND" opcode).
- 2) The source data must be rotated 3 positions to the right so it will be aligned to the destination space specified. Since the BPU cannot do a right shift the data must be shifted 13 positions to the left to achieve this effect. In order to rotate the 1st source word, it must be present in both the Barrel Input Latch (BIL) as well as the Data Input Latch (DIL). This is achieved by allowing the data to "sit" in the DIL for an additional clock cycle. This will allow the data to transfer into the BIL on the clock cycle following the one used to latch the data into the DIL. The shift number must be set to 13 (SN = 13).
- 3) The BIS value is also set at this time. Its value is chosen to be 0 since data is being fetched in a left-to-right sequence and we don't want to swap the data going to the Barrel Shifter.
- 4) The first data word is now latched into the Data Input Latch and one clock cycle later falls into the Barrel Input Latch. This will present a 32-bit wide data word to the Barrel Shifter.



TL/F/9337-34

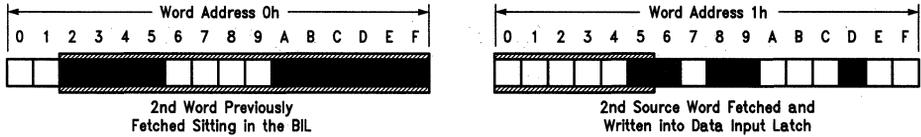
- 5) Data enters the Barrel Shifter and is shifted to the left 13 bits (effectively a right shift of 3). The aligned source is then latched into the FIFO on the next clock cycle by asserting FWR.



TL/F/9337-35

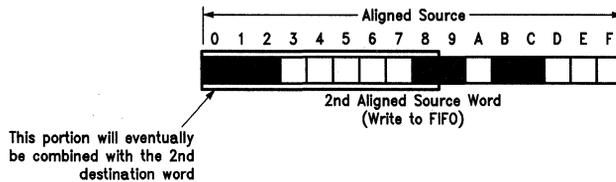
System Applications (Continued)

- 6) The next task is to get the portion of the source data that is to be combined with the 2nd destination word. This will again require the source to be rotated to the right 3 bits. This is effectively a left shift of 13 bits. The shift number must be set to 13. Actually, this value is the same from the first alignment process so it's not necessary to set it again.
- 7) The second source word (word address 1) is latched into the DIL and is concatenated with the 1st source word which has been sitting in the BL.



TL/F/9337-36

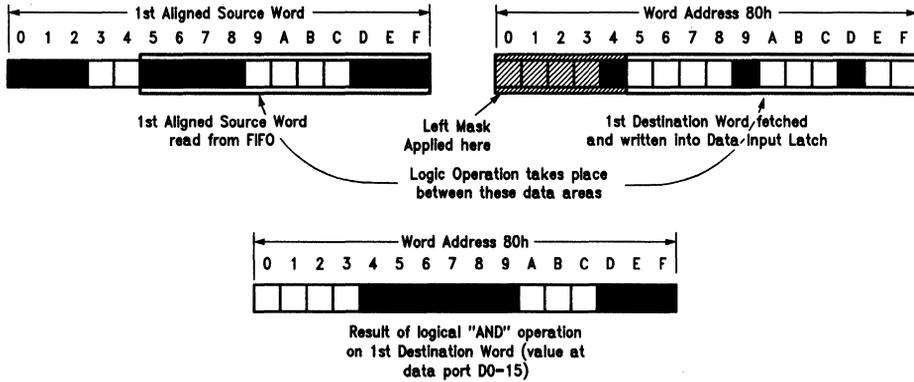
- 8) Data enters the Barrel Shifter and is shifted 13 places to the left which effectively performs a right shift of 3 positions. This aligns the second source word for combination with the second destination word. The aligned source is then latched into the FIFO on the next clock cycle by asserting FWR.



TL/F/9337-37

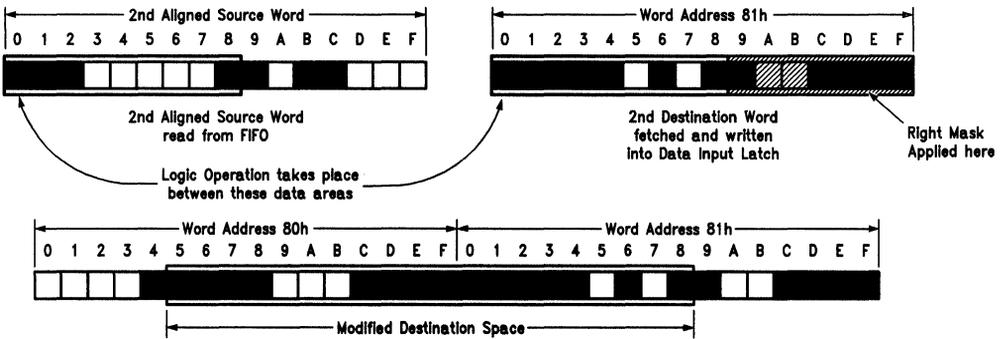
- 9) Now that the source data has been aligned, the next step is to read the two destination words and combine the source data, which is in the FIFO, with them. This requires that the masks be set before doing the two read-modify-write operations. The first destination word has 5 bits on the leftmost side that needs to be masked from alteration. The second word has 7 bits on the rightmost side that need masking. The two mask values can be loaded to the RM and LM bits in the Control Register at one time and are enabled when needed by using LME and RME. LM is set to 5, RM is set to 8 (complement of 7). These values actually should get written to the Control Register at the same time the SN values did to save the unnecessary write cycles.
- 10) Set BSE = 0, LME = 1, RME = 0, and read the first destination word (Adr = 80) into the Data Input Latch. The data will fall into the destination pipe and be routed to the Logic Unit.
- 11) The first aligned source word is read from the FIFO by asserting FRD. This data is "AND'ed" with the destination data in the BITBLT Logic Unit and the result appears on the D0-D15 lines.

System Applications (Continued)



TL/F/9337-38

- 12) Set LME = 0, RME = 1, and read the second destination word (Adr = 81h) into the Data Input Latch. Again this data will be routed through the destination pipe to the Logic Unit.
- 13) Read the second aligned source word from the FIFO. This data will be "AND'ed" with the destination data in the BITBLT Logic Unit and the result appears on the D0-D15 lines.
- 14) The transfer is complete. The destination space has been modified as shown below.



TL/F/9337-39

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature under Bias	
Commercial	0°C to +70°C
Military	-55°C to +125°C
Storage Temperature Range	-65°C to +150°C
All Input or Output Voltage with Respect to GND	-0.5V to +7V
Power Dissipation @ 20 MHz	0.5W
ESD rating to be determined.	

Note: Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.

DC Electrical Characteristics

T_A = 0°C to +70°C Commercial (T_A = -55°C to +125°C Military), V_{CC} = 5V ± 10%, GND = 0V

Symbol	Characteristics	Conditions	Min	Typ	Max	Units
V _{IH}			2.0		V _{CC} + 0.5	V
V _{IL}			-0.5		0.8	V
V _{CH}	MOS Clock High	PH1, PH2 Pins Only, MOS	V _{CC} - 0.5		V _{CC} + 0.5	V
V _{CL}	MOS Clock Low	PH1, PH2 Pins Only, MOS	-0.5		0.3	V
V _{CLT}	MOS Clock Ringing	PH1, PH2 Pins Only, MOS	-0.5		0.5	V
V _{TCH}	TTL Clock High	TTL Clock/PH1 Only	2.5			V
V _{TCL}	TTL Clock Low	TTL Clock/PH1 Only			0.8	V
V _{OH}		I _{OH} = -3 mA	2.4			V
V _{OL}		I _{OL} = 3 mA			0.5	V
I _{IN}	Leakage Current	V _{IN} = V _{IH} or V _{IL}			± 10	μA
I _{OZ}	TRI-STATE® Leakage	V _O = V _{CC} or GND			± 10	μA
I _{CC1}	Quiescent Current	PH1, PH2 at 20 MHz			12	mA
I _{CC2}	Supply Current	PH1, PH2 at 100 kHz			12	mA
I _{CC3}	Supply Current	PH1, PH2 at 20 MHz			25	mA
C _{IN}	Input Capacitance	f _{in} at 1 MHz			10	pF
θ _{JA}	Thermal Resistance - Ceramic PGA Package				5.9	°C/W
θ _{JC}	Junction to Ambient				3.5	°C/W
	Junction to Case					

Note: All output test conditions are 50 pF plus one TTL load.

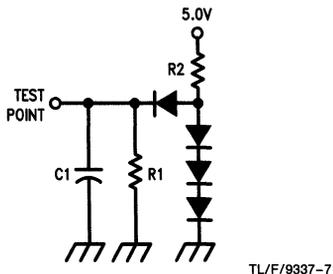


FIGURE 19. BPU Output Test Load Circuitry

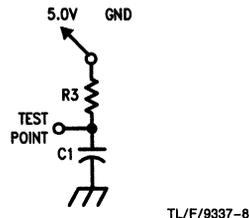


FIGURE 20. BPU Output TRI-STATE Test Load Circuitry

- Note 1: C1 = 50 pF
 R1 = 6 kΩ
 R2 = 1.3 kΩ
 R3 = 1.8 kΩ

Note 2: Connect SW to +5V for tpLZ and tpZL measurements.

Note 3: Connect SW to GND for tpHZ and tpZH measurements.

DP8511 AC Electrical Characteristics*

Name	Figure	Description	Conditions	Commercial		Military**	
				Min	Max	Min	Max
f_{mos}		PH1, PH2 MOS Clock Frequency	TCS = 0		20 MHz		20 MHz
$t_{c[mos]}$		PH1, PH2 MOS Clock Period	TCS = 0	50		50	
f_{ttl}		PH1 TTL Clock Frequency	TCS = 1		17 MHz		17 MHz
$t_{c[ttl]}$		PH1 TTL Clock Period	TCS = 1	58.8		58.8	
t_{ph1}	22	PH1 High Time (TCS=0)	RE 50% to Next FE 50%	19		19	
t_{ph1H}	22	PH1 High Time (TCS=1)	RE 1.5V to Next FE 1.5V	25		25	
t_{ph1L}	22	PH1 Low Time (TCS=1)	FE 1.5V to Next RE 1.5V	25		25	
t_{ph2}	22	PH2 High Time (TCS=0)	RE 50% to Next FE 50%	19		19	
t_{ck1}	22	50% PH1 RE to 50% PH2 RE		25		25	
t_{ck2}	22	50% PH2 RE to 50% PH1 RE		22		22	
t_{nov1}	22	Non-Overlap Time	PH2-to-PH1 50%	3		3	
t_{nov2}	22	Non-Overlap Time	PH1-to-PH2 50%	3		3	
t_{cl}	23	Data Switching Time	From L/ \bar{B} , DOS 1.5V	43		43	
t_{les2f}	24	\overline{DLE} , \overline{PDLE} Setup Time (TCS=0)	Before PH2 FE 50%	10		10	
t_{les1r}	24	\overline{DLE} , \overline{PDLE} Setup Time (TCS=1)	Before PH1 RE 1.5V	10		10	
t_{leh2f}	24	\overline{DLE} , \overline{PDLE} Hold Time (TCS=0)	After PH2 FE 50%	10		10	
t_{leh1r}	24	\overline{DLE} , \overline{PDLE} Hold Time (TCS=1)	After PH1 RE 1.5V	10		10	
t_{leis}	24	\overline{DLE} , \overline{PDLE} Invalid	Before PH2 RE 50%	5		5	
t_{ds2f}	24	Data/Control Setup Time (TCS=0)	Before PH2 FE 50%	8		8	
t_{dh2f}	24	Data/Control Hold Time (TCS=0)	After PH2 FE 50%	8		8	
t_{ds1r}	24	Data/Control Setup Time (TCS=1)	Before PH1 RE 1.5V	5		5	
t_{dh1r}	24	Data/Control Hold Time (TCS=1)	After PH1 RE 1.5V	10		10	
t_{ctls2f}	24	\overline{RESET} Setup Time (TCS=0)	Before PH2 FE 50%	10		10	
t_{ctlh2f}	24	\overline{RESET} Hold Time (TCS=0)	After PH2 FE 50%	8		8	
t_{ctls1r}	24	\overline{RESET} Setup Time (TCS=1)	Before PH1 RE 1.5V	8		8	
t_{ctlh1r}	24	\overline{RESET} Hold Time (TCS=1)	After PH1 RE 1.5V	10		10	
t_{bses2f}	24	BSE Setup Time (TCS=0)	Before PH2 FE 50%	5		5	
t_{bseh2f}	24	BSE Hold Time (TCS=0)	After PH2 FE 50%	8		8	
t_{bses1r}	24	BSE Setup Time (TCS=1)	Before PH1 RE 1.5V	5		5	
t_{bseh1r}	24	BSE Hold Time (TCS=1)	After PH1 RE 1.5V	10		10	
t_{pdn1r}	25	Valid Pixel Data	After PH1 RE 50%		50		50
t_{dpZH}	27	Data TRI-STATE to Active High	After \overline{DOE} , \overline{POE} FE 1.5V		25		30
t_{dpZL}	27	Data TRI-STATE to Active Low	After \overline{DOE} , \overline{POE} FE 1.5V		25		30
t_{dpHZ}	27	Data Active High to TRI-STATE	After \overline{DOE} , \overline{POE} RE 1.5V		25		30
t_{dpLZ}	27	Data Active Low to TRI-STATE	After \overline{DOE} , \overline{POE} RE 1.5V		25		30
t_{pw}	26	\overline{CRE} , \overline{FSE} Pulse Width	FE 1.5V to RE 1.5V	15		15	
t_s	26	Control Reg. Data Setup Time	Before \overline{CRE} , \overline{FSE} RE 1.5V	10		10	
t_h	26	Control Reg. Data Hold Time	After \overline{CRE} , \overline{FSE} RE 1.5V	5		5	
t_{dq1r}	28	Valid Output Data (DOS=0)	After PH1 RE 50%		40		45
t_{dq1r}	28	Valid Output Data (DOS=1)	After PH1 RE 50%		40		45
$t_{vb ses}$	29	BSE Setup for Valid \overline{DLE} Cycle	Before (PH1 * \overline{DLE})	1tc		1tc	
t_{vifwr}	29	Valid FIFO Write†	From \overline{DLE}	1tc	1tc	1tc	1tc
t_{vifrd}	29	Valid FIFO Read††	From FWR	2tc		2tc	

DP8511 AC Electrical Characteristics* (Continued)

Notes:

***Conditions**

Commercial: $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = 5V \pm 10\%$, $GND = 0V$

Military: $T_A = -55^\circ\text{C}$ to $+125^\circ\text{C}$, $V_{CC} = 5V \pm 10\%$, $GND = 0V$

All values are in nanoseconds unless otherwise specified.

**Military Specifications are preliminary. Please contact your National Semiconductor Sales Office or Distributor for availability and specifications.

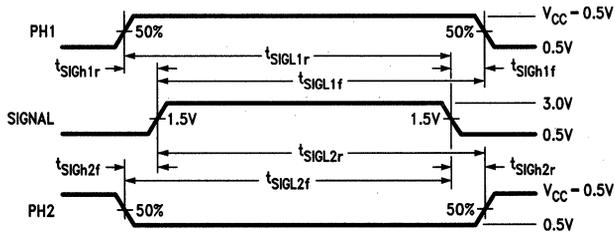
†To insure that valid source data is written to the FIFO, FWR must occur one PH1 cycle time after DLE.

††To ensure that valid data is read from the FIFO, there must be at least two PH1 periods between FIFO write and FIFO read cycles when the FIFO read and write counters are equal. Please consult additional criteria in applications section of this datasheet.

Timing Diagrams

DEFINITIONS

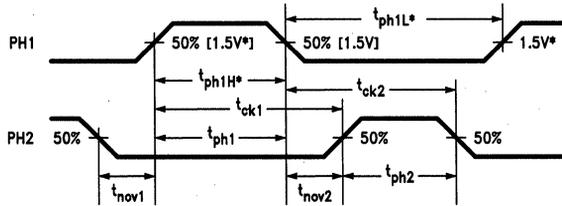
All the timing specifications given in this section refer to 50% of the leading or trailing edges of the appropriate clock phase and 0.5V or 3.0V on the appropriate signal as illustrated in the following figures, unless stated otherwise.



TL/F/9337-9

Note: Data is measured at 1.5V to 50% points of PH1 and PH2.

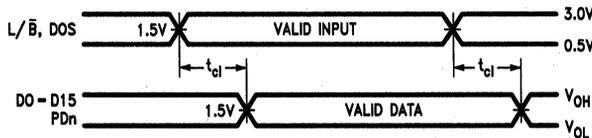
FIGURE 21. Timing Specification Standard



TL/F/9337-10

*TTL Clock Specification

FIGURE 22. Two-Phase MOS/Single-Phase TTL Clock Timing Specification



TL/F/9337-11

FIGURE 23. Mode Control Timing Specification

Timing Diagrams (Continued)

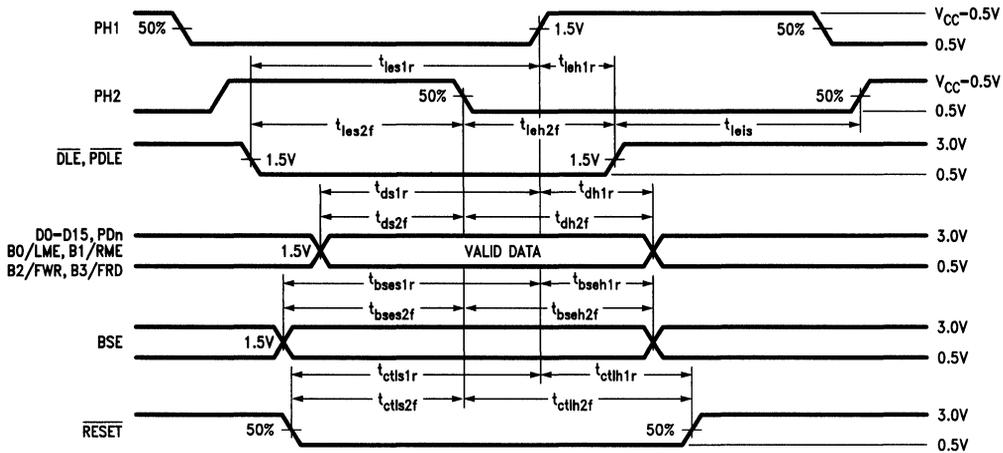


FIGURE 24. Data & Control Inputs Timing Specification

TL/F/9337-12

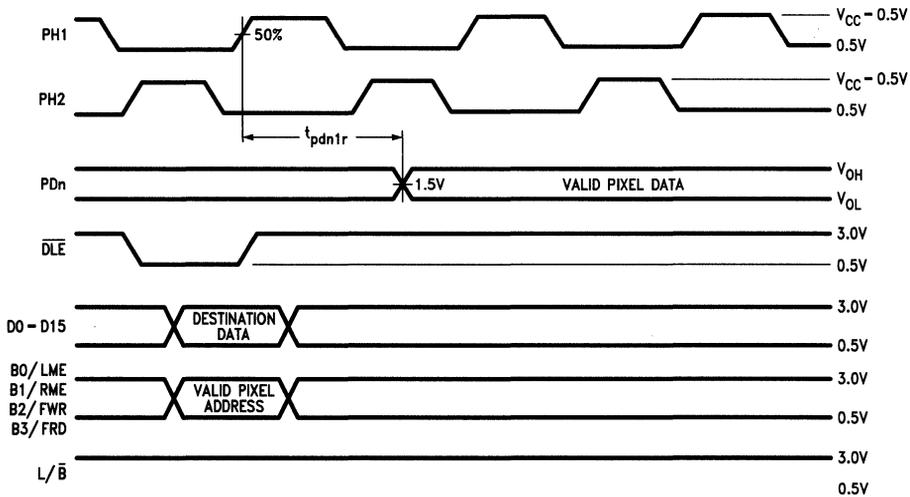


FIGURE 25. Pixel Read Timing Specification

TL/F/9337-13

Timing Diagrams (Continued)

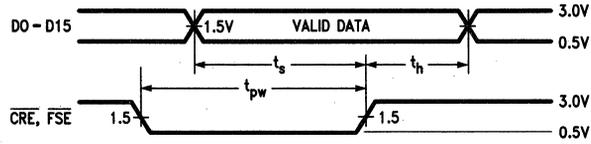


FIGURE 26. CRE and FSE Timing Specification

TL/F/9337-14

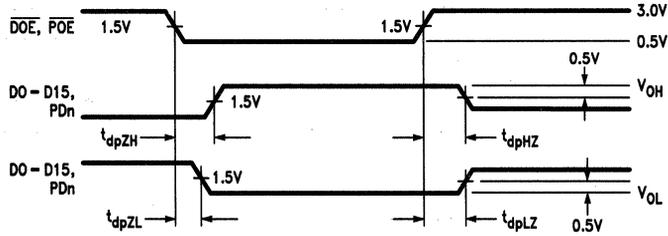


FIGURE 27. Output Enable Timing Specification

TL/F/9337-15

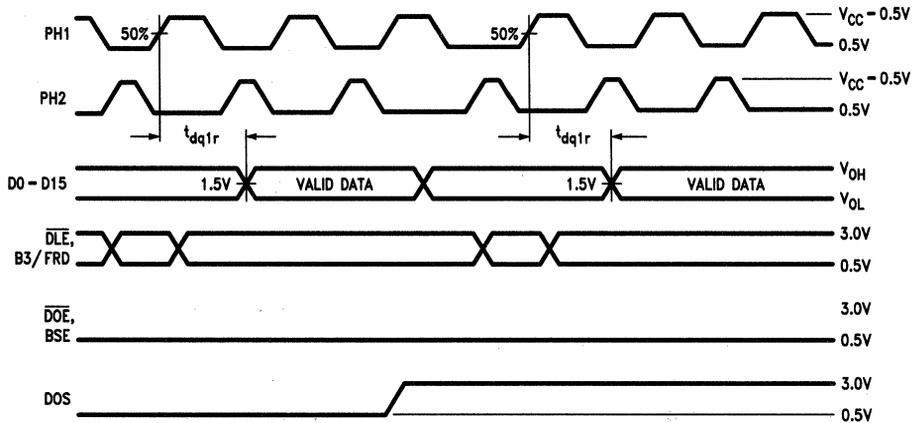


FIGURE 28. FIFO/DIL_Dest to Valid Data Timing Specification

TL/F/9337-16

Timing Diagrams (Continued)

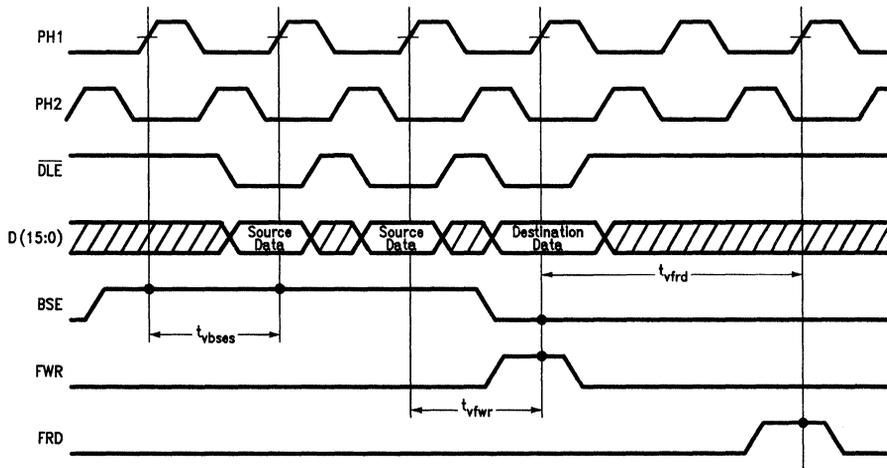


FIGURE 29. BSE and FWR Timing Relationships

TL/F/9337-40



DP8512 Video Clock Generator

General Description

The DP8512 is a clock generator intended for use in medium- to high-performance CRT graphics systems. The device simplifies timing and minimizes phase skew between the various signals involved in the transfer of DRAM (or VDRAM) data into a DAC for display on a CRT. The device generates several synchronous clocks from a single crystal resonator input using digital phase locked loop (PLL) techniques. These clock signals include a graphics processor clock, a raster-scan pixel clock, and various gated TTL and ECL clocks required to transfer data from VRAM to video shift registers. Circuitry is also provided which enables the user to phase lock his graphics system to an external video source. The DP8512 is optimized for single-board graphics systems. The DP8513 is a similar device intended for multi-board synchronous clock generation.

The graphics processor clocks (PHI1 and PHI2) are non-overlapping two-phase clocks with MOS-compatible outputs capable of driving 100 pF loads at up to a 20 MHz rate.

The raster-scan pixel clock (PCLK) and the parallel load clock (LCLK3) required for video shift register operation are generated from the crystal resonator source using two programmable counters in a phase locked loop configuration. Video word widths from 4 to 64 in increments of 4 are accommodated by these counters. The ECL-compatible PCLK outputs are capable of operating at up to 225 MHz from either a positive or a negative supply voltage.

Two gated TTL clock outputs (LCLK1 and LCLK2) are also provided to enable easy transfer of data from a VRAM serial

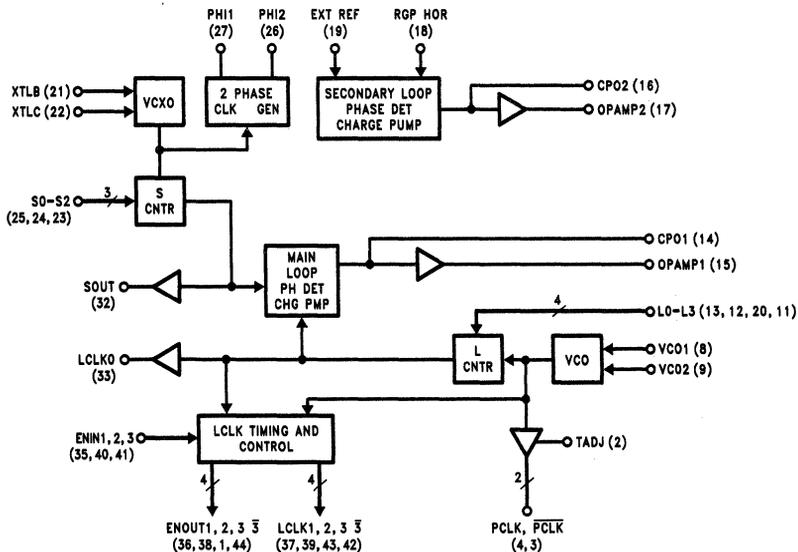
shift register directly into a video shift register or indirectly through a FIFO.

The graphics system can be linked to an external video source by means of a second phase locked loop on the chip. The crystal resonator can be operated as a voltage controlled oscillator allowing adjustment of its frequency until the system's horizontal frequency agrees with that of the external source.

Features

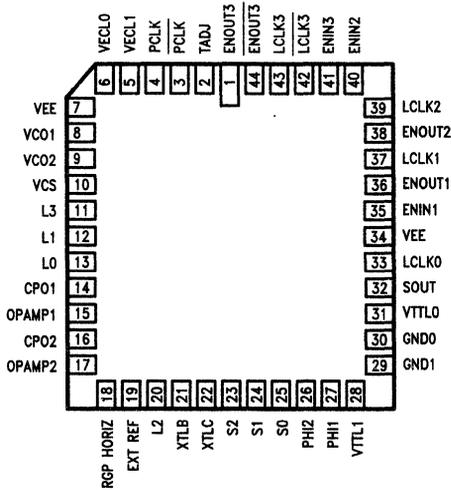
- On-chip crystal oscillator and phase-locked-loop generate synchronized system clock, PCLK, and LCLK
- MOS-compatible single-phase or non-overlapping two-phase system clock output
- 225 MHz ECL differential output pixel clock (PCLK)
- Gated and non-gated load clock (LCLK) outputs ease VRAM-to-VSR synchronization
- Accommodates video word widths from 4 to 64 in increments of 4
- Timing Adjust pin (Tadj) provides a fixed offset adjustment of PCLK to LCLK to ease system design
- ECL circuitry can be referenced to positive or negative power supply
- Enables horizontal synchronization to an external source

Block Diagram (numbers in parentheses indicate pin numbers)



TL/F/8758-1

Connection Diagram



TLF/B758-2

Order Number DP8512V
See NS Package Number V44A

Pin Descriptions

- 1, 44—ENOUT3, ENOUT3:** Differential ECL video enable output synchronous to LCLK0 and gated by ENIN3 (10k and 100k ECL compatible). Outputs require conventional ECL 50Ω terminations.
- 2—TADJ:** Timing Adjust pin allows the PCLK output transition to be offset by 2.5 ns relative to the ECL LCLK3 and ENOUT3 transitions to accommodate the setup and hold requirements of various shift registers more easily. A Logic HI = VECL0 and a Logic LOW = VEE.
- 3, 4—PCLK, PCLK:** Differential ECL pixel clock outputs driven by the VCO in the main loop (10k and 100k ECL compatible). Outputs require conventional ECL 50Ω terminations.
- 5—VECL1:** ECL output buffer positive power supply which can be operated from 0V to +5.7V relative to GND. Note 1.
- 6—VECL0:** ECL internal logic positive power supply which can be operated from 0V to +5.7V relative to GND. Note 1.
- 7, 34—VEE:** ECL negative power supply. Note 1.
- 8, 9—VC01, VC02:** External tank circuit connections for the Pierce VCO. See typical applications for typical wiring.
- 10—VCS:** No connection required.
- 11, 12, 13, 20—L3, L1, L0, L2:** Four bit word input used to select the L Counter modulus. Any modulus from 4 to 64 may be selected in increments of 4. L0 is the least significant bit. A Logic HI = VECL0 and a Logic LOW = VEE. It is recommended that these inputs be bypassed to VECL0 with .01 μF and the jumper lengths be minimized and not cross over any TTL traces as the logic thresholds are VECL0 - 200 mV.
- 14—CP01:** Main loop charge pump output. Used in conjunction with OPAMP1 to form the external loop filter.
- 15—OPAMP1:** Op amp output of the main loop. This output is used to control the pixel clock frequency via the varactor diodes in the LC tank circuit.

16—CP02: Charge pump output of the secondary loop. Used in conjunction with OPAMP2 to form the external loop filter.

17—OPAMP2: Op amp output of the secondary loop. This output is used to vary the crystal frequency (VCXO) in systems where it is desired to lock to an external video source.

18—RGP HORIZ: TTL compatible secondary loop phase comparator input. This signal completes the feedback path from the VCXO by way of the DP8500 Raster Graphics Processor's Horizontal Output. The phase detector #2 is negative edge triggered from this pin.

19—EXT REF: TTL compatible detector #2 reference input. This is the optional horizontal input for systems where it is desired to lock to an external video source. The phase detector #2 is negative edge triggered from this pin.

21, 22—XTLB, XTLC: External connections for the Pierce crystal oscillator. See typical applications for typical wiring.

23, 24, 25—S2, S1, S0: TTL compatible three bit word that determines the S Counter modulus. S0 is the least significant bit.

26, 27—PHI2, PHI1: MOS compatible two-phase non-overlapping clocks. The frequency of these clocks is that of the crystal frequency. If these outputs drive large capacitor loads, a 10 μF or larger capacitor is required directly across the VTTL1 and GND1 pins.

28—VTTL1: TTL output buffer supply. Specified for 5V ± 10% operation. Note 1.

29—GND1: TTL output buffer supply return. Note 1.

30—GND0: TTL internal logic power supply return. Note 1.

31—VTTLO: TTL internal logic positive power supply. Specified for 5V ± 10% operation. Note 1.

32—SOUT: TTL compatible ungated output of the S counter. It is also connected to one of the two inputs of the main loop phase comparator.

33—LCLK0: TTL compatible free-running Load clock. This signal is also connected to an input of the main loop phase comparator.

35—ENIN1: TTL compatible video enable input. A high on this input starts LCLK1 on the next positive transition of LCLK0.

36—ENOUT1: TTL compatible video enable output synchronous to LCLK0 and gated by ENIN1.

37—LCLK1: TTL compatible load clock equivalent to LCLK0, but gated by ENIN1.

38—ENOUT2: TTL compatible video enable output synchronous to LCLK0 and gated on the third positive transition of LCLK0 following a valid ENIN2 input.

39—LCLK2: TTL compatible load clock equivalent to LCLK0 but gated by ENIN2.

40—ENIN2: TTL compatible video enable input. A high on this input starts LCLK2 on the next positive transition of LCLK0.

41—ENIN3: TTL compatible video enable input. A high on this input starts LCLK3 on the next positive transition of LCLK0.

42, 43—LCLK3, LCLK3: Differential ECL compatible load clock synchronous to LCLK0 and gated by ENIN3 (10k and 100k ECL compatible). Outputs require conventional ECL 50Ω terminations.

Note 1: Refer to the Typical Supply wiring diagrams for acceptable wiring of single and dual supply applications.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature	-65°C to +150°C
TTL Signals	
Inputs	7V
Outputs	7V

ECL Signals	
Output Current	-50 mA
Supplies	
VEE to GND	-7V to +0.5V
VECL to GND	-0.5V to +7V
VTTL to GND	-0.5V to +7V
ESD susceptibility (see note 5)	1000V

Recommended Operating Conditions (Notes 1, 2, 3, 4)

Symbol	Parameter	Min	Typ	Max	Units
VTTL0, 1 to GND	TTL Power Supply	4.5		5.5	V
VECL0, 1 to VEE	ECL Power Supply	4.2		5.7	V
VEE to GND		-5.7		0	V
V _{IH}	High Level Input Voltage TTL Inputs	2			V
V _{IL}	Low Level Input Voltage TTL Inputs			0.8	V
I _{OH}	High Level Output Current TTL Outputs			-0.4	mA
	High Level Output Current MOS Compatible Outputs			-0.4	mA
I _{OL}	Low Level Output Current TTL Outputs			8	mA
	Low Level Output Current MOS Compatible Outputs			20	mA
F _{CLK} (Note 6)	Pixel Clock Max Frequency			225	MHz
F _{XTL}	Crystal Oscillator Max Frequency			25	MHz
T _{SU1}	Setup Time ENIN1 to LCLK0	20	11		ns
T _{SU2}	Setup Time ENIN2 to LCLK0	20	11		ns
T _{SU3}	Setup Time ENIN3 to LCLK0	20	11		ns
T _{H1}	Hold Time LCLK0 to ENIN1	0	-9		ns
T _{H2}	Hold Time LCLK0 to ENIN2	0	-9		ns
T _{H3}	Hold Time LCLK0 to ENIN3	0	-9		ns
T _{ambient}	Operating Temp Range	0		70	°C

Note 1: See Timing Waveforms for relevant signal edges (positive or negative) from which all setup and hold times measurements are made.

Note 2: TTL inputs—ENIN1, 2, 3, RGP HORIZ, EXT REF, S0, S1, S2.

Note 3: TTL outputs—SOUT, LCLK0, 1, 2, ENOUT 1, 2; MOS Outputs—PHI1, 2.

Note 4: Inputs L0, L1, L2, L3, T_{ADJ} designed to be tied to VECL for high level or shorted to VEE (or left open) for low level. See input schematics.

Note 5: Human body model; 120 pF thru 1.5 kΩ.

Note 6: F_{CLK} is the maximum frequency that the pixel clock output can be reliably "locked". The VCO range should be controlled to avoid exceeding 235 MHz when the maximum control voltage correction is applied.

DC Electrical Characteristics (Note 1)

Symbol	Parameter	Conditions	Min	Typ	Max	Units		
V_{IC}	Input Clamp Voltage	VTTL0, 1 = 4.5V, $I_{IN} = -18$ mA			-1.5	V		
V_{OH}	Output High Voltage	VTTL0, 1 = 4.5V to 5.5V	TTL Outputs, $I_{OH} = -400$ μ A		VTTL-2	V		
			MOS Outputs	$I_{OH} = -100$ μ A	VTTL-0.4			
				$I_{OH} = -400$ μ A	VTTL-2.3			
		VECL0, 1 = 0V VEE = -4.2V	ECL Outputs, 50 Ω Load to -2V		-1135	-880	mV	
OPAMP Output, $I_{OH} = -1.25 \times I_{CPO}$ Sink			VECL0-1.2		V			
V_{OL}	Output Low Voltage	VTTL0, 1 = 4.5V	TTL Outputs, $I_{OL} = 8$ mA			0.5		
			MOS Outputs	$I_{OL} = +100$ μ A		0.4		
				$I_{OL} = 20$ mA		0.5		
		VECL0, 1 = 0V VEE = -4.2V	ECL Outputs, 50 Ω Load to -2V		-1995	-1490	mV	
OPAMP Output, $I_{OL} = -1.25 \times I_{CPO}$ Source				VEE + 0.5	V			
I_I	Max High Level Input Current	VTTL0, 1 = 5.5V, TTL Inputs, $V_{IN} = 7V$			100	μ A		
I_{IH}	High Level Input Current	VTTL0, 1 = 5.5V, TTL Inputs, $V_{IN} = 2.7V$			20	μ A		
I_{IL}	Low Level Input Current	VTTL0, 1 = 5.5V, TTL Inputs, $V_{IN} = 0.4V$			-200	μ A		
I_O	Output Drive Current	VTTL0, 1 = 5.5V	TTL Outputs, $V_O = 2.25V$		-30	-110	mA	
			MOS Outputs, $V_O = 2.25V$			-135	mA	
I_{CPO}	Charge Pump Current	VEE = -4.2V to -5.7V VECL0, 1 = 0V	Source		-0.2	-0.5	-1.0	mA
			Sink		0.2	0.5	1.0	mA
			TRI-STATE®		-10	0	10	μ A
I_{CC}	Supply Current	VTTL0, 1 = 5.5V	TTL Supply		30	40	mA	
			VECL0, 1 = 0V	VEE = -5.7V, 10k ECL Supply Range		150		210
		VEE = -4.8V, 100k ECL Supply Range		135	185			

Note 1: TTL inputs—ENIN1, 2, 3, RGP HORIZ, EXT REF, S0, S1, S2
 TTL outputs—ENOUT1, 2, SOUT, LCLK0, 1, 2
 MOS outputs—PHI1, PHI2
 ECL outputs—ENOUT3, ENOUT3, PCLK, PCLK, LCLK3, LCLK3

AC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
F_{MAX}	Max VCO Freq (Note 2)	Self Osc	235			MHz
		Ext. Drive	235			
	Max XTL Freq		25	35		MHz
DC	Duty Cycle PHI1, 2	FXTL = 20 MHz	45	55		%
T_{EO1}	LCLK0 to ENOUT1		-7	-2	5	ns
T_{EO2}	LCLK0 to ENOUT2		-7	-2	5	ns
T_{EO3}	LCLK0 to ENOUT3		-15	-7	0	ns
T_{PCLK}	LCLK3, ENOUT3 to PCLK	Tadj = HI (VECL0)	0	2.5	5.0	ns
		Tadj = LOW (VEE)	-2.5	0	2.5	ns

AC Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T _{LCLK1}	LCLK0 to LCLK1		-5	0	5	ns
T _{LCLK2}	LCLK0 to LCLK2		-5	0	5	ns
T _{LCLK3}	LCLK0 to LCLK3		-15	-7	0	ns
T _{PHI1}	LCLK0 to PHI1 			-5		ns
T _{SOUT}	LCLK0 to SOUT 	f _{XTL} = 20 MHz, S = 8		0		ns
T _{SO-PHI1}	SOUT to PHI1 	Smod = 1 thru 8	-15	-5	5	ns
T _{NO}	Nonoverlap Time PHI1 to PHI2	C _L = 50 pF		0		ns
T _R , T _F	Rise, Fall Time PHI1, 2	(Note 1)		4		ns

Note 1: Rise, Fall times measured from 0.5V to VTTL - 2V with C_L = 50 pF.

Note 2: F_{MAX} is not production tested but is assured by characterization to include sufficient margin beyond processing extremes.

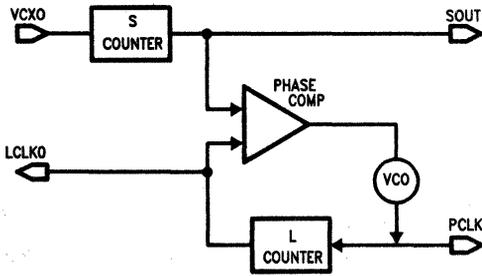
Circuit Operation

The heart of the DP8512 Video Clock Generator is a crystal oscillator which is used as a frequency reference to generate several clock signals required in a video display system. Among the clocks generated are the two-phase clock for driving the RGP and BPU processors, ECL pixel and load clocks (PCLK and LCLK3) for high speed video shift register parallel load and shift operations, and TTL load clocks (LCLK0, LCLK1, and LCLK2) for moving DRAM and FIFO data to the video shift registers. The LCLK and PCLK outputs are all internally synchronized in order to simplify system timing.

The two-phase graphics processor clock (PHI1 and PHI2) operates at the frequency of the crystal oscillator. It is capable of directly driving the raster graphics and BIT-BLT processors. The two-phase clock is closely aligned with the other clocks generated by the device to maximize system operation. The PCLK and LCLK outputs are generated using a digital phase locked loop as shown in *Figure 1*.

where PCLK is the pixel clock frequency, L is the L Counter modulus, and S is the S Counter modulus. When the frequency of the VCO (PCLK) in the phase locked loop is stable the inputs to the phase detector are in phase, thus the S Counter and L Counter outputs are identical in both phase and frequency. The crystal oscillator ensures that the phase and frequency of the S Counter output remains constant. Any drift, or change in frequency, of the VCO will be divided down and appear as a shift in phase at the L Counter output. The phase detector will sense this phase error and generate a correction voltage for the VCO input which is proportional to the magnitude of the frequency error. This correction voltage will change the VCO frequency to eliminate the error thus keeping the loop locked. The correction voltage adjusts the VCO's frequency by changing the capacitance of the varactor in the LC oscillator tank circuit. The varactor's capacitance is proportional to the amount of reverse bias applied across it. The VCO correction voltage is provided from the OP AMP output which is a 3V typical operating range (VEE + 0.5V to VELCO - 1.2V). This operating range provides typically a ±10% VCO frequency tuning range. This operating range limits the device from being used at multiple pixel rates unless an external op amp is added to extend the tuning voltage range or tank circuit components are bandswitched in.

The presence of the S Counter in the loop enables the graphics processor to operate at full speed independent of PCLK frequency. The video shift register's parallel data width determines the L Counter modulus. An 8-bit parallel shift register would use an L Counter modulus of 8 so that a parallel load pulse occurs once every 8 pixel clocks. The L Counter output is used to derive the four LCLK outputs. These 4 LCLKs differ slightly in format to allow for the various system configurations highlighted in the following section.



TL/F/8758-10

FIGURE 1. PLL Block Diagram

The loop consists of the S and L counters, a phase comparator, and a voltage controlled oscillator (VCO) with the relationship between these elements in the loop defined as:

$$PCLK = \frac{VCXO \times L}{S}$$

Circuit Operation (Continued)

The ECL LCLK3 output is used in conjunction with the PCLK output to load data into a high-speed video shift register. The PCLK provides the clock and LCLK3 provides the load signal for the shift register. With a typical video shift register operating at frequencies around 200 MHz timing can become extremely critical. For this reason a Timing Adjust pin (Tadj) is provided that will allow the user to obtain the optimum LCLK3-to-PCLK setup and hold timing relationship.

The other three LCLK outputs (LCLK0, LCLK1, LCLK2) are TTL outputs. They can be used to control a selection of different DRAMs/video shift register configurations as shown in the system architecture section which follows.

Also included on the chip is a secondary phase locked loop which can be used to synchronize the graphics system to an external signal such as the horizontal sync pulse from a television broadcast. A block level diagram of this mode of operation is shown in Figure 2. The crystal oscillator is configured to operate as a voltage controlled crystal oscillator (VCXO). Any change in frequency of the VCXO forces the outputs of the S Counter and the RGP's horizontal counter to shift in phase relative to the external horizontal reference signal. The auxiliary phase comparator senses any phase differences between its two inputs and produces a correction voltage back to the VCXO which is proportional to the amount of the frequency error. This correction voltage will change the VCXO frequency to eliminate the error thus keeping the loop locked.

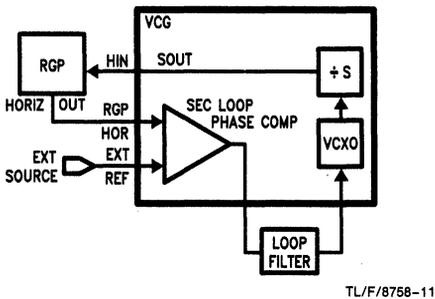


FIGURE 2

TL/F/8758-11

For most applications the crystal oscillator will be in the range of 20 MHz. With a typical 8-bit system this means that the PCLK output will be running at $20 \text{ MHz} \times 8 \text{ bits} = 160 \text{ MHz}$. However, if the system is 16 bits wide instead of 8 bits the PCLK frequency would become $20 \text{ MHz} \times 16 \text{ bits} = 320 \text{ MHz}$ which is beyond the range of the VCO. Therefore the S Counter block must be added to divide the crystal oscillator frequency down to a more manageable frequency. Using the equation and the above example of a 20 MHz crystal oscillator with 16 bits of data, the S Counter is used as a divide-by-two counter to get a PCLK frequency of 160 MHz.

The S Counter can be programmed to divide by any integer up to 8 and the L Counter can be programmed for any word width from 4 bits to 64 bits in increments of 4. Table I shows some of the frequencies possible using various values for the S- and L-modulus.

VARIOUS SYSTEM ARCHITECTURES

Figure 3 demonstrates the DP8512 in a system using a video shift register and a DRAM. Another possible application uses a video DRAM, or VDRAM, in place of the DRAM, as shown in Figure 4. This system differs from Figure 3 in that the VDRAM contains an internal shift register which allows memory to be randomly accessed while data is being output to the video shift registers through the serial port. The output of the VDRAM differs from the standard DRAM in that data is not valid at the output of the internal shift register until an SCLK pulse is provided. The offset in the LCLK3 waveform from LCLK1 is obtained by connecting ENOUT1 to ENIN3.

The third type of system employs a video shift register (such as the DP8515) that contains an onboard FIFO as shown in Figure 5. This architecture simplifies timing requirements in a high-speed multiboard system. The propagation delay time from the DRAM to the video shift register through the back plane is no longer restricted to be less than the clock frequency of the system. The VDRAM can be writing into the FIFO asynchronous to the data being loaded into the shift register. As in the previous systems, one LCLK1 pulse transfers the data to the VDRAM output. LCLK2 then writes several words into the FIFO prior to LCLK3 reading out the first word. The connections required to obtain this timing are shown in the diagram.

TABLE I. Partial Table of PCLK Frequencies

S MOD	8-Bit Word (L = 8)		16-Bit Word (L = 16)		32-Bit Word (L = 32)	
	PCLK Frequency		PCLK Frequency		PCLK Frequency	
	XTL = 10 MHz	XTL = 20 MHz	XTL = 10 MHz	XTL = 20 MHz	XTL = 10 MHz	XTL = 20 MHz
1	80.0 MHz	160.0 MHz	160.0 MHz	na	na	na
2	40.0	80.0	80.0	160.0 MHz	160.0 MHz	na
3	26.7	53.3	53.3	106.7	106.7	na
4	20.0	40.0	40.0	80.0	80.0	160.0 MHz
5	16.0	32.0	32.0	64.0	64.0	128.0
6	13.3	26.7	26.7	53.3	53.3	106.7
7	11.4	22.9	22.9	45.7	45.7	91.4
8	10.0	20.0	20.0	40.0	40.0	80.0

Circuit Operation (Continued)

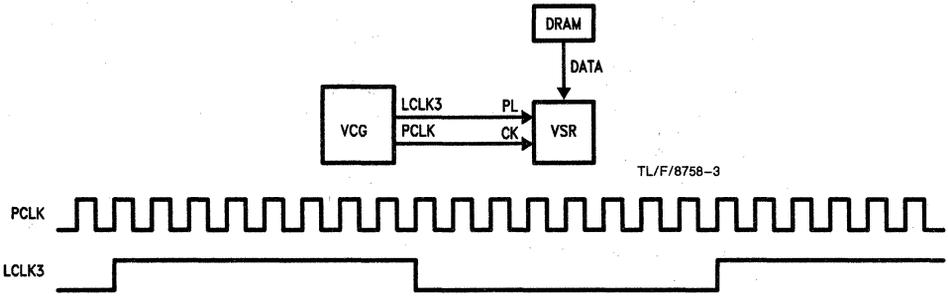


FIGURE 3. System Configuration and LCLK Waveforms using DRAM

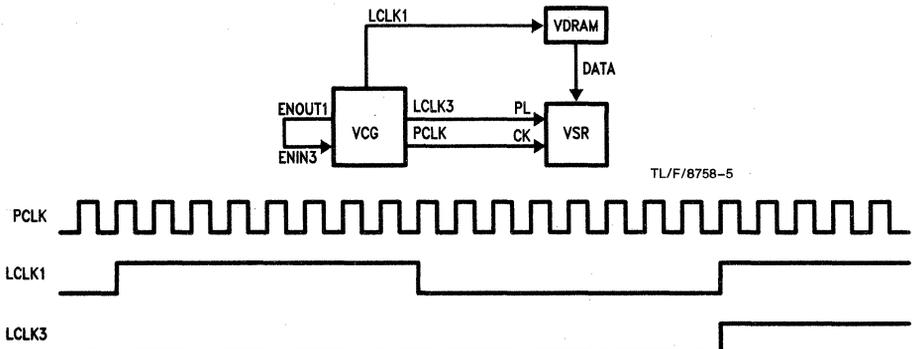


FIGURE 4. System Configuration and LCLK Waveforms using VDRAM

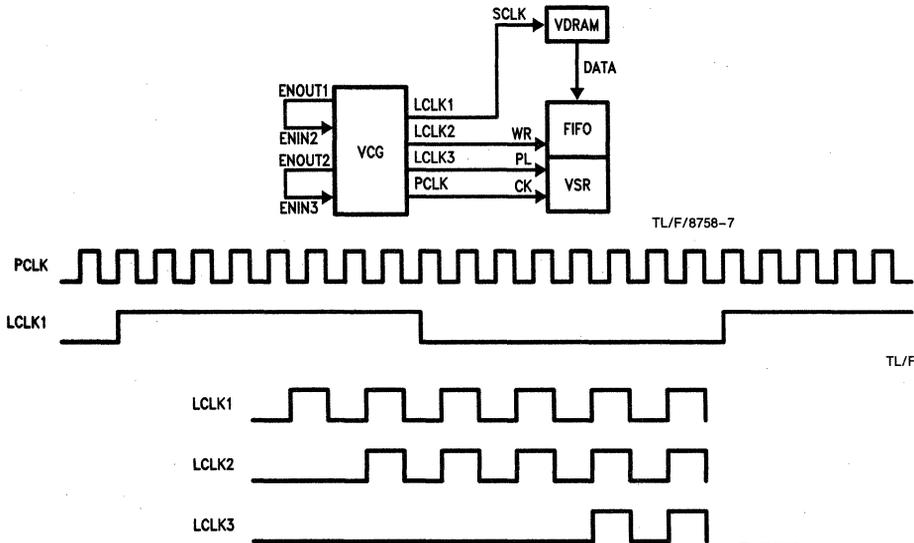
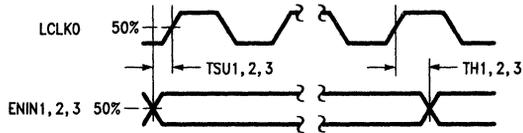


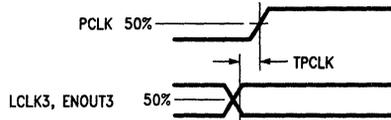
FIGURE 5. System Configuration and LCLK Waveforms using a VSR Containing FIFO

AC Timing Waveforms

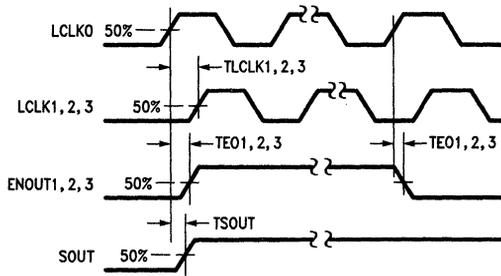
TSETUP, THOLD



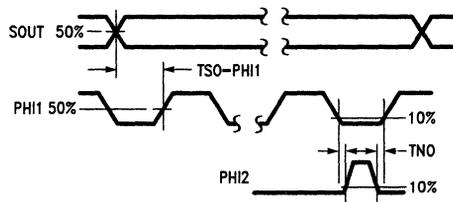
TL/F/8758-12



TL/F/8758-13

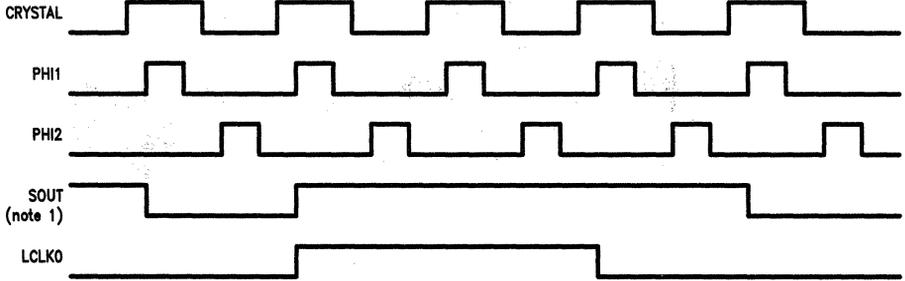


TL/F/8758-14



TL/F/8758-15

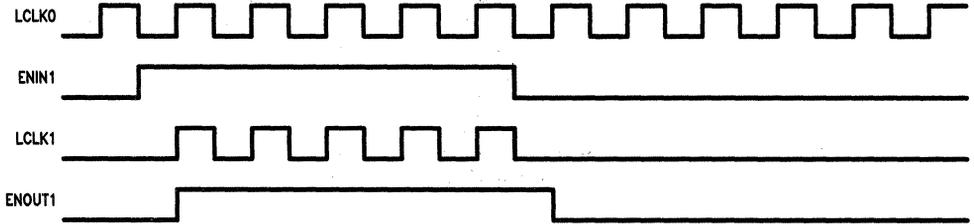
DP8512 Functional Waveforms



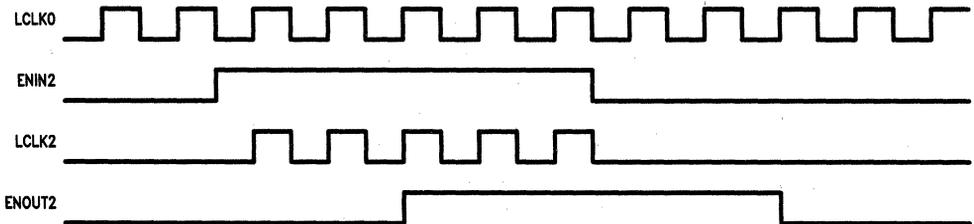
TL/F/8758-16

Note 1: SOUT waveform displayed with S Counter in divide-by-4 mode.

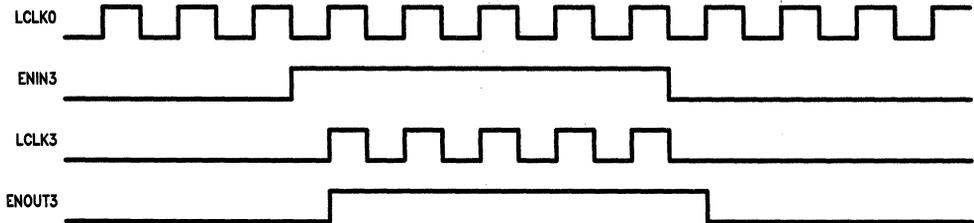
Load Clock Timing Diagram



TL/F/8758-17



TL/F/8758-18



TL/F/8758-19

Counter Modulus Tables

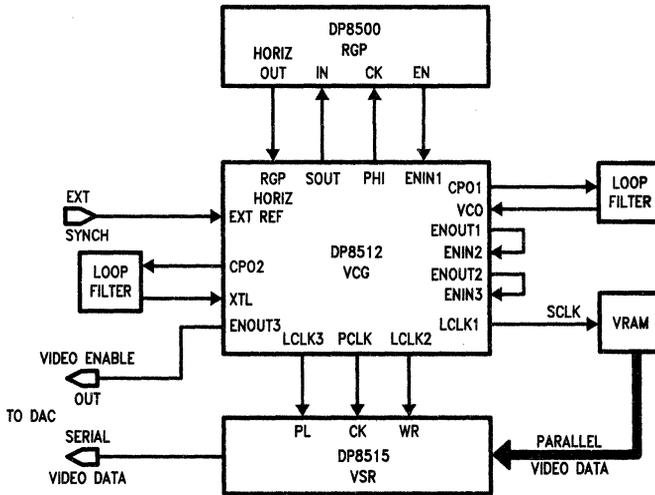
S MOD	S Counter Inputs		
	S2	S1	S0
1	L	L	L
2	L	L	H
3	L	H	L
4	L	H	H
5	H	L	L
6	H	L	H
7	H	H	L
8	H	H	H

L MOD	L Counter Inputs			
	L3	L2	L1	L0
4	0	0	0	0
8	0	0	0	1
12	0	0	1	0
16	0	0	1	1
20	0	1	0	0
24	0	1	0	1
28	0	1	1	0
32	0	1	1	1
36	1	0	0	0
40	1	0	0	1
44	1	0	1	0
48	1	0	1	1
52	1	1	0	0
56	1	1	0	1
60	1	1	1	0
64	1	1	1	1

L = TTL Logic Low Level
 H = TTL Logic High Level

0 = VEE or open
 1 = VECL0, 1

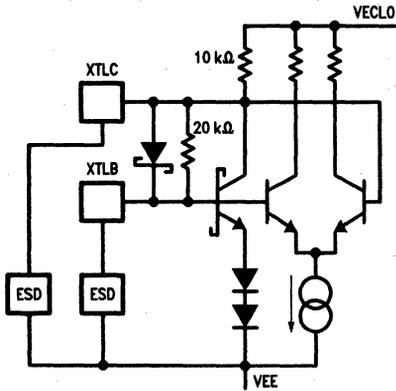
Typical System Diagram



TL/F/8758-20

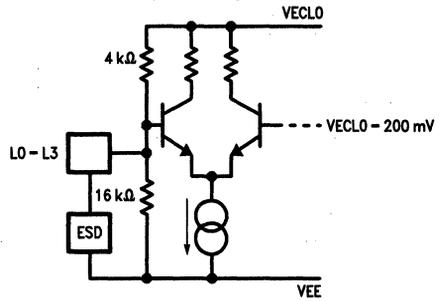
Input Schematics (Note 1)

VCXO Inputs



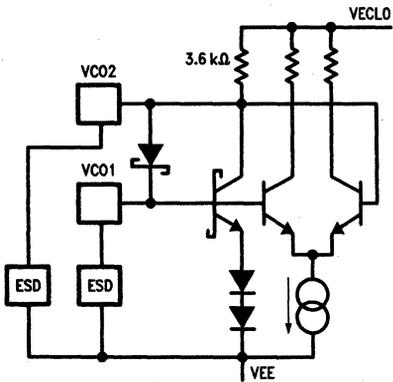
TL/F/8758-21

L0-L3 Inputs



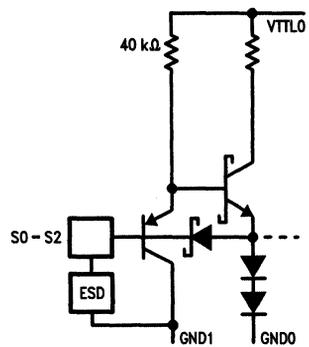
TL/F/8758-22

VCO Inputs



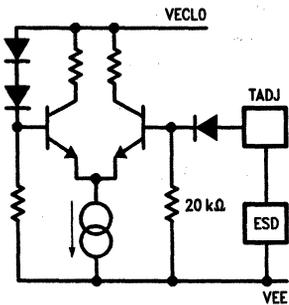
TL/F/8758-23

S Counter Inputs (TTL)



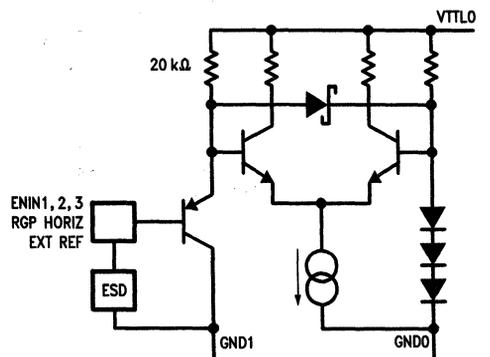
TL/F/8758-24

Timing Adjust Input



TL/F/8758-25

TTL Inputs

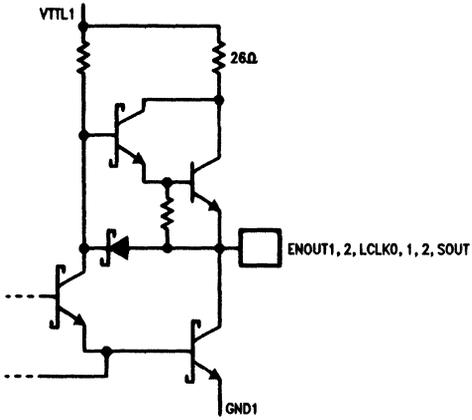


TL/F/8758-26

Note 1: Refer to the typical ESD circuit for all schematics.

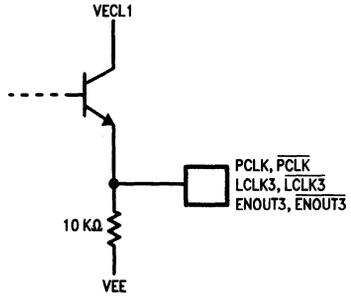
Output Schematics

TTL Outputs



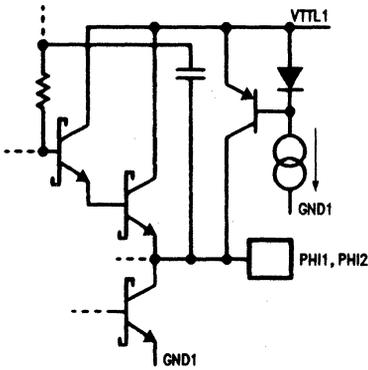
TL/F/8758-27

ECL Outputs



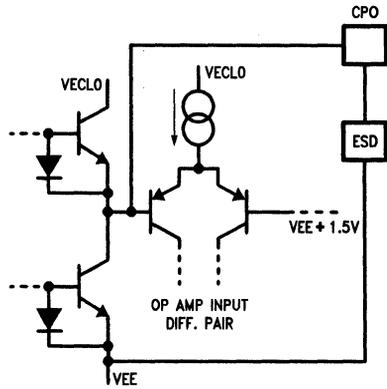
TL/F/8758-28

PHI1, PHI2 Outputs



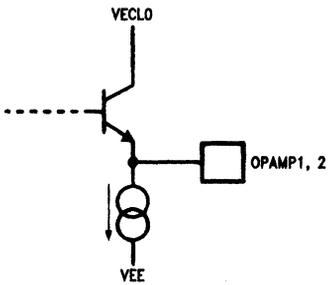
TL/F/8758-29

Charge Pump Output/Op Amp Input



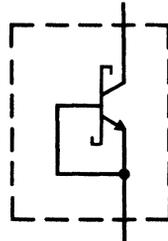
TL/F/8758-30

Op Amp Output



TL/F/8758-31

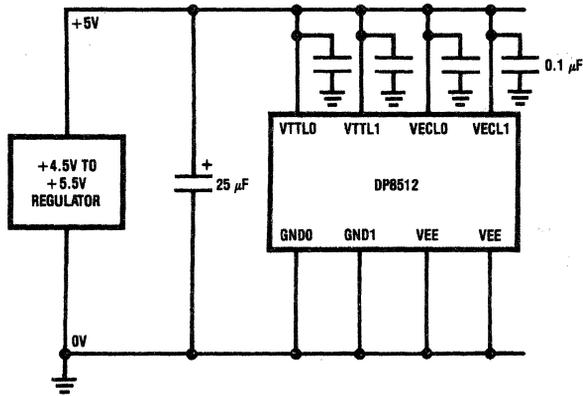
Typical ESD Circuit



TL/F/8758-33

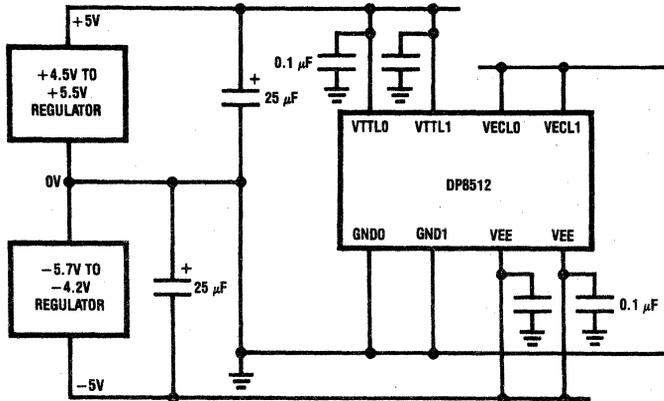
Typical Supply Wiring

Single Power Supply Operation



TL/F/8758-34

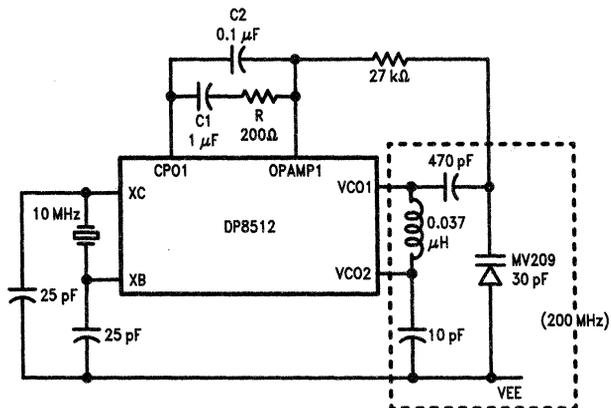
Dual Power Supply Operation



TL/F/8758-35

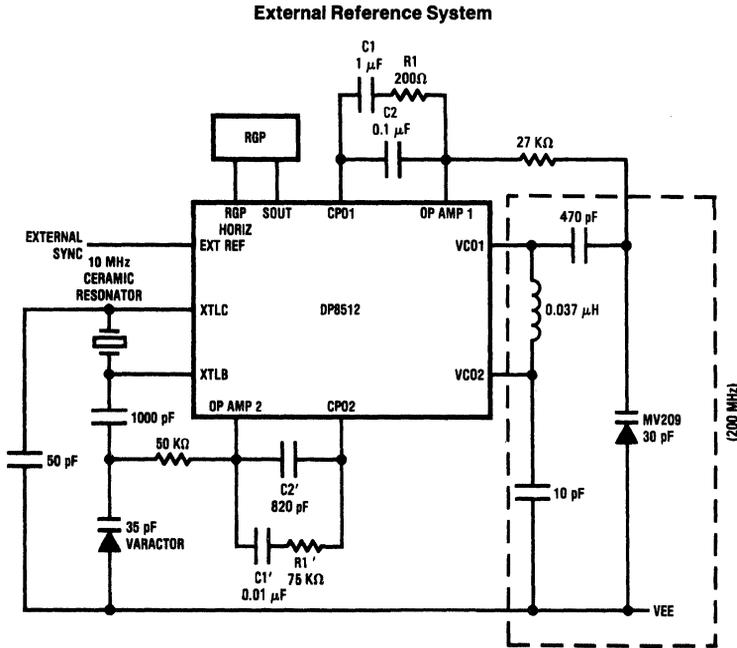
Typical Applications

Crystal Reference System



TL/F/8758-32

Typical Applications (Continued)



TL/F/8758-36

Loop Filter Calculations

Several constants need to be known in order to determine the loop filter components. They are the loop divide ratio N, the phase detector gain K_p , the VCO gain K_o , the loop bandwidth ω_o and phase margin ϕ .

The constant K_p is fixed at $80 \mu A/rad$ for the DP8513. N is simply the L counter modulus for the main loop. For the secondary loop, N is the S counter modulus times any external division between the SOUT pin and the RGP HORIZ pin. (i.e., if $S = 1$ and there is a $\div 100$ counter between SOUT and RGP HORIZ, $N = 1 \times 100 = 100$.) A 60° phase margin is recommended, however, the equations allow other values to be used if desired.

The oscillator gain constant of K_o can be obtained from Table III or determined experimentally. This is done by driving the 27k resistor which normally connects the varactor to the op amp output with an external power supply. Set the supply to $V_{EE} + 3V$ and note the PCLK frequency. Next set the supply to $V_{EE} + 2V$ and note the frequency again. The difference in these two frequencies (times 2π to convert to radians) is K_o . For optimum performance, the desired PCLK frequency should be somewhere between the two frequencies measured above. This may require adjustment of the coil.

Before choosing a value of ω_o , one fact should be pointed out. The 27k resistor and the 500 pF coupling capacitor between the coil and the varactor form a low pass filter with a cutoff of about 12 kHz. Thus, the loop bandwidth must be chosen to be less than this value. We recommend $2\pi \times 100$ Hz to $2\pi \times 3$ kHz for ω_o .

Having found all these constants, the following equations are used to find the component values:

$$R1 = \frac{1.08 N \omega_o}{K_p K_o} \quad C1 = \frac{3.46 K_p K_o}{N \omega_o^2} \quad C2 = \frac{0.27 K_p K_o}{N \omega_o^2}$$

To use a phase margin of other than 60° , use the following:

$$R1 = \frac{N \omega_o}{2 K_p K_o} (\text{cosec } \phi + 1)$$

$$C1 = \frac{2 K_p K_o}{N \omega_o^2} \tan \phi$$

$$C2 = \frac{K_p K_o}{N \omega_o^2} (\sec \phi - \tan \phi)$$

Example: Design a system with the following characteristics:

- External horizontal sync of 100 kHz
- 1560 pixels per line (2000 pixels including retrace)
- 20 bit wide video data
- 10 MHz processor rate

Note that this system will sync to an external source so that both loops must be used.

The PCLK frequency will be $100 \text{ kHz} \times 2000 \text{ pixels per line} = 200 \text{ MHz}$. The components in Table III will be used. Note that $K_o = 24 \text{ Mrad}$. Because it is a 20 bit wide system the L counter modulus must be 20.

By choosing $\omega_o = 2\pi \times 2800 \text{ Hz}$, the equations give $R1 = 200\Omega$, $C1 = 1.0 \mu F$ and $C2 = 0.08 \mu F$ (use $C2 = 0.1 \mu F$).

In the secondary loop, a ceramic resonator is used in place of a crystal to allow more pullability. Care should be taken when selecting a resonator to ensure its Q value is low. Its K_o is found experimentally to be 84 krads/V. The SOUT frequency will be the same as LCLK0 or $200 \text{ MHz} \div 20 = 10 \text{ Mhz}$. Thus, there must be a $\div 100$ counter between SOUT and RGP HORIZ so $N = 100$ for the secondary loop.

In choosing ω_o , it should be noted that ω_o for the secondary loop should be smaller than ω_o for the primary loop so that the main loop will be able to track the secondary without losing lock. Picking $\omega_o = 2\pi \times 750 \text{ Hz}$ gives $R1' = 75k$, $C1' = 0.01 \mu F$ and $C2' = 820 \text{ pF}$.

Loop Filter Calculations (Continued)

If a resonator is not available, a coil can be substituted. Since the tolerance of most coils is large, this will require that this coil be tuned to the center frequency of the internal oscillator. In this case, the varactor's capacitance needs to be chosen as a small percentage of the tanks total net capacitance in order to restrict its contribution to the variation in the reference frequency.

BOARD LAYOUT CONSIDERATIONS

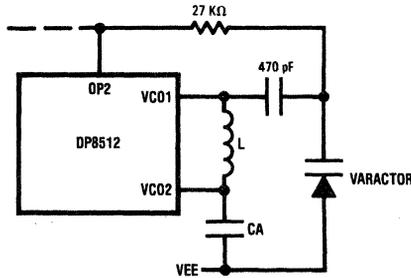
The VCO components/circuitry need to be isolated from such signals as PHI1, PHI2, and SOUT. These lines can produce sufficient noise to cause the DP8512's VCO or even an external VCO to attempt to injection lock to them.

If this problem is present there will be a low frequency oscillation (1-30 kHz) on top of the operational amplifier's DC tuning voltage. This oscillation indicates that the VCO is trying to lock to the phase transitions of the noise. The noise disrupts the ability of the VCO to provide a linear voltage to frequency translation and the loop will tend to wander.

Another test that can be done to verify the existence of this problem is to add a large value (1 MΩ) variable resistor between the CPO pin and ground. This resistor will cause the loop's reference and feedback phase transitions to shift relative to one another. Although these two signals' phase relationship is being varied, they should continue to be running, in lock, at the correct frequency. This should not result in any change in the operational amplifier's DC voltage. If the VCO is injection locking, the operational amplifier's DC voltage will be observed to vary as the resistor value is varied. Additionally it can be observed by comparing LCLK and SOUT simultaneously, the part will go from locking solidly with zero phase jitter to large phase steps. This will happen at several resistor settings (phase offsets).

The DP8512/13/14 Video Clock Generator Evaluation Board application note demonstrates an external VCO layout which minimizes the sensitivity to injection locking and also provides other helpful hints related to board layout.

Recommended VCO Components



TL/F/8758-37

$$F_{VCO} = \frac{1}{2\pi\sqrt{LC_{TOT}}}$$

$$C_{TOT} = \frac{1}{\frac{1}{CA} + \frac{1}{C_{VARACTOR}}}$$

TABLE III. Recommended VCO Components

Frequency (MHz)	L μH	TOKO Part #	CA pF	Cvaractor		Ko Mrad/V
				pF	Motorola #	
60	0.258	E 502HNS-6000026	56	30	MV209	16
80	0.17	E 502HNS-4000024	39	30	MV209	19
100	0.12	E 502HNS-3000023	30	30	MV209	21
120	0.07	E 502HNS-2000022	39	30	MV209	31
140	0.07	E 502HNS-2000022	22	30	MV209	27
160	0.07	E 502HNS-2000022	15	15	MV2205	27
180	0.07	E 502HNS-2000022	10	15	MV2205	26
200	0.037	E 502HNS-1000029	10	30	MV209	24
220	0.037	E 502HNS-1000029	10	15	MV2205	34

DP8513 Multi-Board Video Clock Generator

General Description

The DP8513 is a clock generator intended for use in medium- to high-performance CRT graphics systems. The device simplifies timing and minimizes phase skew between the various signals involved in the transfer of DRAM (or VDRAM) data into a DAC for display on a CRT. The DP8513 is used in conjunction with the DP8514 Crystal Clock Generator to simplify synchronization problems in multiboard systems. The device generates several synchronous clocks from a reference input using digital phase locked loop (PLL) techniques. These synchronous clocks include a graphics processor clock, a raster-scan pixel clock, and various gated TTL and ECL clocks required to transfer data from VRAM to video shift registers. Circuitry is also provided which enables the user to phase lock his graphics system to an external video source.

In a multiboard system the REFIN and REFCLK inputs enable the motherboard and the slave boards to be synchronously driven from a single master clock source such as the DP8514.

The graphics processor clocks (PHI1 and PHI2) are non-overlapping two-phase clocks with MOS-compatible outputs capable of driving 100 pF loads at up to a 20 MHz rate.

The raster-scan pixel clock (PCLK) and the parallel load clock (LCLK3) required for video shift register operation are generated from the REFIN and REFCLK inputs using two

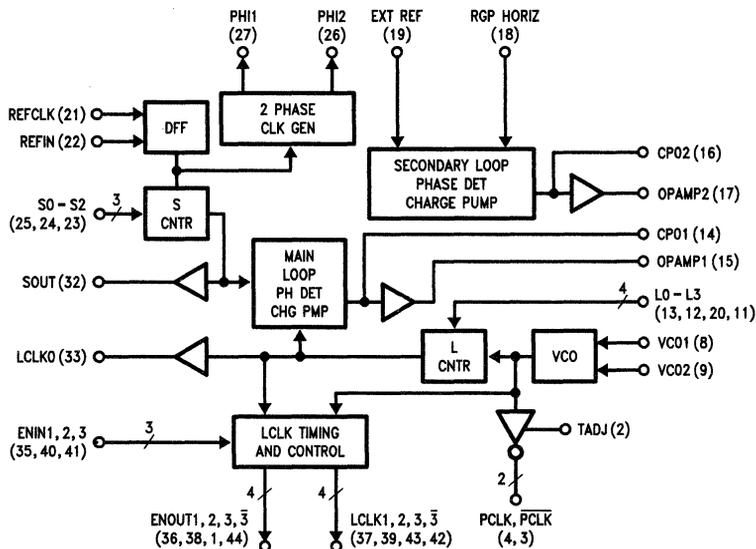
programmable counters in a phase locked loop configuration. Video word widths from 4 to 64 bits in increments of 4 are accommodated by these counters. The ECL-compatible PCLK outputs are capable of operating at up to 225 MHz from either a positive or a negative supply voltage.

Two gated TTL clock outputs (LCLK1 and LCLK2) are also provided to enable easy transfer of data from a VRAM serial shift register directly into a video shift register or indirectly through a FIFO.

Features

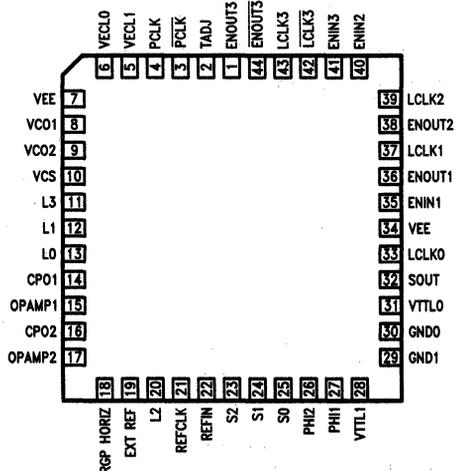
- Phase-locked-loop generates synchronized system clock, PCLK, and LCLK
- MOS-compatible single-phase or non-overlapping two-phase system clock output
- 225 MHz ECL differential output pixel clock (PCLK)
- Gated and non-gated load clock (LCLK) outputs ease VRAM-to-VSR synchronization
- Accommodates video word widths from 4 to 64 bits in increments of 4
- Timing Adjust pin (TADJ) provides a fixed offset adjustment of PCLK to LCLK to ease system design
- ECL circuitry can be referenced to positive or negative power supply
- Enables horizontal synchronization from an external source

Block Diagram (numbers in parentheses indicate pin numbers)



TL/F/9283-1

Connection Diagram



TL/F/9283-2

Order Number DP8513V
See NS Package Number V44A

Pin Descriptions

- 1, 44 **ENOUT3, ENOUT3:** Differential ECL video enable outputs synchronous to LCLK0 and gated by ENIN3 (10k and 100k ECL compatible). Require conventional ECL 50Ω termination.
- 2 **TADJ:** Timing Adjust pin allows the PCLK output transition to occur 2.5 ns earlier than LCLK3 and ENOUT3 when taken to a high logic level. This allows the set-up and hold times of various Video Shift Registers to be accommodated. A logic high is VECL0, and a logic low is VEE.
- 3, 4 **PCLK, PCLK:** Differential ECL pixel clock outputs driven by the VCO in the main loop (10k and 100k ECL compatible). Require conventional ECL 50Ω termination.
- 5 **VECL1:** ECL output buffer positive power supply.
- 6 **VECL0:** ECL internal logic positive power supply.

Note: VECL0 and VECL1 can be operated from 0V to +5.7V relative to GND.

- 7, 34 **VEE:** ECL negative power supply.
- 8, 9 **VC01, VC02:** External tank circuit connections for the Pierce VCO. See typical applications for typical wiring.
- 10 **VCS:** No connection required.
- 11, 12, 13, 20 **L3, L1, L0, L2:** A four bit word input used to select the L Counter modulus. Any modulus from 4 to 64 may be selected in increments of 4. L0 is the least significant bit. High = VECL0, Low = VEE. These pins should be bypassed to VECL0 with 0.01 μF caps. These traces should not be allowed to cross any signal traces.
- 14 **CP01:** Main loop charge pump output. Used in conjunction with OPAMP1 to form the external loop filter.
- 15 **OPAMP1:** Op amp output of the main loop. This output is used to control the pixel clock frequency via the varactor diodes in the LC tank circuit.
- 16 **CP02:** Charge pump output of the secondary loop. Used in conjunction with OPAMP2 to form the external loop filter when synchronization to an external video source is desired.
- 17 **OPAMP2:** Op amp output of the secondary loop. This output is used to vary the crystal frequency (VCXO) in systems where it is desired to lock to an external video source.
- 18 **RGP HORIZ:** TTL compatible secondary loop phase comparator input. This signal completes the feedback path from the VCXO by way of the DP8500 Raster Graphics Processor's Horizontal Output. The falling edge triggers the secondary phase detector.
- 19 **EXT REF:** TTL compatible external horizontal reference input. This is the optional horizontal input for systems where it is desired to lock to an external video source. The falling edge triggers the secondary phase detector.
- 21, 22 **REFCLK, REFIN:** Reference source inputs from which all output clocks are generated. See the typical applications for typical wiring.
- 23, 24, 25 **S2, S1, S0:** TTL compatible three bit word that determines the S Counter modulus. S0 is the least significant bit.

Pin Descriptions (Continued)

- 26, 27** **PHI2, PHI1:** MOS compatible two-phase non-overlapping clocks. The frequency of these clocks is that of the REFIN input. Use of these outputs to drive large capacitive loads requires the use of high frequency bypass caps across VTTL1 and GND1 as close to the part as possible.
- 28** **VTTL1:** TTL output buffer supply. Specified for $5V \pm 10\%$ operation.
- 29** **GND1:** TTL output buffer supply return.
- 30** **GND0:** TTL internal logic power supply return.
- 31** **VTTL0:** TTL internal logic positive power supply. Specified for $5V \pm 10\%$ operation.
- 32** **SOUT:** TTL compatible ungated output of the S Counter. It is also connected to one of the inputs of the main loop phase comparator.
- 33** **LCLK0:** TTL compatible free-running Load Clock. This signal is also connected to an input of the main loop phase comparator.
- 35** **ENIN1:** TTL compatible video enable input. A high on this input starts LCLK1 on the next positive transition of LCLK0.
- 36** **ENOUT1:** TTL compatible video enable output synchronous to LCLK0 and gated by ENIN1.
- 37** **LCLK1:** TTL compatible load clock equivalent to LCLK0 but gated by ENIN1.
- 38** **ENOUT2:** TTL compatible video enable output synchronous to LCLK0 and gated on the third positive transition of LCLK0 following a valid ENIN2 input.
- 39** **LCLK2:** TTL compatible load clock equivalent to LCLK0 but gated by ENIN2.
- 40** **ENIN2:** TTL compatible video enable input. A high on this input starts LCLK2 on the next positive transition of LCLK0.
- 41** **ENIN3:** TTL compatible video enable input. A high on this input starts LCLK3 on the next positive transition of LCLK0.
- 42, 43** **LCLK3 LCLK3:** Differential ECL compatible load clock synchronous to LCLK0 and gated by ENIN3 (10k and 100k ECL compatible). Require conventional ECL 50 Ω termination.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature	-65°C to +150°C
TTL Signals	
Inputs	7.0V
Outputs	7.0V

ECL Signals	
Output Current	-50.0 mA
Supply	
VEE to GND	-7V to 0.5V
VECL to GND	-0.5V to 7V
VTTL to GND	-0.5V to 7V
ESD susceptibility (see Note 5)	1000V

Recommended Operating Conditions (Notes 1, 2, 3, 4)

Symbol	Parameter	Min	Typ	Max	Units
VTTL0, 1 to GND	TTL Power Supply	4.5		5.5	V
VECL0, 1 to VEE	ECL Power Supply	4.2		5.7	V
VEE to GND		-5.7		0	V
V _{IH}	High Level Input Voltage TTL Inputs	2			V
V _{IL}	Low Level Input Voltage TTL Inputs			0.8	V
I _{OH}	High Level Output Current TTL Outputs			-0.4	mA
	High Level Output Current MOS Compatible Outputs			-0.4	mA
I _{OL}	Low Level Output Current TTL Outputs			8	mA
	Low Level Output Current MOS Compatible Outputs			20	mA
FPCLK (Note 6)	Pixel Clock Max Frequency			225	MHz
F _{in} MAX	REFCLK Frequency, REFIN = fREFCLK/2			50	MHz
T _{su1}	Setup Time ENIN1 to LCLK0	20	11		ns
T _{su2}	Setup Time ENIN2 to LCLK0	20	11		ns
T _{su3}	Setup Time ENIN3 to LCLK0	20	11		ns
T _{su4}	Setup Time REFIN to REFCLK 	5			ns
T _{h1}	Hold Time LCLK0 to ENIN1	0	-9		ns
T _{h2}	Hold Time LCLK0 to ENIN2	0	-9		ns
T _{h3}	Hold Time LCLK0 to ENIN3	0	-9		ns
T _{h4}	Hold Time REFCLK  to REFIN	2			ns
T _{ambient}	Operating Temp. Range	0		70	°C

Note 1: See Timing Waveforms for relevant signal edges (positive or negative) from which all setup and hold times measurements are made.

Note 2: TTL inputs; ENIN1, 2, 3, RGP HORIZ, EXT REF, S0, S1, S2, REFIN, REFCLK

Note 3: TTL outputs; SOUT, LCLK0, 1, 2, ENOUT1, 2, MOS outputs; PHI1, 2

Note 4: Inputs L0, L1, L2, L3, and TADJ designed to be tied to VECL for high level or shorted to VEE (or left open) for low level. See input schematics.

Note 5: Human body model; 120 pF thru 1.5 kΩ.

Note 6: Maximum frequency that PCLK can reliably be "locked". The VCO range should be controlled to avoid exceeding 235 MHz with the maximum control voltage applied to the VCO.

DC Electrical Characteristics (Note 1)

Symbol	Parameter	Conditions		Min	Typ	Max	Units	
V_{IC}	Input Clamp Voltage	VTTL0, 1 = 4.5V, $I_{in} = -18$ mA				-1.5	V	
V_{OH}	Output High Voltage	VTTL0, 1 = 4.5V to 5.5V	TTL Outputs, $I_{OH} = -400$ μ A	VTTL-2			V	
			MOS Outputs	$I_{OH} = -100$ μ A	VTTL-0.4			
				$I_{OH} = -400$ μ A	VTTL-2.3			
		VECL0, 1 = 0V VEE = -4.2V	ECL Outputs, 50 Ω load to -2V	-1135		-880	mV	
		OPAMP Output, $I_{OH} = -1.25$ ICPO Sink	VECL0-1.2			V		
V_{OL}	Output Low Voltage	VTTL0, 1 = 4.5V	TTL Outputs, $I_{OL} = 8$ mA			0.5	V	
			MOS Outputs	$I_{OL} = 100$ μ A				0.4
				$I_{OL} = 20$ mA				0.5
		VECL0, 1 = 0V VEE = -4.2V	ECL Outputs, 50 Ω Load to -2	-1995		-1490	mV	
		OPAMP Output, $I_{OL} = -1.25$ ICPO Source			VEE + 0.5	V		
I_I	Max High Level Input Current	VTTL0, 1 = 5.5V, TTL Inputs, $V_{in} = 7$ V				100	μ A	
I_{IH}	High Level Input Current	VTTL0, 1 = 5.5V, TTL Inputs, $V_{in} = 2.7$ V				20	μ A	
I_{IL}	Low Level Input Current	VTTL0, 1 = 5.5V, TTL Inputs, $V_{in} = 0.4$ V				-200	μ A	
I_O	Output Drive Current	VTTL0, 1 = 5.5V	TTL Outputs, $V_o = 2.25$ V	-30		-110	mA	
			MOS Outputs, $V_o = 2.25$ V		-135		mA	
I_{CPO}	Charge Pump Current	VEE = -4.2V to -5.7V VECL0, 1 = 0V	Source	-0.2	-0.5	-1.0	mA	
			Sink	0.2	0.5	1.0	mA	
			TRI-STATE®	-10	0	10	μ A	
I_{CC}	Supply Current	VTTL0, 1 = 5.5V			30	40	mA	
		VECL0, 1 = 0V	TTL SUPPLY					
			VEE = -5.7V, 10k ECL Supply			150		210
		VEE = -4.8V, 100k ECL Supply			135	185		

Note 1: TTL Inputs; ENIN1, 2, 3 RGP HORIZ, EXT REF, S0, S1, S2, REFIN, REFCLK

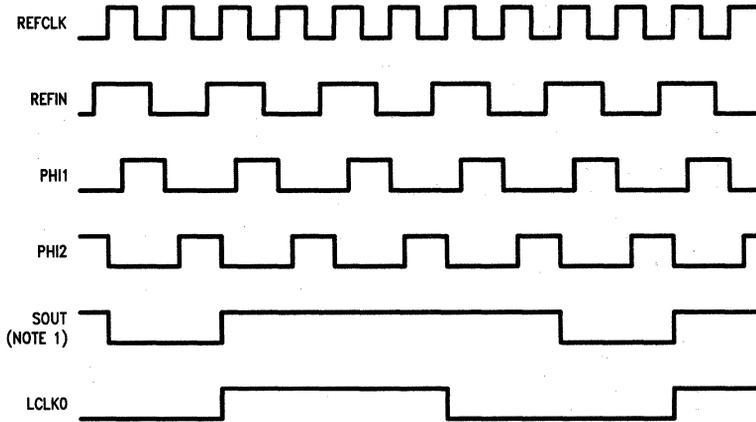
TTL Outputs; ENOUT1, 2, 3, SOUT

MOS Outputs; PHI1, PHI2

ECL Outputs; ENOUT3, ENOUT3, PCLK, PCLK, LCLK3, LCLK3

Functional Waveforms

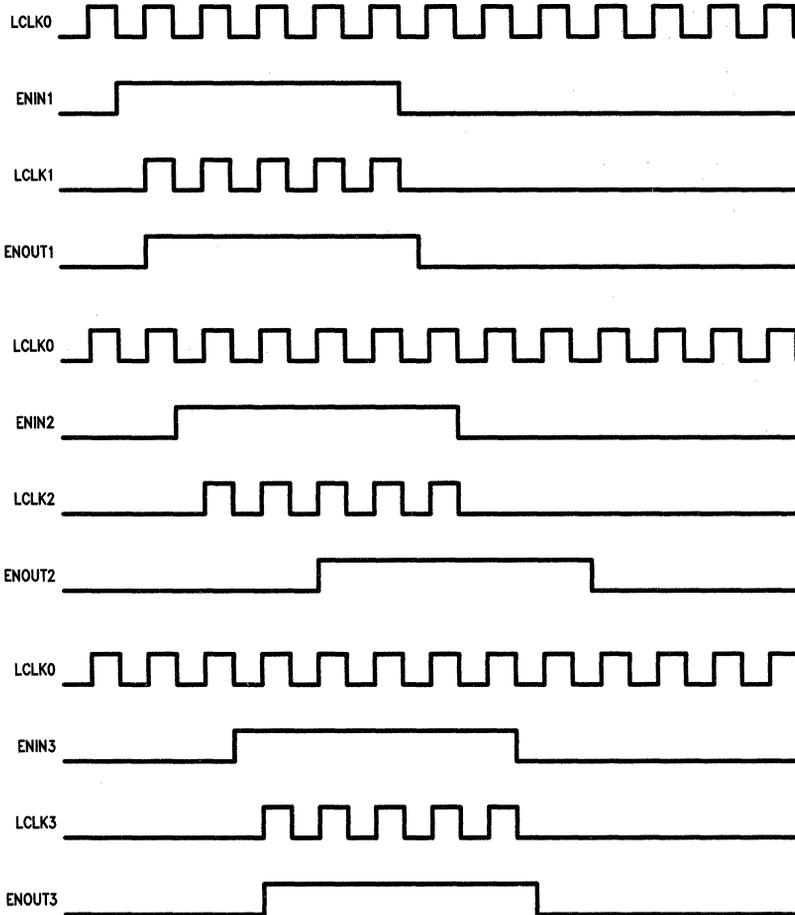
DP8513 Functional Waveforms



TL/F/9283-17

Note 1: SOUT waveform displayed with S Counter in divide-by-4 mode.

Load Clock Timing Diagram



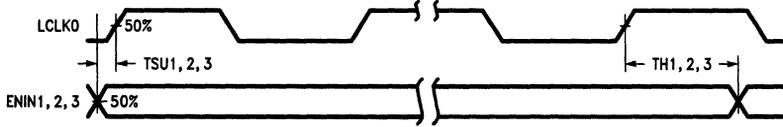
TL/F/9283-18

TL/F/9283-19

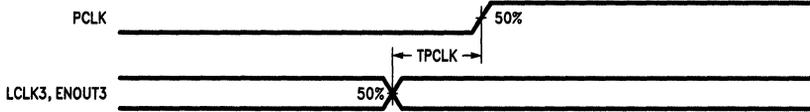
TL/F/9283-20

AC Timing Waveforms

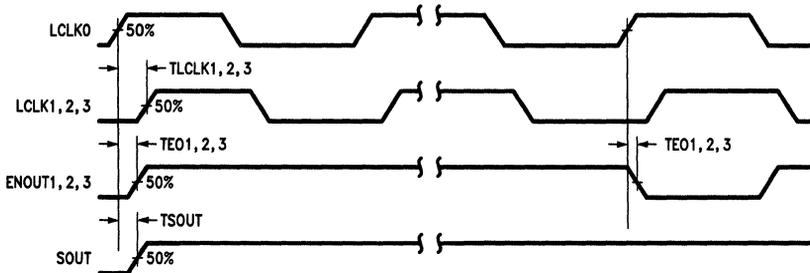
TSETUP, THOLD
LCLK0 to ENIN1, 2, 3



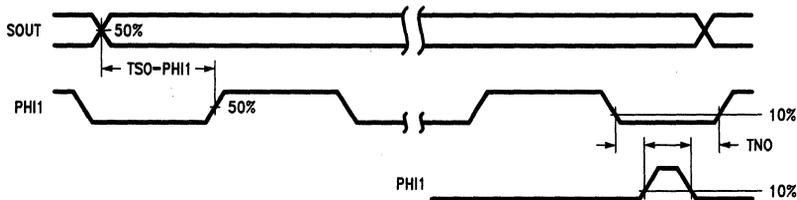
TL/F/9283-13



TL/F/9283-14



TL/F/9283-15



TL/F/9283-16

AC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
FMAX	Max VCO Freq (Note 2)		235			MHz
FMAX	Max REFCLK Freq		50			MHz
PW	Pulse Width High PHI1, 2	REFCLK = 40 MHz, REFIN = 20 MHz		21		ns
TEO1	LCLK0 to ENOUT1		-7	-2	5	ns
TEO2	LCLK0 to ENOUT2		-7	-2	5	ns
TEO3	LCLK0 to ENOUT3		-15	-7	0	ns
TPCLK	LCLK3, ENOUT3 to PCLK	TADJ = HI (VECL0)	0	2.5	5.0	ns
		TADJ = LOW (VEE)	-2.5	0	+2.5	ns
TLCLK1	LCLK0 to LCLK1		-5	0	5	ns
TLCLK2	LCLK0 to LCLK2		-5	0	5	ns
TLCLK3	LCLK0 to LCLK3		-15	-7	0	ns
T _{PHI1}	LCLK0 to PHI1	Loop Locked REFIN = 20 MHz S = 8		-5		ns
T _{SOUT}	LCLK0 to SOUT	Loop Locked REFIN = 20 MHz S = 8		0		ns
T _{SO-PHI1}	SOUT to PHI1		-15	-5	5	ns
TNO	Nonoverlap Time PHI1 to PHI2	C _L = 50 pF		0		ns
TR, TF	Rise, Fall Time PHI1, PHI2	(Note 1)		4		ns

Note 1: Rise and Fall times measured from 0.5V to VTTL1 - 2V with C_L = 50 pF

Note 2: This is not production tested but is assured by characterization to include sufficient margin beyond processing extremes.

Circuit Operation

The DP8513 Video Clock Generator Slave generates several clock signals required in a video display system from two reference frequency inputs: REFCLK and REFIN. These signals are provided by the DP8514 Crystal Clock Generator. The 2X REFCLK input is used as a resynchronizing clock for the 1X REFIN input thus eliminating duty cycle distortion introduced by the backplane. Among the clocks generated are a two-phase clock for driving the RGP and BPU processors, ECL pixel and load clocks (PCLK and LCLK3) for high speed video shift register parallel load and shift operations, and TTL load clocks (LCLK0, LCLK1, and LCLK2) for moving DRAM and FIFO data to the video shift registers. The LCLK and PCLK outputs are all internally synchronized in order to simplify system timing.

The two-phase graphics processor clock (PHI1 and PHI2) operates at the frequency of the REFIN input. It is capable of directly driving the raster graphics and BIT-BLT processors. The two-phase clock is closely aligned with the other

clocks generated by the device to maximize system operation. The PCLK and LCLK outputs are generated using a digital phase locked loop as shown in *Figure 1*.

The loop consists of the S and L counters, a phase comparator, and a voltage controlled oscillator (VCO) with the relationship between these elements in the loop defined as:

$$PCLK = \frac{REFIN \times L}{S}$$

where PCLK is the pixel clock frequency, REFIN is the REFIN frequency, L is the L Counter modulus, and S is the S Counter modulus. When the frequency of the VCO (PCLK) in the phase locked loop is stable the inputs to the phase detector are in phase, thus the S Counter and L Counter outputs are identical in both phase and frequency. As long as the REFCLK input is constant, the phase and frequency of the S Counter output remains constant. Any drift, or change in frequency, of the VCO will be divided down and appear as a shift in phase at the L Counter output. The

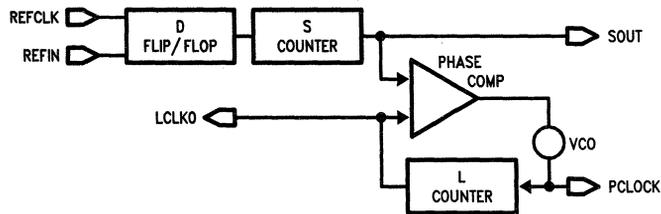


FIGURE 1. PLL Block Diagram

TL/F/9283-3

Circuit Operation (Continued)

phase detector will sense this phase error and generate a correction voltage for the VCO input which is proportional to the magnitude of the frequency error. This correction voltage will change the VCO frequency to eliminate the error and keep the loop locked by changing the capacitance of the varactor in the LC oscillator tank circuit. The varactor's capacitance is proportional to the amount of reverse bias applied across it. The correction voltage is provided by the OPAMP output which has a 3V typical operating range (VEE +0.5V to VECL0 -1.2V). This operating range provides a ±10% (typical) VCO frequency range. This restricts the use of multiple pixel rates. However, an external OPAMP can be added to extend the tuning voltage range or tank circuit components can be bandswitched in.

The presence of the S Counter in the loop enables the graphics processor to operate at full speed independent of PCLK frequency. The video shift register's parallel data width determines the L Counter modulus. An 8-bit parallel shift register would use an L Counter modulus of 8 so that a parallel load pulse occurs once every 8 pixel clocks. The L Counter output is used to derive the four LCLK outputs. These 4 LCLKs differ slightly in format to allow for the various system configurations highlighted in the following section.

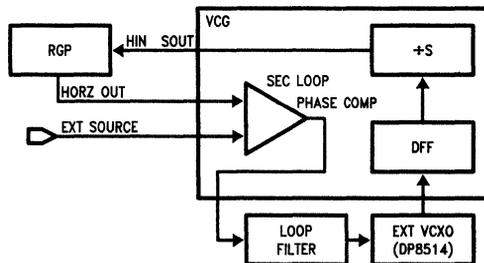
The ECL LCLK3 output is used in conjunction with the PCLK output to load data into a high-speed video shift register. The PCLK provides the clock and LCLK3 provides the load signal for the shift register. With a typical video shift register operating at frequencies around 200 MHz timing can become extremely critical. For this reason a Timing Adjust pin (TADJ) is provided that will allow the user to obtain the optimum LCLK3-to-PCLK setup and hold timing relationship. See Pin Definitions for further info.

The other three LCLK outputs (LCLK0, LCLK1, LCLK2) are TTL outputs. They can be used to control a selection of different DRAMs/video shift register configurations as shown in the system architecture section which follows.

Also included on the chip is a secondary phase locked loop which can be used to synchronize the graphics system to an external signal such as the horizontal sync pulse from a television broadcast. A block level diagram of this mode of operation is shown in *Figure 2*. An external oscillator (such as the DP8514) is configured to operate as a voltage controlled crystal oscillator (VCXO). Any change in frequency of the VCXO forces the outputs of the S Counter and the

RGP's horizontal counter to shift in phase relative to the external horizontal reference signal. The auxiliary phase comparator senses any phase difference between its two inputs and produces a correction voltage for the VCXO which is proportional to the amount of the frequency error.

This correction voltage will change the VCXO frequency to eliminate the error thus keeping the loop locked.



TL/F/9283-4

FIGURE 2

For most applications the REFIN input (and thus PHI1 and PHI2) will be in the range of 20 MHz. With a typical 8-bit system this means that the PCLK output will be running at 20 MHz × 8 bits = 160 MHz. However if the system is 16 bits wide instead of 8 bits the PCLK frequency would become 20 MHz × 16 bits = 320 MHz which is beyond the range of the VCO. Therefore the S Counter block must be added to divide the REFIN input frequency down to a more manageable frequency. Using the equation and the above example of a 20 MHz REFIN input with 16 bits of data, the S Counter is used as a divide-by-two counter to get a PCLK frequency of 160 MHz.

The S Counter can be programmed to divide by any integer up to 8 and the L Counter can be programmed for any word width from 4 bits to 64 bits in increments of 4. Table I shows some of the frequencies possible using various values for the S- and L-modulus.

VARIOUS SYSTEM ARCHITECTURES

Figure 4 demonstrates the DP8513 in a system using a video shift register and a DRAM. Another possible application uses a video DRAM or VDRAM, in place of the DRAM, as shown in *Figure 5*. This system differs from *Figure 4* in that the VDRAM contains an internal shift register which allows

TABLE I. Partial Table of PCLK Frequencies

S MOD	8-Bit Word (L = 8)		16-Bit Word (L = 16)		32-Bit Word (L = 32)	
	PCLK Frequency		PCLK Frequency		PCLK Frequency	
	REFIN = 10 MHz	REFIN = 20 MHz	REFIN = 10 MHz	REFIN = 20 MHz	REFIN = 10 MHz	REFIN = 20 MHz
1	80.0 MHz	160.0 MHz	160.0 MHz	na	na	na
2	40.0	80.0	80.0	160.0 MHz	160.0 MHz	na
3	26.7	53.3	53.3	106.7	106.7	na
4	20.0	40.0	40.0	80.0	80.0	160.0 MHz
5	16.0	32.0	32.0	64.0	64.0	128.0
6	13.3	26.7	26.7	53.3	53.3	106.7
7	11.4	22.9	22.9	45.7	45.7	91.4
8	10.0	20.0	20.0	40.0	40.0	80.0

Circuit Operation (Continued)

memory to be randomly accessed while data is being output to the video shift registers through the serial port. The output of the VDRAM differs from the standard DRAM in that data is not valid at the output of the internal shift register until an SCLK pulse is provided. The offset in the LCLK3 waveform from LCLK1 is obtained by connecting ENOUT1 to ENIN3.

The third type of system employs a video shift register (such as the DP8515) that contains an onboard FIFO as shown in *Figure 6*. This architecture simplifies timing requirements in a high-speed multiboard system. The propagation delay time from the DRAM to the video shift register through the back plane is no longer restricted to be less than one LCLK period. The VDRAM can be writing into the FIFO asynchronous to the data being loaded into the shift register. As in the previous systems, one LCLK1 pulse transfers the data to the VDRAM output. LCLK2 then writes several words into the FIFO prior to the LCLK3 reading out the first word. The connections required to obtain this timing are shown in the diagram.

To implement a multiboard system the DP8513 is used in conjunction with the DP8514 and a Video Shift Register with a FIFO (such as the DP8515/16). The various LCLK signals from the DP8513 control the VRAM shifting and FIFO read and write operations to minimize the problems associated with backplane delays.

Typically the main board will contain a DP8513 Video Clock Generator Slave as well as a DP8514 Crystal Clock Generator, the DP8515/16 Video Shift Registers (one per plane), and a main graphics processor (such as the DP8500). See the multiboard diagram which follows. The other boards will contain one or more planes of memory and the associated BIT-BLT processors (such as the DP8510), and one DP8514 to generate the two-phase clock for the processors.

The 2X and 1X clocks (XOUT and DATA OUT) provided by the DP8514 on the main board are used by the DP8513 to generate non-overlapping clocks for the graphics proces-

sor. These 2x and 1x signals are also sent across the backplane to the memory boards where other DP8514's generate non-overlapping clocks for the BIT-BLT processors. The display of information is started by an ENABLE signal from the RGP to the ENIN1 of the DP8513. The remaining ENIN and ENOUT pins are connected as shown in *Figure 6* along with the resulting LCLK waveforms. LCLK3 and PCLK are used to read data from the FIFO and shift it to the DAC's. All of the VSR's must be on the main board to keep the 225 MHz PCLK signal off the backplane. LCLK1 is sent through the backplane to the SCLK inputs of all the VRAMs and causes data to be shifted out to the VSR's FIFO's. LCLK2 is sent through the backplane to a buffer and then through the backplane again to the WR inputs of the FIFO's. These two backplane delays are used to cancel the back plane delays from the DP8513's LCLK1 pin to the VRAM's SCLK pin and from the VRAM's output to the FIFO's input.

Following the path from the LCLK2 pin of the DP8513 to the WR pin of the VSR, we find two backplane delays and a buffer prop delay which is assumed to be negligible in the following discussion. Note from *Figure 6* that the LCLK3 occurs three LCLK periods after LCLK2. This means that two backplane delays can be as much as three LCLK periods, or $T_{BP} < 3/2 T_{LCLK}$ for no loss of data. To increase this time further one or more D Flip Flops may be inserted between ENOUT2 and ENIN3 as shown in *Figure 3*. Each flip flop will increase the allowable backplane delay by approximately 1/2 LCLK period.

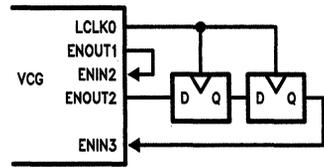
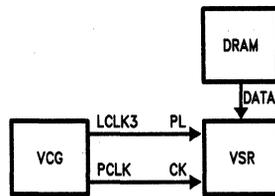


FIGURE 3

TL/F/9283-5



TL/F/9283-6



FIGURE 4. System Configuration and LCLK Waveforms Using DRAM

TL/F/9283-7

Circuit Operation (Continued)

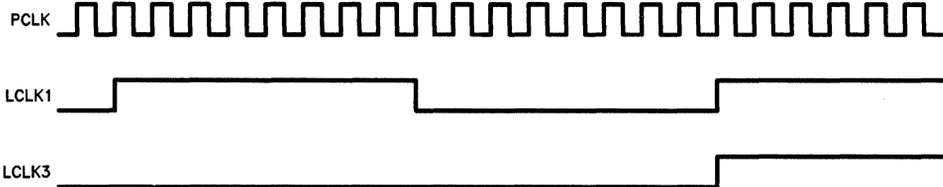
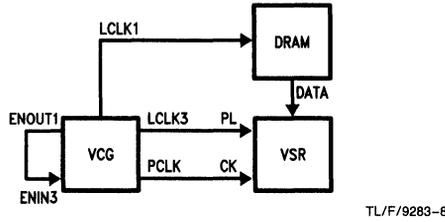


FIGURE 5. System Configuration and LCLK Waveforms Using VDRAM

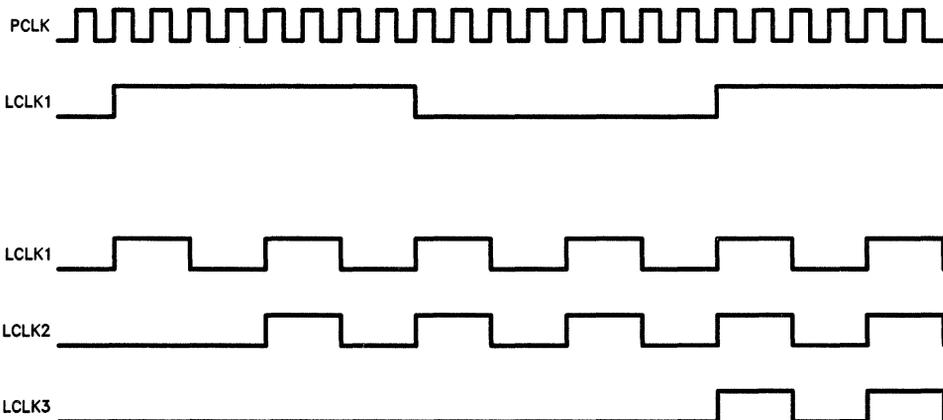
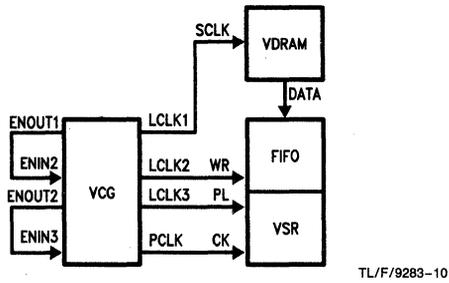
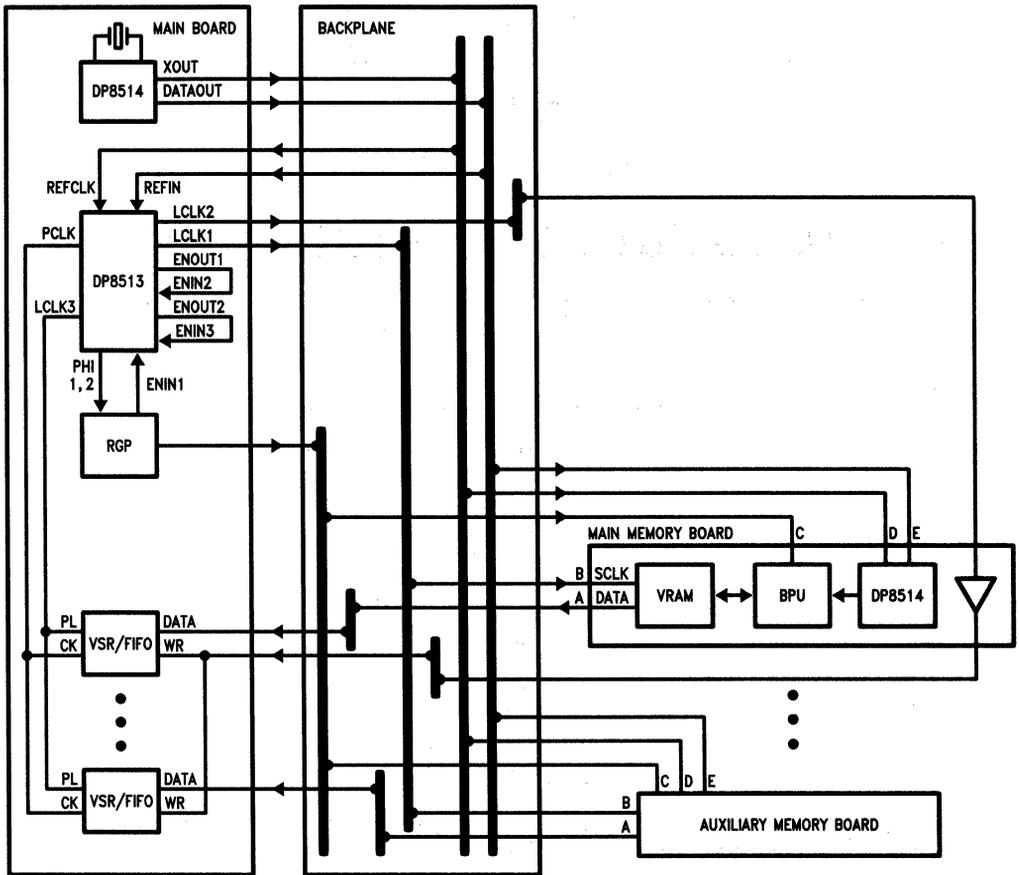


FIGURE 6. System Configuration and LCLK Waveforms Using a VSR Containing FIFO

Multi-Board Diagram



TL/F/9283-12

TABLE II. Counter Modulus Tables

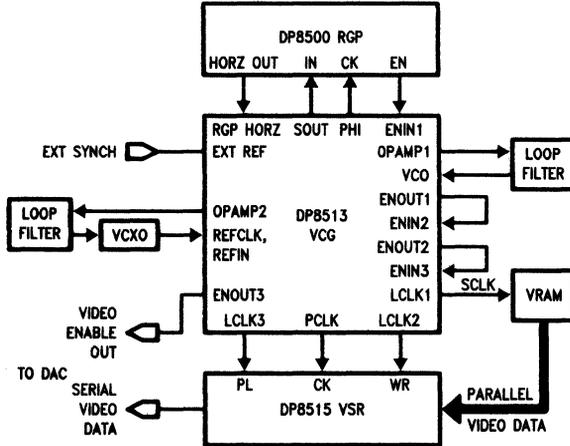
S MOD	S Counter Inputs		
	S2	S1	S0
1	L	L	L
2	L	L	H
3	L	H	L
4	L	H	H
5	H	L	L
6	H	L	H
7	H	H	L
8	H	H	H

L = TTL Logic Zero (GND0, 1)
 H = TTL Logic One (VTTL0, 1)

L MOD	L Counter Inputs			
	L3	L2	L1	L0
4	0	0	0	0
8	0	0	0	1
12	0	0	1	0
16	0	0	1	1
20	0	1	0	0
24	0	1	0	1
28	0	1	1	0
32	0	1	1	1
36	1	0	0	0
40	1	0	0	1
44	1	0	1	0
48	1	0	1	1
52	1	1	0	0
56	1	1	0	1
60	1	1	1	0
64	1	1	1	1

0 = VEE
 1 = VECL0, 1

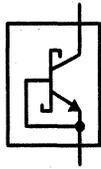
Typical System Diagram



TL/F/9283-21

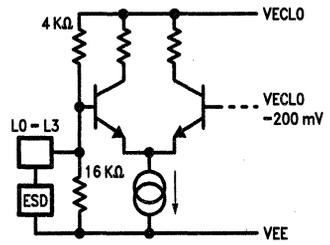
Input Schematics

Typical ESD Circuit



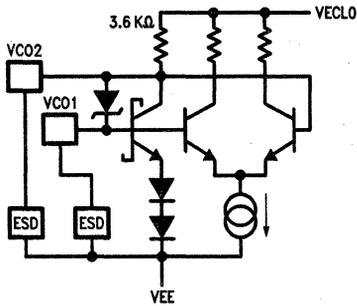
TL/F/9283-22

L0-L3 Inputs



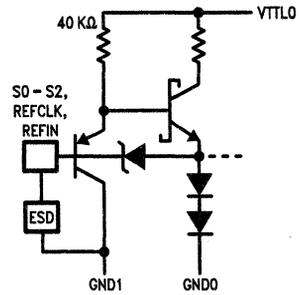
TL/F/9283-23

VCO Inputs



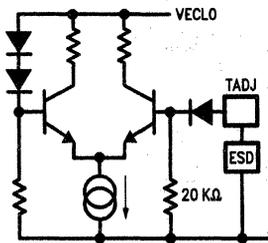
TL/F/9283-24

TTL Inputs
S0-S2, REFCLK, REFIN



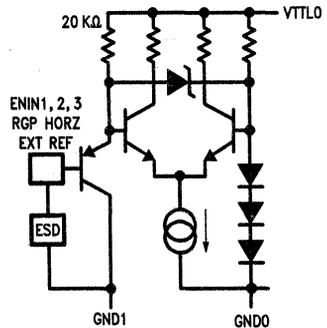
TL/F/9283-25

Timing Adjust Input



TL/F/9283-26

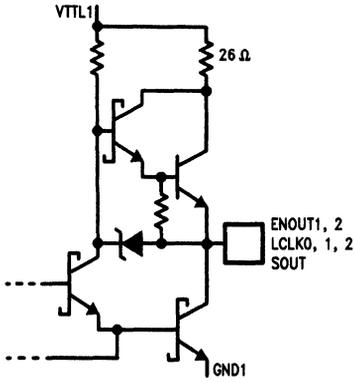
TTL Inputs
(ENIN1, 2, 3, RGP HORZ, EXT REF)



TL/F/9283-27

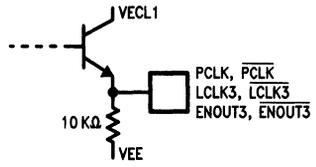
Output Schematics

TTL Outputs
(ENOUT1, 2, LCLK0, 1, 2, SOUT)



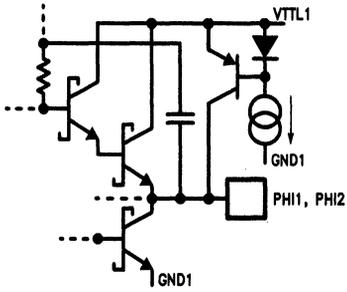
TL/F/9283-28

ECL Outputs



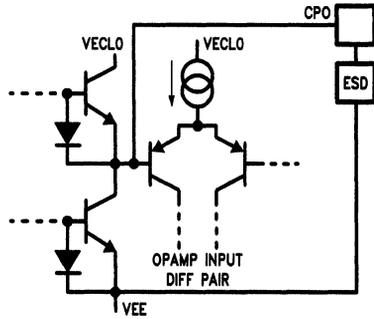
TL/F/9283-29

PHI1, PHI2



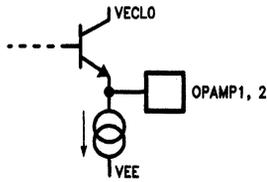
TL/F/9283-30

Charge Pump Output/OP Amp Input



TL/F/9283-31

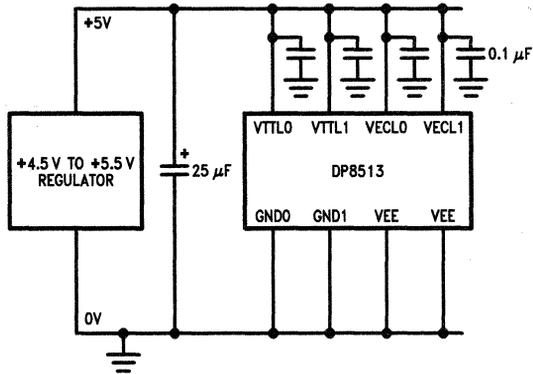
Op Amp Output



TL/F/9283-32

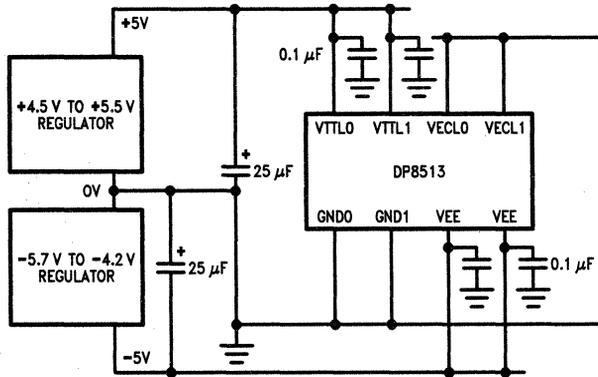
Typical Power Connections

Single Power Supply Operation



TL/F/9283-33

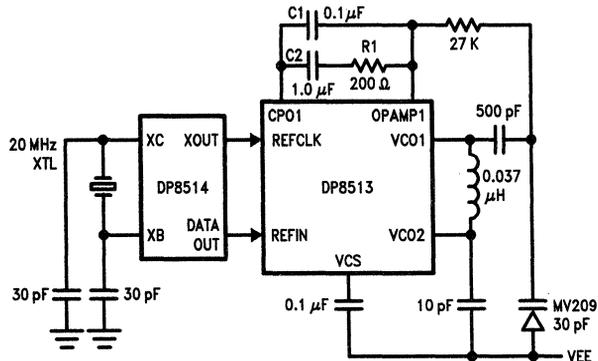
Dual Power Supply Operation



TL/F/9283-34

Typical Applications

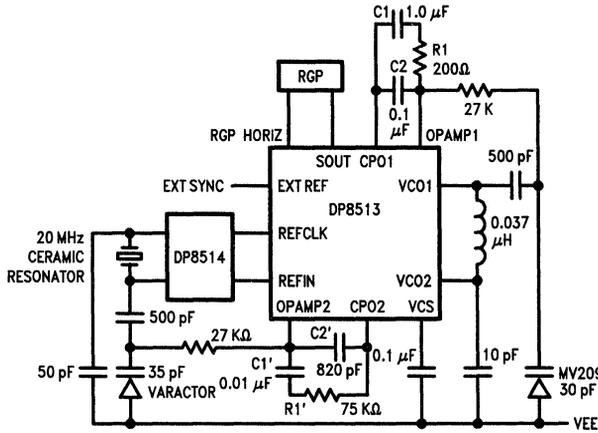
Crystal Referenced System
PCLK = 200 MHz, L = 20, S = 1



TL/F/9283-35

Typical Applications (Continued)

External Referenced System
PCLK = 200 MHz, L = 20, S = 1



TL/F/9283-36

Loop Filter Calculations

Several constants need to be known in order to determine the loop filter components. They are the loop divide ratio N , the phase detector gain K_p , the VCO gain K_o , the loop bandwidth ω_o , and phase margin ϕ .

The constant K_p is fixed at $80 \mu A/rad$ for the DP8513. N is simply the L counter modulus for the main loop. For the secondary loop, N is the S counter modulus times any external division between the SOUT pin and the RGP HORIZ pin. (i.e., if $S = 1$ and there is a $\div 100$ counter between SOUT and RGP HORIZ, $N = 1 \times 100 = 100$.) A 60° phase margin is recommended, however, the equations allow other values to be used if desired.

The oscillator gain constant of K_o can be obtained from Table III or determined experimentally. This is done by driving the 27k resistor which normally connects the varactor to the op amp output with an external power supply. Set the supply to $V_{EE} + 3V$ and note the PCLK frequency. Next set the supply to $V_{EE} + 2V$ and note the frequency again. The difference in these two frequencies (times 2π to convert to radians) is K_o . For optimum performance, the desired PCLK frequency should be somewhere between the two frequencies measured above. This may require adjustment of the coil.

Before choosing a value of ω_o , one fact should be pointed out. The 27k resistor and the 500 pF coupling capacitor between the coil and the varactor form a low pass filter with a cutoff of about 12 kHz. Thus, the loop bandwidth must be chosen to be less than this value. We recommend $2\pi \times 100$ Hz to $2\pi \times 3$ kHz for ω_o .

Having found all these constants, the following equations are used to find the component values:

$$R1 = \frac{1.08 N \omega_o}{K_p K_o} \quad C1 = \frac{3.46 K_p K_o}{N \omega_o^2} \quad C2 = \frac{0.27 K_p K_o}{N \omega_o^2}$$

To use a phase margin of other than 60° , use the following:

$$R1 = \frac{N \omega_o}{2 K_p K_o} (\text{cosec } \phi + 1)$$

$$C1 = \frac{2 K_p K_o}{N \omega_o^2} \tan \phi$$

$$C2 = \frac{K_p K_o}{N \omega_o^2} (\sec \phi - \tan \phi)$$

Example: Design a system with the following characteristics:

- External horizontal sync of 100 kHz
- 1560 pixels per line (2000 pixels including retrace)
- 20 bit wide video data
- 10 MHz processor rate

Note that this system will sync to an external source so that both loops must be used.

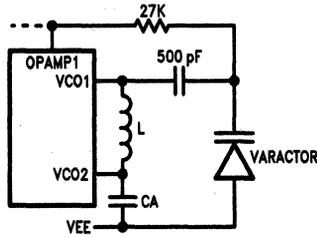
The PCLK frequency will be $100 \text{ kHz} \times 2000 \text{ pixels per line} = 200 \text{ MHz}$. The components in Table III will be used. Note that $K_o = 24 \text{ Mrad}$. Because it is a 20 bit wide system the L counter modulus must be 20.

By choosing $\omega_o = 2\pi \cdot 2800 \text{ Hz}$, the equations give $R1 = 200\Omega$, $C1 = 1.0 \mu F$ and $C2 = 0.08 \mu F$ (use $C2 = 0.1 \mu F$).

In the secondary loop, a ceramic resonator is used in place of a crystal to allow more pullability. Its K_o is found experimentally to be 84 krad/V. The SOUT frequency will be the same as LCLK0 or $200 \text{ MHz} \div 20 = 10 \text{ MHz}$. Thus, there must be a $\div 100$ counter between SOUT and RGP HORIZ so $N = 100$ for the secondary loop.

In choosing ω_o , it should be noted that ω_o for the secondary loop should be smaller than ω_o for the primary loop so that the main loop will be able to track the secondary without losing lock. Picking $\omega_o = 2\pi \times 750 \text{ Hz}$ gives $R1' = 75k$, $C1' = 0.01 \mu F$ and $C2' = 820 \text{ pF}$.

Recommended VCO Components



TL/F/8283-37

$$F_{VCO} = \frac{1}{2\pi\sqrt{LC_{TOT}}} \quad C_{TOT} = \frac{1}{\frac{1}{CA} + \frac{1}{C_{VARACTOR}}}$$

TABLE III. Recommended VCO Components

Frequency (MHz)	L μ H	TOKO Coil Type S18 Part #	CA pF	Cvaractor pF	Motorola #	K _o Mrad/volt
60	0.258	E502HNS-6000026	56	30	MV209	16
80	0.17	E502HNS-4000024	39	30	MV209	19
100	0.12	E502HNS-3000023	30	30	MV209	21
120	0.07	E502HNS-2000022	39	30	MV209	31
140	0.07	E502HNS-2000022	22	30	MV209	27
160	0.07	E502HNS-2000022	15	15	MV2205	27
180	0.07	E502HNS-2000022	10	15	MV2205	26
200	0.037	E502HNS-1000029	10	30	MV209	24
220	0.037	E502HNS-1000029	10	15	MV2205	34

DP8514 Crystal Clock Generator

General Description

The DP8514 Crystal Clock Generator consists of a crystal or LC tank oscillator and a synchronizer/2-phase nonoverlapping MOS clock driver. It is designed to interface directly with the DP8513 Video Clock Generator in multiboard graphics applications. However its features and flexible design allow it to be used in numerous other applications as well.

There are two outputs from the Pierce crystal oscillator. One is the same frequency as the crystal and has an approximate 50% duty cycle while the other is half the crystal frequency with a 50% duty cycle. Both of these outputs are TTL-compatible. The oscillator may also be used as an LC oscillator, if desired.

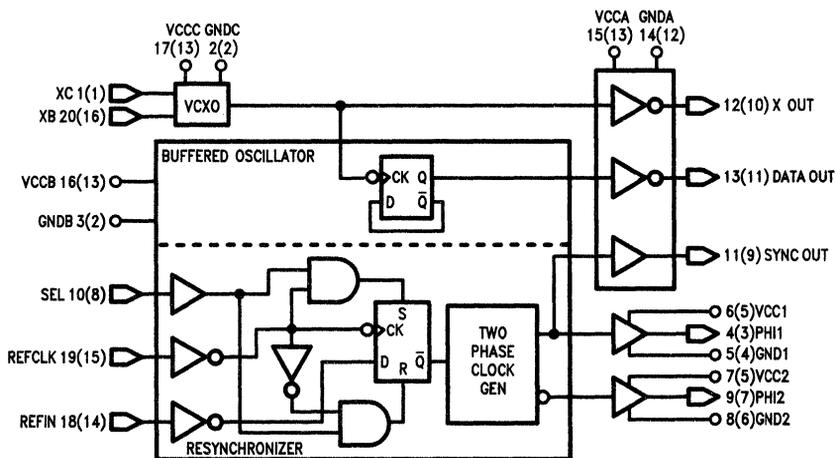
The other section of the die contains a resynchronizer with additional clock follow logic, and a 2-phase nonoverlapping MOS-compatible clock driver. Both sections of the die may be used independently of each other. The synchronizer is a D register which has a clock input, REFCLK, a data input, REFIN, and a mode control input, SEL, which allows the

REFCLK input to control the synchronizer's output. This feature allows either the clock or the resynchronized clock $\div 2$ to be fed to the MOS clock driver. A TTL output (SYNC OUT) in phase with the PHI1 output is also provided.

Features

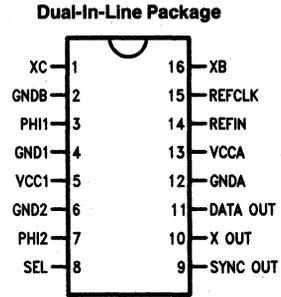
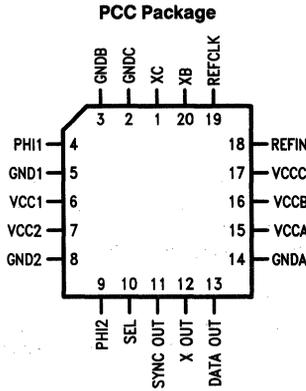
- Pierce oscillator may be used with crystal, ceramic resonator, or LC tank circuit. External varactor allows VCO or VCXO mode.
- TTL-compatible oscillator and oscillator $\div 2$ outputs.
- Two-phase nonoverlapping MOS-compatible clock outputs drive 50 pF loads at 20 MHz.
- Synchronizer/driver eases synchronization of PHI1 and PHI2 clocks on multiple boards.
- TTL-compatible SYNC OUT in phase with PHI1.
- Available in standard 16 pin DIP, 16 pin SO, and 20 pin PCC packages.

Block Diagram pin numbers are for 20 pin PCC (16 pin DIP) package



TL/F/9284-1

Connection Diagrams



TL/F/9284-3

TL/F/9284-2

Order Number DP8514
 See NS Package Number M16A, N16A or V20A

Pin Descriptions (parenthesis indicate 16 pin DIP)

- 1, 20 **XC, XB:** External connections for the Pierce oscillator. XC requires a 5.6 kΩ resistor to VCCC. (1, 16)
- 2, 3 **GNDC, GNDB (GNDB):** Power supply return for the crystal oscillator, and power supply return for all circuitry except the VCXO and output buffers. The 16 pin version has a single power supply return for all circuitry except the output buffers. (2)
- 4, 9 **PHI1, PHI2:** MOS-compatible two-phase non-overlapping clocks. The frequency of these signals is that of the REFCLK input when SEL is high and that of the REFIN input when SEL is low. (3, 7)
- 5, 8 **GND1, GND2:** PHI1 and PHI2 output buffer power supply return. (4, 6)
- 6, 7 **VCC1, VCC2 (VCC1):** PHI1 and PHI2 positive power supplies specified for operation at 5V ± 10%. (5)
- 10 **SEL:** TTL-compatible MUX control input selects either the REFCLK input or the REFIN input (resynchronized to REFCLK by a Flip Flop) to be passed to the PHI generator circuitry. A low selects the REFIN input frequency and a high selects the REFCLK input frequency. (8)
- 11 **SYNC OUT:** TTL-compatible output equivalent to PHI1. (9)
- 12 **X OUT:** TTL-compatible output of the crystal oscillator. (10)
- 13 **DATA OUT:** TTL-compatible output whose frequency is that of the crystal divided by two. (11)
- 14 (12) **GND4:** TTL output buffer power supply return.
- 15, 16, 17 **VCCA, VCCB, VCCC (VCCA):** TTL output buffer, internal circuitry, and crystal oscillator positive power supply, respectively. The 16 pin version has a single positive power supply for all circuitry except the PHI1 and PHI2 buffers. (13)
- 18 **REFIN:** TTL-compatible input typically used to generate PHI1 and PHI2. Equivalent to the REFIN input on the DP8513. (14)
- 19 **REFCLK:** TTL-compatible input typically used to synchronize multiple DP8514's. This is the CK input of a positive edge triggered D Flip Flop. Equivalent to the REFCLK input on the DP8513. (15)

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature	-65°C to +150°C
V _{CC}	7.0V
Inputs	7.0V
Outputs	7.0V
ESD Sensitivity (Note 2)	1000V

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.5	5.0	5.5	V
V _{IH}	High Level Input Voltage	2			V
V _{IL}	Low Level Input Voltage			0.8	V
I _{OH}	High Level Output Current			-0.4	mA
I _{OL}	Low Level Output Current	TTL		8	mA
		-MOS		20	mA
F _{OSC}	Oscillator Frequency			40	MHz
F _{RESYNC}	Resynchronizer Frequency			40	MHz
T _{SU}	Setup Time REFIN to REFCLK	5			ns
T _H	Hold Time REFCLK to REFIN	2			ns

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Nom	Max	Units
V _{OH}	High Level Output Voltage TTL Outputs	I _{OH} = -0.4 mA	V _{CC} - 2			V
	High Level Output Voltage MOS Outputs	I _{OH} = -0.4 mA	V _{CC} - 2.3			
		I _{OH} = -100 μA	V _{CC} - 0.4			
V _{OL}	Low Level Output Voltage TTL Outputs	I _{OL} = 8 mA			0.5	V
	Low Level Output Voltage MOS Outputs	I _{OL} = 20 mA			0.5	
		I _{OL} = 100 μA			0.4	
I _I	Max High Level Input Current	V _{TTL0, 1} = 5.5V, V _{IN} = 7V			100	μA
I _{IH}	High Level Input Current	V _{TTL0, 1} = 5.5V, V _{IN} = 2.7V			20	μA
I _{IL}	Low Level Input Current	V _{TTL0, 1} = 5.5V, V _{IN} = 0.4V			-200	μA
I _O	Output Drive Current TTL Outputs	V _O = 2.25V	-30		-150	mA
	Output Drive Current MOS Outputs	V _O = 2.25V		-135		
I _{CC}	Supply Current	V _{CC} = 5.5V		20	35	mA

MOS = PHI1,2 outputs TTL = X OUT, DATA OUT, SYNC OUT

AC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
F _{MAX} (Note 1)	Max Oscillator Frequency		40			MHz
	Max Resynchronizer Clock Frequency	40% ≤ F _{IN} Duty Cycle ≤ 60%	40			
D _C	X OUT Duty Cycle			50		%
PW	Pulse Width High; PHI1, 2	REFCLK = 40 MHz, REFEN = 20 MHz	20	23	30	ns
T _{DATA}	X OUT to DATA OUT		2	6	10	ns
T _{NO}	Non-overlap Time PHI1 to PHI2	C _L = 50 pF		0		ns

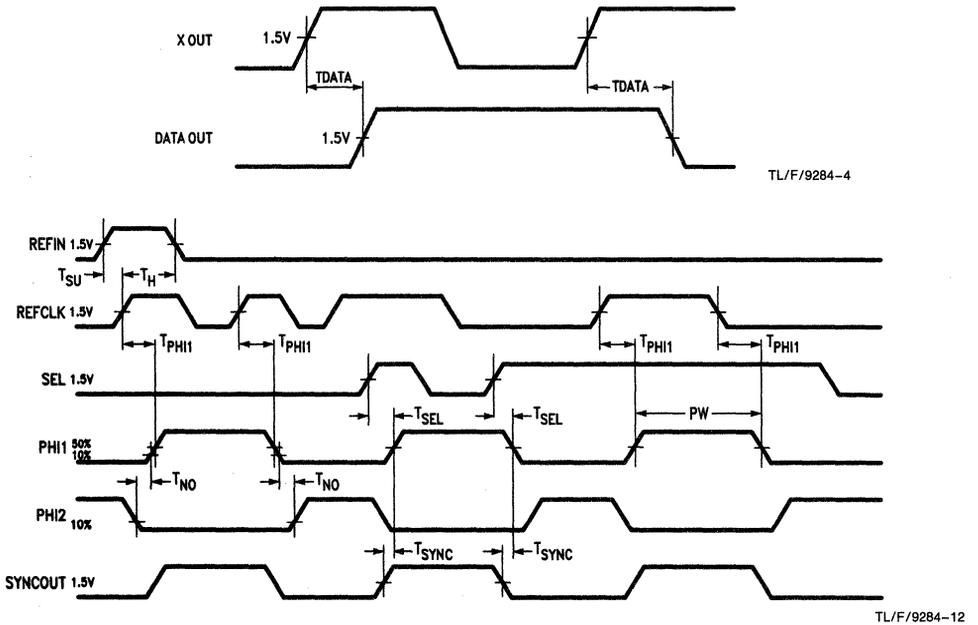
AC Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_R, T_F	Rise, Fall Time PHI1 and PHI2	$C_L = 50 \text{ pF}, 0.5\text{V to } V_{CC} - 2\text{V}$		4		ns
T_{SYNC}	SYNC OUT to PHI1		-5	0	5	ns
T_{PHI1}	REFCLK to PHI1		9	16	25	ns
	REFCLK to PHI1		9	13	25	
T_{SEL}	SEL to PHI1	REFCLK = HIGH	9	16	25	ns
	SEL to PHI1	REFCLK = LOW	9	13	25	

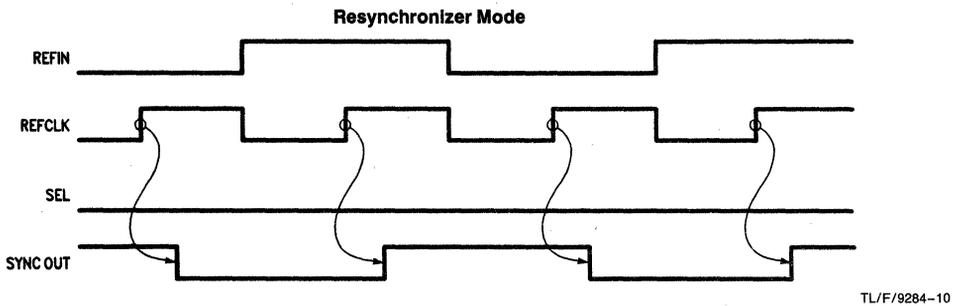
Note 1: This parameter is not production tested but is assured by characterization to include sufficient margin beyond processing extremes.

Note 2: Human body model; 120 pF thru 1.5 kΩ.

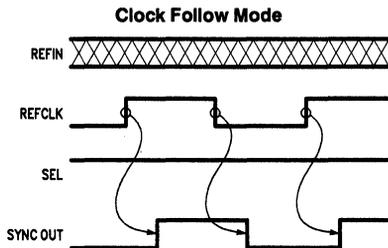
AC Timing Waveforms



Functional Waveforms

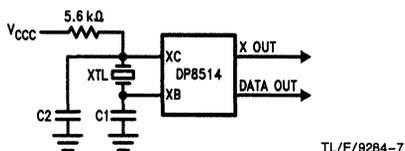


Functional Waveforms (Continued)



TL/F/9284-11

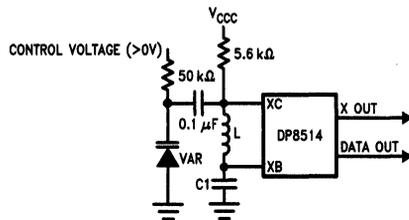
Wiring Diagrams



C1, C2 dependent on crystal ≈ 30 pF

FIGURE 1. Crystal Oscillator

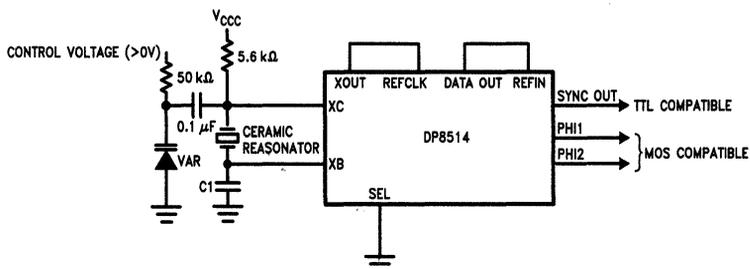
TL/F/9284-7



TL/F/9284-8

$$F_{X\text{ OUT}} = \frac{1}{2\pi\sqrt{LC_T}} \text{ where } C_T = \frac{C_1 \times \text{VAR}}{C_1 + \text{VAR}} \text{ given } 0.1 \mu\text{F} > C_T$$

FIGURE 2. VCO



TL/F/9284-9

$$F_{\text{SYNC OUT}} = F_{\text{CERAMIC RESONATOR}} \div 2$$

FIGURE 3. VCXO with MOS Compatible Outputs

Typical Applications

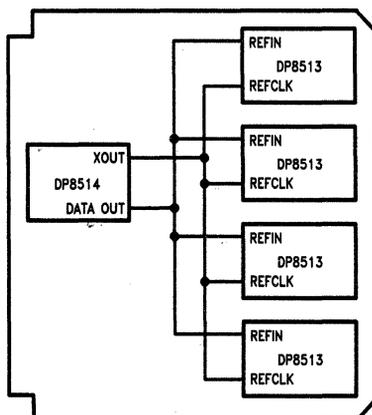


FIGURE 4. Single Board System

TL/F/9284-13

Typical Applications (Continued)

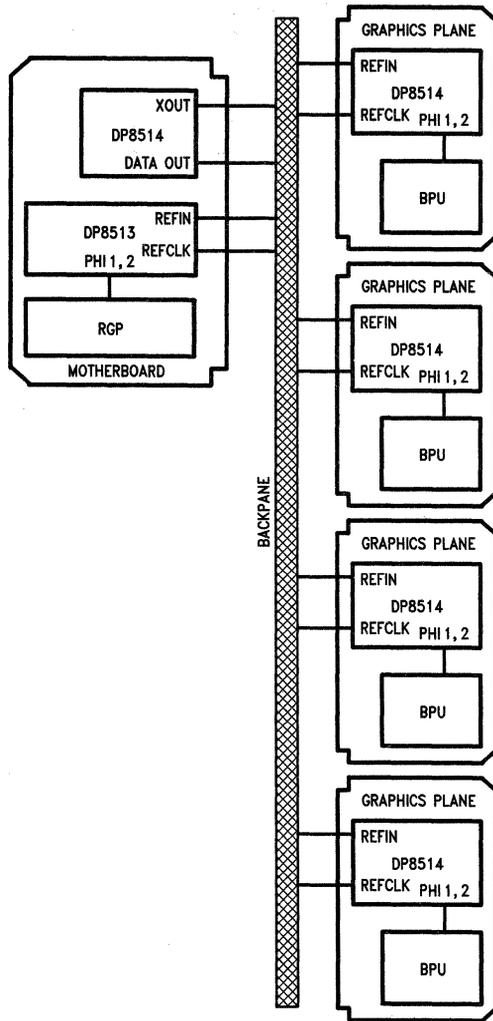


FIGURE 5. Multiboard System

TL/F/9284-14

DP8515/DP8515-350/DP8516/DP8516-350 Video Shift Register (VSR)

General Description

The DP8515/DP8515-350/DP8516/DP8516-350 Video Shift Register (VSR) provides the functions of a high speed sixteen bit shift register and parallel data input latches/flip-flops required in high performance raster scan video systems. Also on the VSR are four words of FIFO which by means of the mode control input pins M0 and M1 may be placed in front of the shift register if the user so desires.

The VSR has three operating modes; inputs configured as transparent latches, inputs configured as flip-flops (one word FIFO mode), and four word FIFO mode. As mentioned above, the mode control input pins, M0 and M1 select in which mode the part is operating. In all three modes, the WR input allows data into the part, the PARALLEL LOAD input loads data into the shift register, and the PIXEL CLOCK input shifts data out of the shift register.

In the four word FIFO mode, four write operations may occur before a shift register load operation is required in order to avoid writing over previously written data. The four words of FIFO significantly ease the timing constraints which are present when working with high speed multiple board systems.

The VSR has a HOLD input which, when activated, inhibits the shift function of the shift register. Two other inputs, OUTPUT CONTROL and OUTPUT LEVEL CONTROL, hold the last bit of the shift register at a level chosen by the user while allowing the internal bits of the shift register to continue shifting. Another control input, ENABLE, causes the parallel loading of the shift register to be inhibited when deactivated.

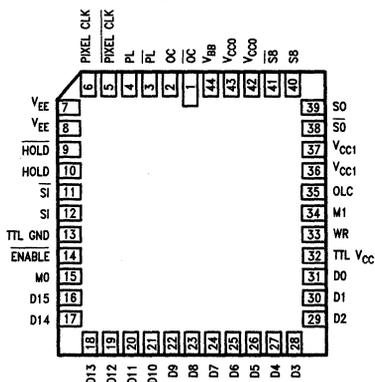
The inputs to the VSR, except those associated with the shift register, are TTL compatible. The shift register in-

puts and control signals are ECL compatible as are the outputs. Furthermore, two versions of the chip are available; the DP8515/DP8515-350 has ECL outputs which are 10K compatible and the DP8516/DP8516-350 has ECL outputs which are 100K compatible. All the ECL inputs and outputs are differential, however, a V_{BB} reference output is provided for the user who wishes to use only single ended signals. The VSR implements all the TTL to ECL conversions and gives the customer the choice of using positive or negative supplies for the ECL circuitry. When using positive supplies, the ECL and TTL may be operated off of the same, single, +5V supply.

Features

- TTL compatible parallel data inputs
- Data inputs may be used as transparent latches or flip-flops
- Four words of FIFO available—essential for high speed multiple board systems
- Accepts input data at rates up to 20 MHz (30 MHz for -350)
- Tap at eighth bit, allows use as 2 8-bit shift registers
- ECL inputs and outputs may be differential or single ended
- TTL to ECL conversion performed on chip
- Shift register clock rate of 225 MHz (350 MHz for -350)
- Can use positive or negative ECL supplies
- Entire chip can operate off of single +5V supply
- Total chip I_{CC} less than 200 mA
- Packaged in a 44-pin PLCC
- A member of National's Advanced Graphics Chip Set

Connection Diagram

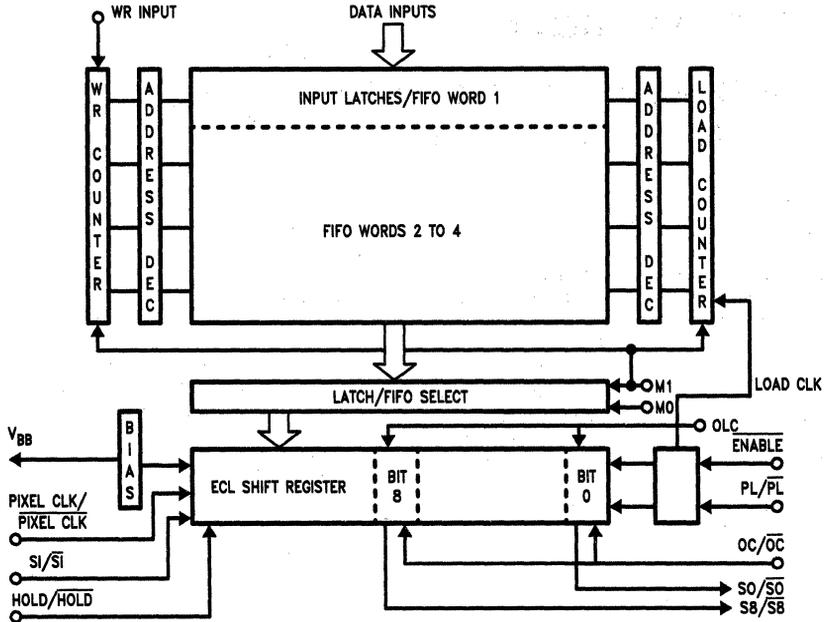


Top View

TL/F/8684-1

**Order Number DP8515V,
DP8515-350V, DP8516V,
or DP8516-350V
See NS Package Number V44A**

Block Diagram



TL/F/8684-2

Pin Descriptions

36,37 V_{CC1}—This is the supply for the collectors of the ECL emitter follower outputs. Both the 10K and 100K options are specified for a supply from 4.2V to 5.5V, assuming a positive supply is used. This allows use of the standard 5V $\pm 10\%$ supply, 5.2V $\pm 5\%$ 10K supply, or 4.5V $\pm 0.3V$ 100K supply. For a negative supply, these pins are at ground potential.

42 V_{CC0}—This is one supply for the most positive rail of the ECL circuitry. It is a separate supply for the output buffers used to reduce noise coupling. The ranges are the same as those specified for V_{CC1}.

43 V_{CC0}—This is the supply for the most positive rail of the ECL internal circuitry. The ranges are the same as those specified for V_{CC1}.

32 TTL V_{CC}—This is the positive supply for the TTL circuitry. The supply is specified at +5.0V $\pm 10\%$.

13 TTL GROUND

7,8 V_{EE}—This is the most negative rail for the ECL circuitry. Using a positive supply, this pin is at ground potential. For a negative supply, both the 10K and 100K options are specified from -4.2V to -5.5V. This allows use of the -5.2V $\pm 5\%$ 10K supply or the -4.5V $\pm 0.3V$ 100K supply.

44 V_{BB}—This is the bias reference for the ECL inputs. All of the ECL inputs are differential; however, if single ended use is desired, the unused input may be tied to this V_{BB} pin. This pin is nominally set at V_{CC0} - 1.3V at room temperature for 10K ECL and V_{CC0} - 1.3V over temperature for 100K ECL. See the electrical characteristics table for the exact specifications for V_{BB}.

16-31 DATA INPUTS—These are TTL compatible inputs designed to meet the ALS specifications (as are all TTL inputs on this circuit). The data present at these inputs will

be converted to serial data by the ECL shift register. Data on pin 31 (D0) will be the first bit shifted out of the shift register. The data on pin 16 (D15) will be the last bit shifted out of the shift register. Data at these inputs must meet the setup time requirements for the WR input.

15,34 MODE CONTROL INPUTS—M0, M1—These two TTL-compatible inputs control whether the data inputs are transparent latches or edge triggered flip flops. They also control whether or not the four word FIFO is placed in front of the shift register. With M0 set high the FIFO is not present and the data inputs are transparent latches. When M0 is high the level of the M1 input does not matter. With M0 low and M1 low the FIFO is again not present and the inputs are edge triggered flip flops. Actually this is equivalent to there being one word of FIFO present. A low level on M0 and a high level on M1 results in the four word FIFO being placed in front of the shift register. Although a power up reset is present for the FIFO address counters, if the user wishes to reset the FIFO address to word zero, a low pulse on M1 while the PARALLEL LOAD input is in a high state will accomplish this.

33 WR INPUT—This is a TTL compatible input. When data inputs are configured as flip flops, data on the bus is latched into the input flip flops on the positive edge of the WR input. If the inputs are configured as transparent latches, data is passed on to the shift register inputs so long as the WR input is low. If the four word FIFO is being used (M0 = 0 and M1 = 1), data on the bus will be latched into the first word on the positive edge of the WR input.

Pin Descriptions (Continued)

6,5 PIXEL CLOCK, PIXEL CLOCK—These are the differential clock inputs for the ECL shift register. If single ended use is desired, one input should be connected to the V_{BB} pin. The first positive PIXEL CLOCK transition after a positive transition on the PARALLEL LOAD input loads into the ECL shift register the sixteen bit word present at or already latched into the inputs by the WR input. Subsequent positive PIXEL CLOCK transitions will shift the remaining 15 bits out of the shift register.

4,3 PARALLEL LOAD, PARALLEL LOAD—These are the differential load inputs for the ECL shift register. If single ended use is desired, one input should be connected to the V_{BB} pin. When using the PARALLEL LOAD function, the setup time between PARALLEL LOAD and the positive going edge of PIXEL CLOCK must be met. To load a sixteen bit word from the input (latched or just present at the input depending on mode of operation) to the ECL shift register, a positive edge on the PARALLEL LOAD input is required and the first positive going edge of the PIXEL CLOCK following this positive edge will load the data into the shift register.

12,11 SERIAL INPUT, SERIAL INPUT—These are the differential serial data inputs to the ECL shift register. If single ended use is desired, one input should be connected to the V_{BB} pin. These inputs may be used for expansion to a wider bus system where they are connected to the SERIAL OUTPUT, SERIAL OUTPUT of the previous shift register. The information on these input pins is shifted into the shift register on the positive edge of the PIXEL CLOCK.

39,38 SERIAL OUTPUT, SERIAL OUTPUT—These are the differential data outputs of the ECL shift register. They are ECL compatible and are capable of driving 50 Ω loads. Termination resistors are required when they drive the SERIAL INPUT pins of the next shift register in an expanded bus system. The first bit shifted out is D0.

40,41 S8, S8 OUTPUTS—These are the differential data outputs of the ECL shift register for D8. If an eight bit wide word is used instead of a sixteen bit wide word, this output is the first bit of the second word. By providing this output, the number of Video Shift Registers required for eight bit wide systems is cut in half.

10,9 HOLD, HOLD—These are the differential ECL inputs used to inhibit the shifting of the ECL shift register. A high level on the HOLD input will disable the shifting of the ECL shift register. If single ended use is desired, one input should be connected to the V_{BB} pin. When using the HOLD function, the setup time between HOLD and the positive going edge of the PIXEL CLOCK must be met. When HOLD is released, the output data will not change until the following positive edge of the PIXEL CLOCK. The HOLD function can be used as a simple way of doing a "zoom" operation in graphics systems.

2,1 OUTPUT CONTROL, OUTPUT CONTROL—These are the differential ECL inputs which allow the user to manipulate the shift register data if he so chooses. If single ended use is desired, one input should be connected to the V_{BB} pin. With the OUTPUT CONTROL input high, the last bits of the shift register, that is the SERIAL OUTPUT S0 and the S8 bit are held at a level determined by the OUTPUT LEVEL CONTROL input while all the other bits are shifted by the PIXEL CLOCK. As with the HOLD input, the setup time between OUTPUT CONTROL and the positive going edge of the PIXEL CLOCK must be met. This feature provides for a simple way of implementing a scrolling function. Since all the internal bits of the shift register are shifting, the user can effectively control the shift of the data on the screen.

35 OUTPUT LEVEL CONTROL—This TTL compatible input selects the level to which the output bit of the shift register will be set when the OUTPUT CONTROL inputs are activated. A high level on this input sets the output to a one while a low level sets the output to a zero.

14 ENABLE—This TTL compatible input, when taken to a high level, inhibits the PARALLEL LOAD operation of the ECL shift register. So long as the ENABLE input remains low, every positive edge of the PARALLEL LOAD input followed by a positive edge of the PIXEL CLOCK results in a load operation on the shift register. When the ENABLE input is taken high the next PARALLEL LOAD command is ignored and all subsequent PARALLEL LOAD commands are ignored until the ENABLE input is taken low. Once taken low, the first PARALLEL LOAD positive edge and PIXEL CLOCK positive edge pair will result in a load operation.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature -65°C to +150°C

TTL Signals
 Inputs 7V
 Supply 7V

ECL Signals

Inputs (Using Negative Supply) V_{EE} to +0.5V
 Inputs (Using Positive Supply) GND to V_{CC} +0.5V
 Output Current (DC Output High) -50 mA
 Supply -7V to +0.5V
 (V_{EE} to GND Using Negative Supply)
 Supply -0.5V to +7V
 (V_{CC} to GND Using Positive Supply)

ESD Susceptibility (Note 4) 1500V

Operating Conditions (Notes 1, 3 and 5) $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$

Symbol	Parameter	Conditions	Min		Typ	Max		Units
			DP8515/16	-350		DP8515/16	-350	
V_{CC}	TTL Supply		4.5	4.5	5.0	5.5	5.5	V
V_{EE}	10K/100K ECL Using Negative Supply		-5.5	-5.5		-4.2	-4.2	V
V_{CC1}/V_{CC0}	10K/100K ECL Using Positive Supply		4.2	4.2		5.5	5.5	V
T_a	Ambient Temp		0	0	25	70	70	°C
$f_{(\text{pixel clk.})}$	Shift Rate (See Note 1 A.C. Elec. Char.)					225	350	MHz
$f_{(\text{write})}$	WR Rate					20	30	MHz
$f_{(\text{read})}$	PARALLEL LOAD Rate FIFO Mode					14	25 (30)	MHz
$f_{(\text{read})}$	PARALLEL LOAD Rate Transparent Mode					20	30	MHz
t_{W1}	Width of PIXEL CLOCK HI or LO		2.0	1.5				ns
t_{W2}	Width of WR Input LO		25	15				ns
t_{W3}	Width of WR Input HI		30 (25)	20 (18)				ns
t_{W4}	Width of PARALLEL LOAD Input HI FIFO MODE		22 (20)	14 (12)				ns
t_{W5}	Width of PARALLEL LOAD Input LO FIFO MODE		42 (32)	26 (21)				ns
t_{W6}	Width of PARALLEL LOAD Input HI Transparent Mode		15	8				ns
t_{W7}	Width of PARALLEL LOAD Input LO Transparent Mode		30 (25)	20 (17)				ns
t_{SU1}	Setup Time HOLD to PIXEL CLK		1.5	1.5				ns
t_{SU2}	Setup Time OUTPUT CONTROL to PIXEL CLK		3.0	2.5				ns
t_{SU3}	Setup Time PARALLEL LOAD to PIXEL CLK		1.5	1.5				ns
t_{SU4}	Setup Time SERIAL IN to PIXEL CLK		1.5	1.0				ns
t_{SU5}	Setup Time DATA to PARALLEL LOAD		20	10				ns
t_{SU6}	Setup Time ENABLE Inactive to PARALLEL LOAD		34	26				ns
t_{SU7}	Setup Time DATA to WR		5	4				ns
t_{SU8}	Setup Time ENABLE Active to PARALLEL LOAD		12	10				ns
t_{SU9}	Setup Time WR to PARALLEL LOAD (Note 2)		25	20				ns
t_{SU10}	Setup Time WR to PARALLEL LOAD Transparent Mode		35	26				ns
t_{H1}	Hold Time HOLD to PIXEL CLK		2.5	2.5				ns
t_{H2}	Hold Time OUTPUT CONTROL to PIXEL CLK		1.0	0.5				ns
t_{H3}	Hold Time PARALLEL LOAD to PIXEL CLK		1.0	1.0				ns
t_{H4}	Hold Time SERIAL IN to PIXEL CLK		1.3	1.1				ns
t_{H5}	Hold Time DATA to PARALLEL LOAD		0	0				ns
t_{H6}	Hold Time ENABLE Inactive to PARALLEL LOAD		0	0				ns
t_{H7}	Hold Time DATA to WR		12	10				ns
t_{H8}	Hold Time ENABLE Active to PARALLEL LOAD		0	0				ns

Note 1: See timing waveforms for relevant signal edges (positive or negative) from which all setup measurements are made.

Note 2: This is the time that a write operation on the WR input must precede a read operation on the PARALLEL LOAD input when in the one word FIFO mode (edge triggered flip flop inputs) and when in the FIFO mode and the FIFO is empty.

Note 3: Numbers in parenthesis are guaranteed when using a negative ECL supply with a V_{EE} max of -4.75V or when using a positive ECL supply with a V_{CC1}/V_{CC0} min of 4.75V.

Note 4: Human body model; 120 picofarads thru 1.5 k Ω .

Note 5: All measurements are made WRT 1.3V level on TTL signals and V_{CC0} - 1.3V on ECL signals.

DC Electrical Characteristics

TTL INPUTS AND SUPPLY (Note 3)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	$V_{CC} = \text{Min.}, I_I = -18 \text{ mA}$			-1.5	V
V_{IH}	High Level Input Voltage	$V_{CC} = \text{Max}$	2			V
V_{IL}	Low Level Input Voltage	$V_{CC} = \text{Max}$			0.8	V
I_{IH}	High Level Input Current	$V_{CC} = \text{Max.}, V_I = 2.7\text{V}$			20	μA
I_I	Max High Input Current	$V_{CC} = \text{Max.}, V_{IH} = 7\text{V}$			100	μA
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max.}, V_I = 0.4\text{V}$			-200	μA
I_{CC}	TTL Supply Current	$V_{CC} = \text{Max.}$		20	35	mA

ECL INPUTS/OUTPUTS DP8516/DP8516-350 (100K) $V_{EE} = -4.2\text{V}$ Output Load = 50Ω to -2V (Note 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage	(Notes 1, 2)	-1025	-950	-880	mV
V_{OL}	Output Low Voltage	(Notes 1, 2)	-1810	-1700	-1620	mV
V_{IH}	Input High Voltage	(Notes 1, 2)	-1165		-880	mV
V_{IL}	Input Low Voltage	(Notes 1, 2)			-1475	mV
V_{BB}	Bias Output	(Notes 1, 2) $I_{SINK}/I_{SOURCE} < 1 \text{ mA}$	-1465	-1300	-1175	mV
I_{IH}	High Level Input Current	$V_{IN} = V_{IH}(\text{max})$			100	μA
I_{IL}	Low Level Input Current	$V_{IN} = V_{IL}(\text{min})$	-100		100	μA

ECL INPUTS/OUTPUTS DP8515/DP8515-350 (10K) $V_{EE} = -5.2\text{V}$ Output Load = 50Ω to -2V (Note 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage	0°C 25°C (Note 2) 70°C	-1000 -960 -900		-840 -810 -720	mV
V_{OL}	Output Low Voltage	0°C 25°C (Note 2) 70°C	-1870 -1850 -1830		-1665 -1650 -1625	mV
V_{IH}	Input High Voltage	0°C 25°C (Note 2) 70°C	-1145 -1105 -1045		-840 -810 -720	mV
V_{IL}	Input Low Voltage	0°C 25°C (Note 2) 70°C			-1490 -1475 -1450	mV
V_{BB}	Bias Output	0°C 25°C (Note 2) 70°C $I_{SINK}/I_{SOURCE} < 1 \text{ mA}$	-1480 -1465 -1440		-1155 -1115 -1055	mV
I_{IH}	High Level Input Current	$V_{IN} = V_{IH}(\text{max})$			100	μA
I_{IL}	Low Level Input Current	$V_{IN} = V_{IL}(\text{min})$	-100		100	μA

ECL 10K/100K SUPPLIES

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{CC}	ECL Current	$V_{CC} = \text{Max.}, \text{Outputs Open}$		100	163	mA

Note 1: These ECL 100K specifications are guaranteed over the temperature range 0°C to 70°C.

Note 2: ECL voltage levels are referenced to V_{CC0} .

Note 3: TTL voltage levels are referenced to TTL Ground. TTL specifications are guaranteed over the temperature range 0°C to 70°C.

AC Electrical Characteristics

Symbol	Parameter	Conditions	Min		Typ		Max		Units
			DP8515/16	-350	DP8515/16	-350	DP8515/16	-350	
f_{MAX}	PIXEL CLK		225	350	350 (Note 1)	450 (Note 2)			MHz
f_{MAX}	WR Input		20	30					MHz
f_{MAX}	PARALLEL LOAD (read)	FIFO Mode	14	25 (30)					MHz
f_{MAX}	PARALLEL LOAD (read)	Transparent Mode	20	30					MHz
t_{PD1}	PIXEL CLK to S0 or S8	wrt PIXEL CLK POS EDGE			2.0	1.8	3.0	2.7	ns
t_{PD0}	PIXEL CLK to S0 or S8	wrt PIXEL CLK POS EDGE			2.0	1.8	3.0	2.7	ns

Note 1: f_{MAX} is not tested but is assured by correlation with characterization data. f_{MAX} of 350 MHz typical is at room temp. and 5.0V. Typical parts run at 310 MHz over temp. and V_{CC} .

Note 2: f_{MAX} is not tested but is assured by correlation with characterization data. f_{MAX} of 450 MHz typical is at room temp. and 5.0V. Typical parts run at 400 MHz over temp. and V_{CC} .

Circuit Operation

When the VSR is powered up, on chip power up reset circuitry resets the FIFO write and read pointers. This is only necessary if the FIFO mode ($M0 = 0$, $M1 = 1$) is selected, although it is performed on every power up. Although no random information in the FIFO is cleared, this is not necessary since it will be written over. With the FIFO reset, and if the FIFO mode is selected, the circuit is now ready for a write operation into word number one. If, during the operation of the circuit, a reset of the FIFO is desired, this may be accomplished by applying a low pulse to the M1 input. To ensure proper internal circuit operation, starting this FIFO "reset" operation while the PARALLEL LOAD input is in a HIGH state is NECESSARY. Again, the circuit must be in the FIFO mode to do this.

Writing into the FIFO is accomplished on the positive edge of the WR input. When this occurs, data at the input which has met the specified setup time will be latched into the first word of the FIFO. Up to four write operations may be performed without a read operation; that is, without a positive edge of the PARALLEL LOAD input. If more than four consecutive write operations occur, previously written data will be overwritten. The WR input and the PARALLEL LOAD input may be asynchronous and writing and reading may occur simultaneously. However, when the FIFO is empty, if the user wants to read the new data about to be entered, then the read operation must occur no sooner than one WR to Parallel Load Setup Time after the write operation. If this condition is not met, the old data will be read from the FIFO.

So long as the ENABLE input is low, reading data from the FIFO will not be inhibited. Reading data out of the FIFO is accomplished on the positive edge of the first PIXEL CLOCK following a positive edge of the PARALLEL LOAD input which has met the specified setup time. The first word read will be the first word written into the FIFO. The sixteen bit word will be parallel loaded into the ECL shift register and Bit 0 will appear at the S0 SERIAL OUTPUT. Subsequent PIXEL CLOCK positive edges will shift the data out of the shift register so long as the HOLD input is low; if this input is high the shifting of data by the PIXEL CLOCK will be inhibited. With a positive edge of the PARALLEL LOAD input occurring every sixteen PIXEL CLOCKS, sixteen bit words will be shifted out of the shift register. If more than sixteen PIXEL CLOCKS occur before a PARALLEL LOAD positive edge, then the information at the SERIAL INPUT pin will be shifted out of the register. This input can be used to cascade shift registers for longer word lengths.

When the M0 input is low and the M1 input is low, the FIFO depth is reduced to one word; that is, the DATA inputs become edge triggered flip flops. With the inputs configured as edge triggered flip flops, data which meets the specified setup time is accepted on the positive edge of the WR input. As is the case in the four word FIFO mode, data will be loaded into the ECL shift register on the first positive edge of the PIXEL CLOCK following a positive edge of the PARALLEL LOAD input. Since the FIFO depth is one word, it is essential that each write operation precedes a read operation by the WR to Parallel Load Setup Time. Also, in this mode, every write operation must be followed by a read operation if the user does not want to write over any data. All of the operations pertaining to the shift register remain the same in this mode as in the four word FIFO mode.

When the M0 input is high, and the M1 input is at any value, the FIFO is disconnected and the data inputs become transparent latches. In the transparent latch mode information at the data inputs is passed through to the shift register parallel load inputs so long as the WR input is low. When the WR input is taken high, the information at the data inputs is latched into the input buffers.

As in the other operating modes, data is loaded into the shift register on the positive edge of the first PIXEL CLOCK following a positive edge of the PARALLEL LOAD input. If the WR input remains low, new data is loaded into the shift register following every PARALLEL LOAD/PIXEL CLOCK positive edge pair providing the data meets the setup time with respect to the PARALLEL LOAD input. If the WR input returns high after latching in the information at the data inputs, PARALLEL LOAD/PIXEL CLOCK positive edge pairs will load the shift register with the latched data. All of the operations pertaining to the shift register remain the same in this mode as in the other modes.

The OUTPUT CONTROL and OUTPUT LEVEL CONTROL inputs allow the user to manipulate the shift register data if he so chooses. With the OUTPUT CONTROL input high, the last bits of the shift register, that is the SERIAL OUTPUT, S0, and the S8 bit, are prohibited from shifting while all the other bits are being shifted by the PIXEL CLOCK. This has the effect of performing a basic scrolling function on the screen. The OUTPUT LEVEL CONTROL input determines to what level the last bit is set. When using the OUTPUT CONTROL, even though the eighth bit is held at the output, internally, information will be shifted through this bit.

Circuit Operation (Continued)

As mentioned above, the HOLD input must be low for the PIXEL CLOCK to be able to shift data out of the shift register. A basic zoom feature may be performed by holding the HOLD input at a high level. When the HOLD input is held high for a certain number of PIXEL CLOCK periods, the same information will be present at the shift register output for the duration of the hold time. This has the effect of "stretching" the information from one pixel to multiple pixels.

The VSR may be used in systems employing word lengths other than 16 bits. For example, in a 32-bit system two

VSR chips may be cascaded to form a 32-bit word. The S0 output of the first VSR is fed into the SI input of the second to accomplish this. In systems using 8 bit words, the S8 output may be used in addition to the S0 output resulting in one VSR being used for two words and cutting in half the number of packages required to do the parallel to serial conversion.

Any word length can be used so long as the PIXEL CLOCK rate is equal to the PARALLEL LOAD rate times the word length.

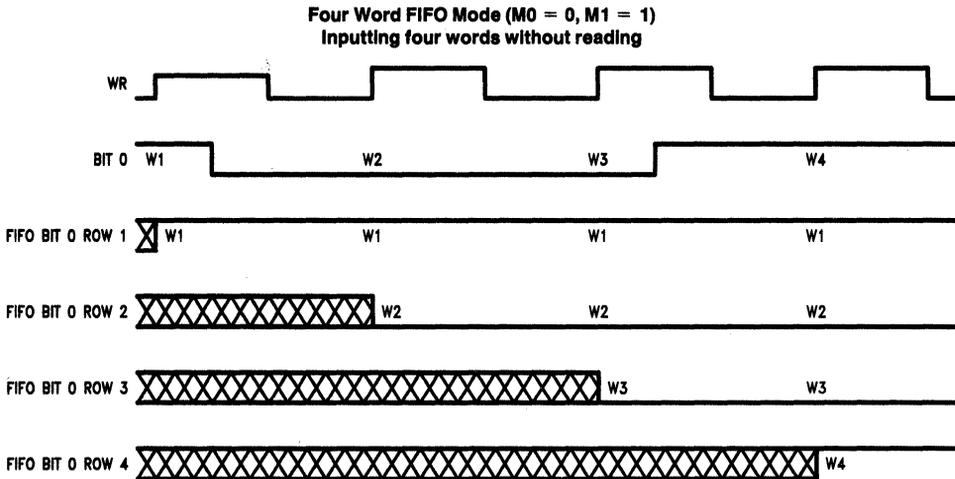


FIGURE 1

TL/F/8684-3

Four Word FIFO Mode ($M0 = 0, M1 = 1$)
 Reading four words after writing four words

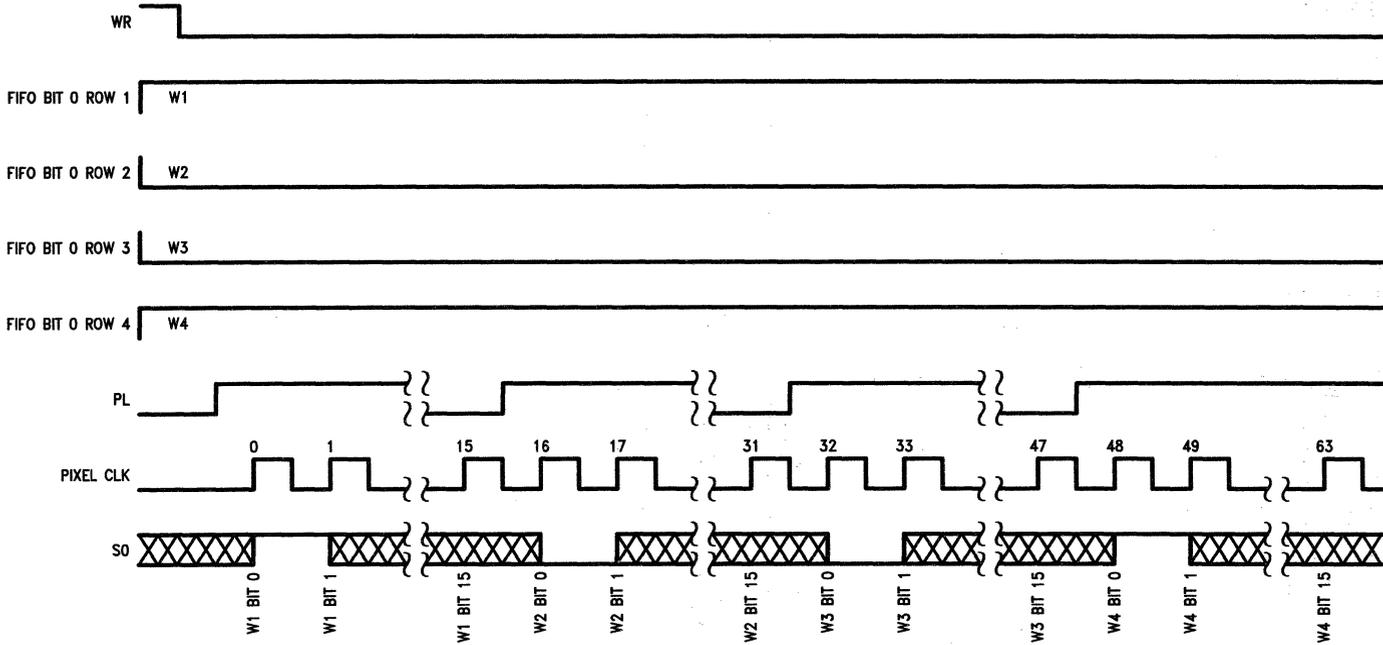


FIGURE 2

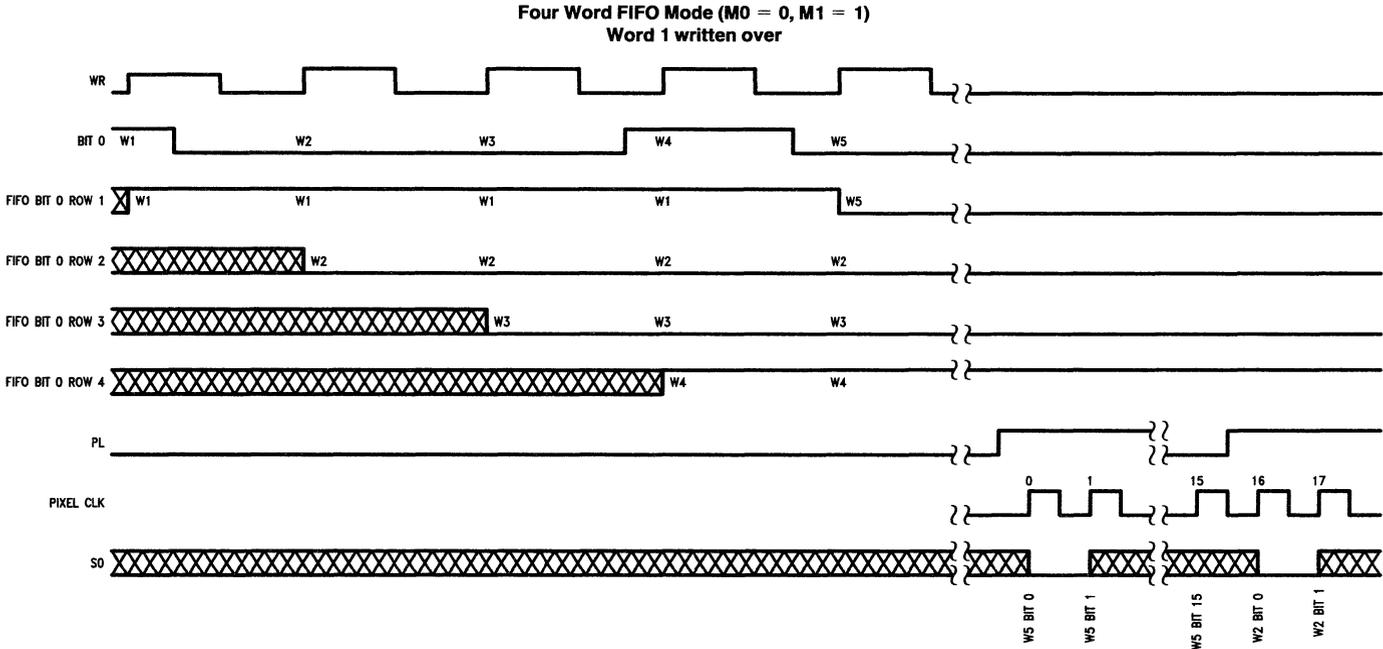


FIGURE 3

TL/F/8684-5

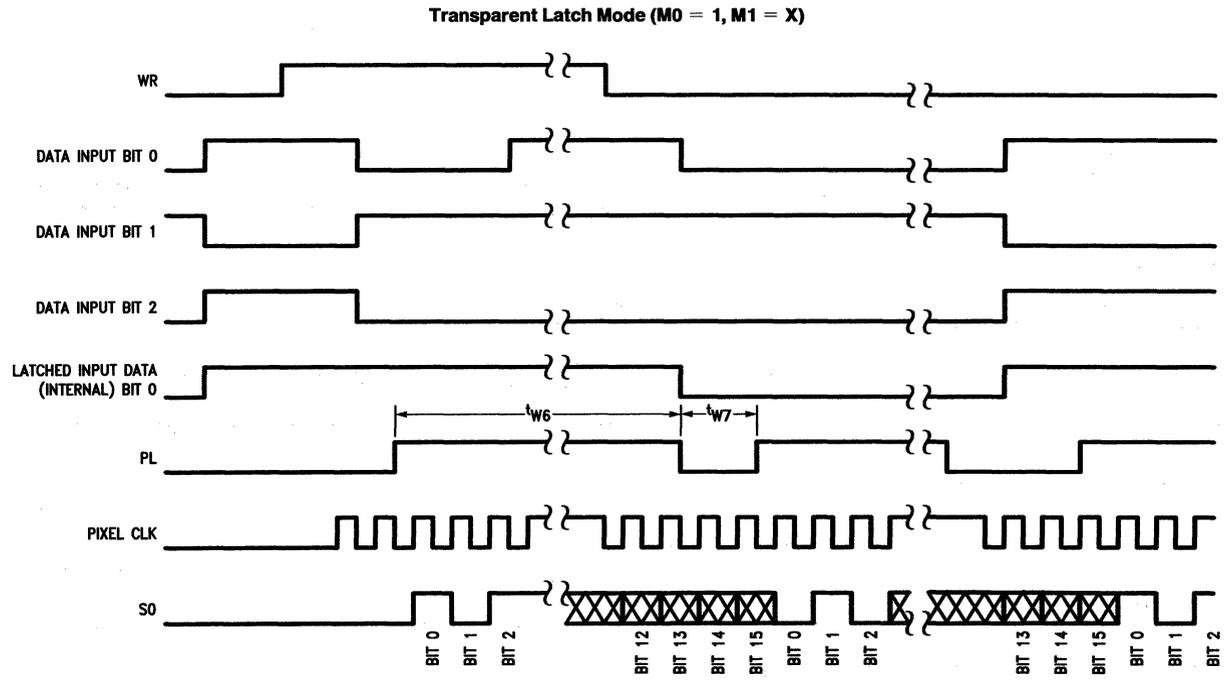


FIGURE 4

Circuit Operation (Continued)

One Word FIFO (D Flip Flop) Mode ($M0 = 0, M1 = 0$)

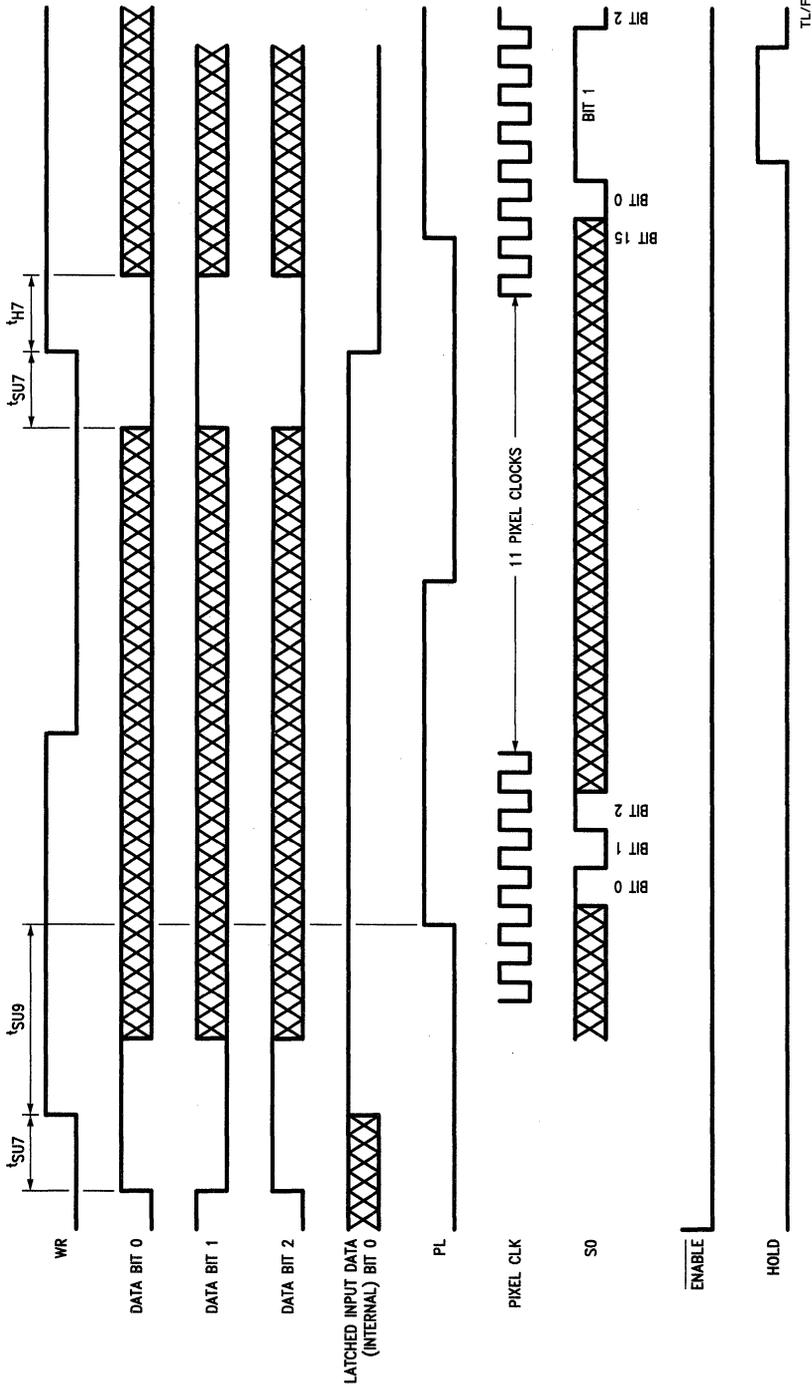
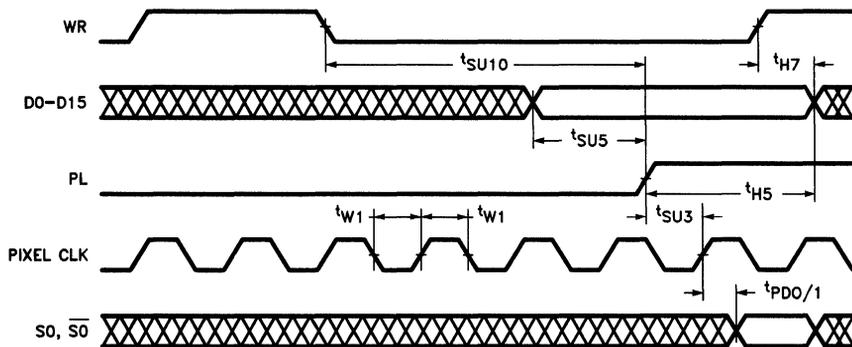


FIGURE 5

Circuit Operation (Continued)

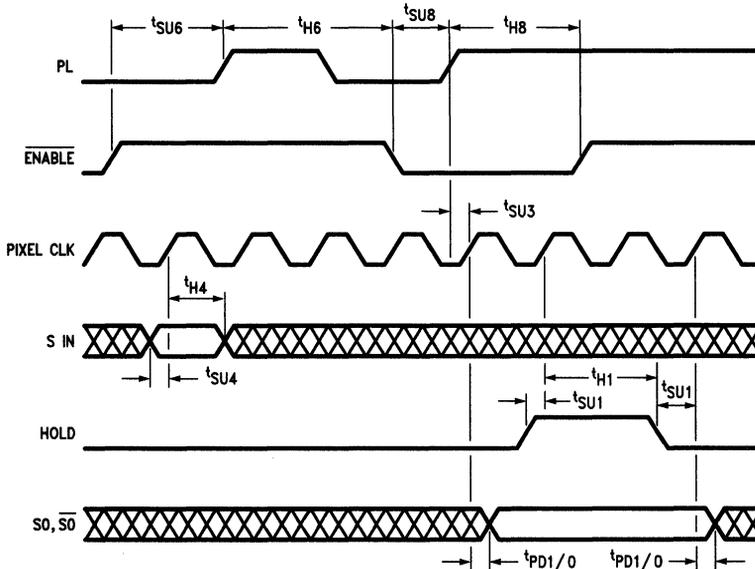
Transparent Latch Timing Requirements



TL/F/8684-16

FIGURE 8

Shift Register Timing Requirements

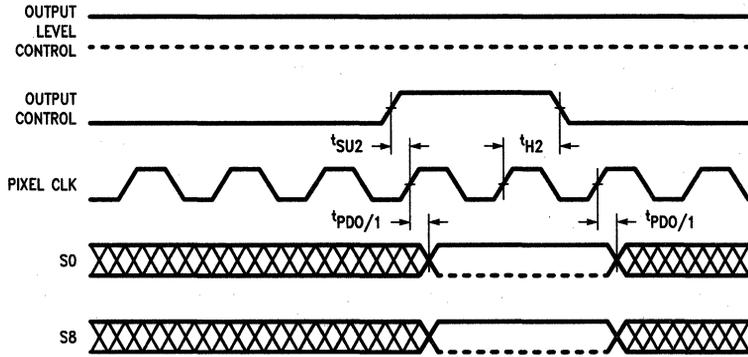


TL/F/8684-9

FIGURE 9

Circuit Operation (Continued)

Shift Register Timing Requirements



TL/F/8684-18

FIGURE 10

Equivalent Input/Output Schematics

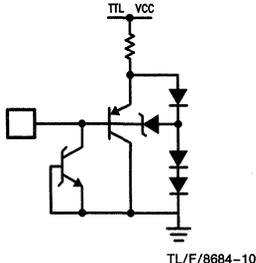


FIGURE 11. TTL Inputs

TL/F/8684-10

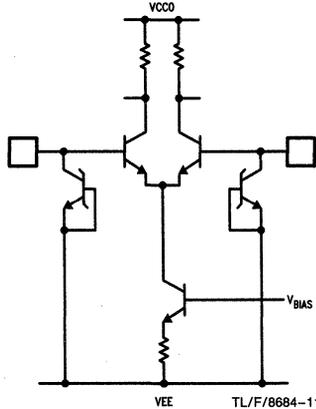


FIGURE 12. ECL Inputs

TL/F/8684-11

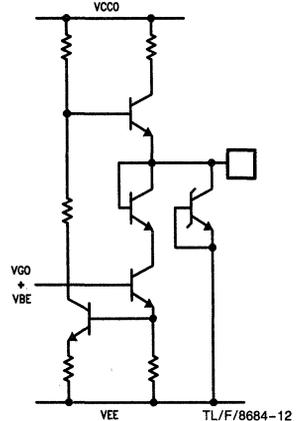


FIGURE 13. V_{BB} Output

TL/F/8684-12

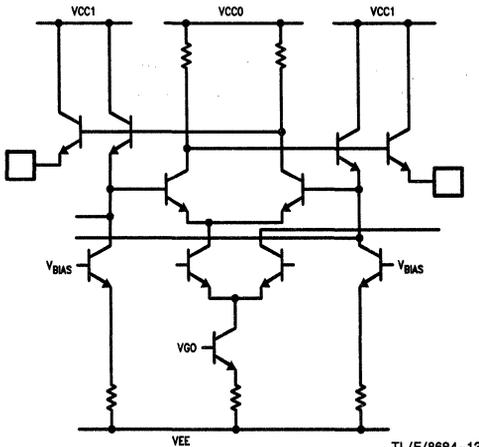


FIGURE 14. DP8515/DP8515-350 10K ECL Output

TL/F/8684-13

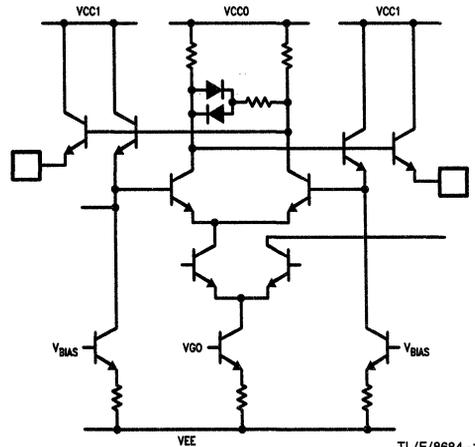


FIGURE 15. DP8516/DP8516-350 100K ECL Output

TL/F/8684-14

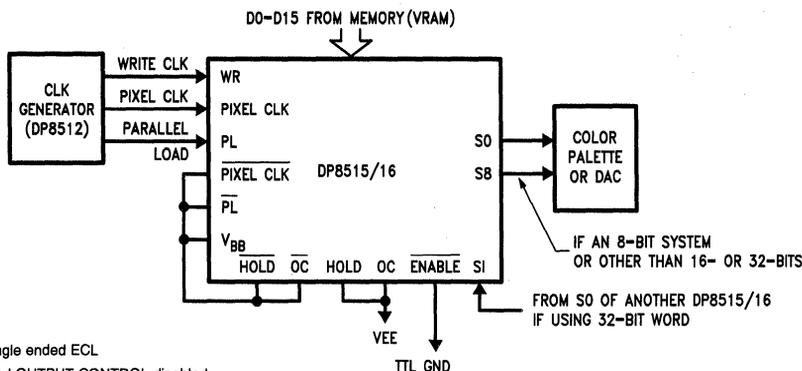
Connections to the DP8515/16

Figure 16 is a typical connection diagram for the VSR. Not shown in this diagram are the power connections. The data inputs, D0-D15, come from memory which may be Video Rams as an example. The clock signals are derived from clock generator circuitry such as National Semiconductor's DP8512. The VSR is capable of receiving differential ECL inputs. If differential use is not desired, the unused inputs may be tied to the V_{BB} reference which is provided as an output on the VSR. Although Figure 16 shows the PIXEL CLOCK and PARALLEL LOAD inputs as single ended signals, the DP8512 provides these as differential outputs.

The ECL outputs, S0 and S8, are differential. Again, the user has the option of using these single ended if he so chooses. In systems where words other than 8 or 16 bits in length are used, the SI inputs may be fed from the S0 outputs of another VSR in order to modify the word length, such as increasing the length to 32 bits.

Figure 16 shows the part being continuously enabled by having the \overline{EN} input tied to TTL ground. When this is done the clock generator circuitry should be able to inhibit the PARALLEL LOAD signal while a screen retrace is in progress. Another configuration might be to use the \overline{ENABLE} input to gate the PARALLEL LOAD input. In this mode the clock generator circuitry must still ensure that the PARALLEL LOAD signal has the proper phase relationship with respect to the video sync signal. By using the \overline{ENABLE} signal to gate the PARALLEL LOAD signal, the PARALLEL LOAD signal may be selectively gated to various video shift registers.

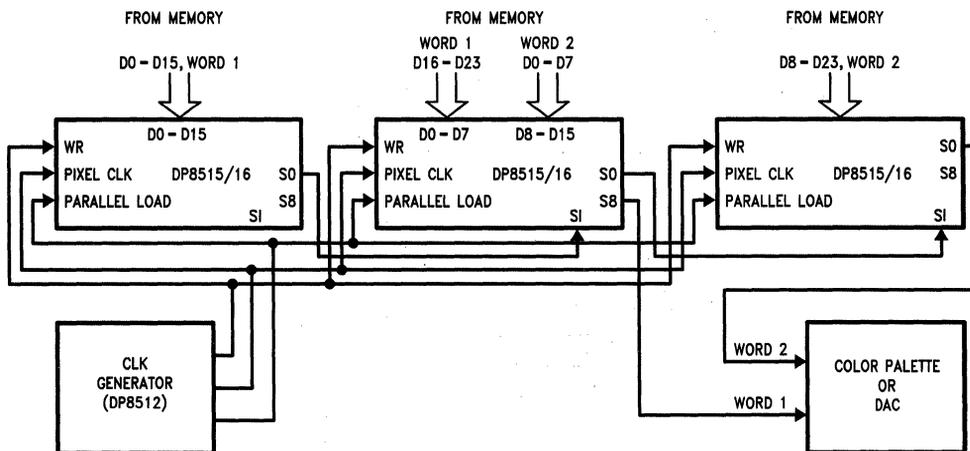
Figure 17 shows how when word lengths other than 32, 16, or 8 bits are used, several VSR circuits can be interfaced to reduce the total number of shift registers required for a system. In the example shown, three VSR circuits are used to shift two 24-bit words. Similar configurations can be used for other word lengths.



- Note 1: Using single ended ECL
- Note 2: HOLD and OUTPUT CONTROL disabled
- Note 3: Device continuously enabled

TL/F/8684-15

FIGURE 16. Typical Connection to the DP8515/DP8515-350/DP8516/DP8516-350



TL/F/8684-19

FIGURE 17. DP8515/DP8515-350/DP8516/DP8516-350 Interconnection for 24-bit Word



DP8520A/DP8521A/DP8522A microCMOS Programmable 256k/1M/4M Video RAM Controller/Drivers

General Description

The DP8520A/21A/22A video RAM controllers provide a low cost, single chip interface between video RAM and all 8-, 16- and 32-bit systems. The DP8520A/21A/22A generate all the required access control signal timing for VRAMs. An on-chip refresh request clock is used to automatically refresh the VRAM array. Refreshes and accesses are arbitrated on chip. If necessary, a WAIT or DTACK output inserts wait states into system access cycles, including burst mode accesses. RAS low time during refreshes and RAS precharge time after refreshes and back to back accesses are guaranteed through the insertion of wait states. Separate on-chip precharge counters for each RAS output can be used for memory interleaving to avoid delayed back to back accesses because of precharge. An additional feature of the DP8522A is two access ports to simplify dual accessing. Arbitration among these ports and refresh is done on chip.

Features

- On chip high precision delay line to guarantee critical VRAM access timing parameters
- microCMOS process for low power
- High capacitance drivers for $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, $\overline{\text{DT/OE}}$ and VRAM address on chip
- On chip support for nibble, page and static column VRAMs
- Byte enable signals on chip allow byte writing in a word size up to 16 bits with no external logic
- Selection of controller speeds: 20 MHz and 25 MHz
- On board Port A/Port B (DP8522A only)/refresh arbitration logic
- Direct interface to all major microprocessors (application notes available)
- 4 $\overline{\text{RAS}}$ and 4 $\overline{\text{CAS}}$ drivers (the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ configuration is programmable)

Control	# of Pins (PLCC)	# of Address Outputs	Largest VRAM Possible	Direct Drive Memory Capacity	Access Ports Available
DP8520A	68	9	256 kbit	4 Mbytes	Single Access Port
DP8521A	68	10	1 Mbit	16 Mbytes	Single Access Port
DP8522A	84	11	4 Mbit	64 Mbytes	Dual Access Ports (A and B)

Block Diagram

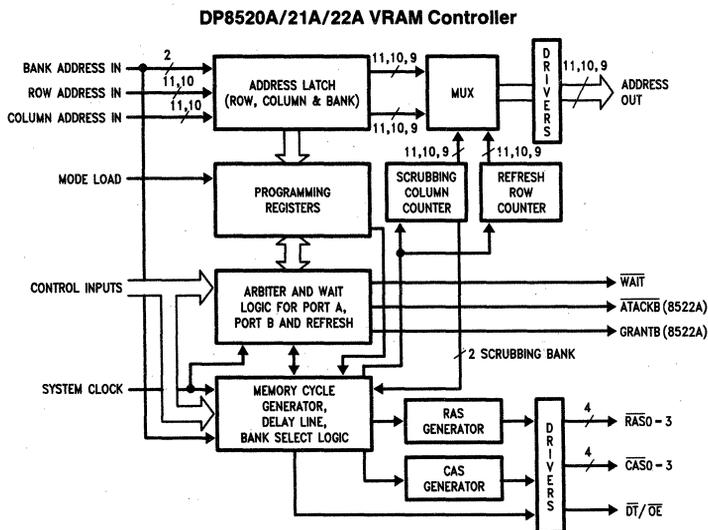


FIGURE 1

TL/F/9338-5

Table of Contents

1.0 INTRODUCTION	6.0 DP8520A/21A/22A VIDEO RAM SUPPORT
2.0 SIGNAL DESCRIPTIONS	6.1 Support for VRAM Transfer Cycles
2.1 Address, R/W and Programming Signals	6.2 Support for VRAM Access Cycles through Port A
2.2 VRAM Control Signals	7.0 ADDITIONAL ACCESS SUPPORT FEATURES
2.3 Refresh Signals	7.1 Address Latches and Column Increment
2.4 Port A Access Signals	7.2 Address Pipelining
2.5 Port B Access Signals (DP8522A)	7.3 Delay \overline{CAS} During Write Accesses
2.6 Common Dual Port Signals (DP8522A)	8.0 \overline{RAS} AND \overline{CAS} CONFIGURATION MODES
2.7 Power Signals and Capacitor Input	8.1 Byte Writing
2.8 Clock Inputs	8.2 Memory Interleaving
3.0 PORT A ACCESS MODES	8.3 Address Pipelining
3.1 Access Mode 0	8.4 Error Scrubbing
3.2 Access Mode 1	8.5 Page/Burst Mode
4.0 REFRESH OPTIONS	9.0 PROGRAMMING AND RESETTING
4.1 Refresh Control Modes	9.1 Mode Load Only Programming
4.1.1 Automatic Internal Refresh	9.2 Chip Selected Access Programming
4.1.2 Externally Controlled/Burst Refresh	9.3 External Reset
4.1.3 Refresh Request/Acknowledge	9.4 Definition of Programming Bits
4.2 Refresh Cycle Types	10.0 TEST MODE
4.2.1 Conventional Refresh	11.0 VRAM CRITICAL TIMING OPTIONS
4.2.2 Staggered Refresh	11.1 Programming Values of t_{RAH} and t_{ASC}
4.2.3 Error Scrubbing Refresh	11.2 Calculation of t_{RAH} and t_{ASC}
4.3 Extending Refresh	12.0 DUAL ACCESSING (DP8522A)
4.4 Clearing the Refresh Address Counter	12.1 Port B Access Mode
4.5 Clearing the Refresh Request Clock	12.2 Port B Wait State Support
5.0 PORT A WAIT STATE SUPPORT	12.3 Common Port A and Port B Dual Port Functions
5.1 \overline{WAIT} Type Output	12.3.1 GRANTB Output
5.1.1 \overline{WAIT} During Single Accesses	12.3.2 \overline{LOCK} Input
5.1.2 \overline{WAIT} During Page/Burst Accesses	13.0 ABSOLUTE MAXIMUM RATINGS
5.2 \overline{DTACK} Type Output	14.0 DC ELECTRICAL CHARACTERISTICS
5.2.1 \overline{DTACK} During Single Accesses	15.0 AC TIMING PARAMETERS
5.2.2 \overline{DTACK} During Page/Burst Accesses	16.0 FUNCTIONAL DIFFERENCES BETWEEN THE DP8520A/21A/22A AND THE DP8520/21/22
5.3 Wait State Support for VRAM Transfer Cycles	17.0 DP8520A/21A/22A USER HINTS
5.4 Dynamically Increasing the Number of Wait States	18.0 DESCRIPTION OF A DP8522A/DP8500 SYSTEM INTERFACE
5.5 Guaranteeing \overline{RAS} Low Time and \overline{RAS} Precharge Time	

1.0 Introduction

The DP8520A/21A/22A are CMOS Video RAM controllers that incorporate many advanced features including the capabilities of address latches, refresh counter, refresh clock, row, column and refresh address multiplexor, delay line, refresh/access/VRAM transfer cycle arbitration logic and high capacitive drivers. The programmable system interface allows any manufacturer's microprocessor or bus to directly interface via the DP8520A/21A/22A to VRAM arrays up to 64 Mbytes in size.

After power up, the DP8520A/21A/22A must first be programmed before accessing the VRAM. The chip is programmed through the address bus.

There are two methods of programming the chip. The first method, mode load only, is accomplished by asserting the signal mode load, \overline{ML} . A valid programming selection is presented on the row, column, bank and $\overline{ECAS0}$ inputs, then \overline{ML} is negated. When \overline{ML} is negated, the chip is programmed with the valid programming bits on the address bus.

The second method, chip selected access, is accomplished by asserting \overline{ML} and performing a chip selected access. When \overline{CS} and \overline{AREQ} are asserted for the access, the chip is programmed. During this programming access, the programming bits affecting the wait logic become effective immediately, allowing the access to terminate. After the access, \overline{ML} is negated and the rest of the programming bits take effect.

Once the DP8520A/21A/22A has been programmed, a 60 ms initialization period is entered. During this time, the DP8520A/21A/22A controllers perform refreshes to the VRAM array so further VRAM warm up cycles are unnecessary.

The DP8520A/21A/22A can now be used to access the VRAM. There are two modes of accessing with the controller. The two modes are Mode 0, which initiates \overline{RAS} synchronously, and Mode 1, which initiates \overline{RAS} asynchronously.

To access the VRAM using Mode 0, the signal ALE is asserted along with \overline{CS} to ensure a valid VRAM access. ALE asserting sets an internal latch and only needs to be pulsed and not held throughout the entire access. On the next rising clock edge, after the latch is set, \overline{RAS} will be asserted for that access. The DP8520A/21A/22A will place the row address on the VRAM address bus, guarantee the programmed value of row address hold time of the VRAM, place the column address on the VRAM address bus, guarantee the programmed value of column address setup time and assert \overline{CAS} . \overline{AREQ} can be asserted anytime after the clock edge which starts the access \overline{RAS} . \overline{RAS} and \overline{CAS} will extend until \overline{AREQ} is negated.

The other access mode, Mode 1, is asynchronous to the clock. When \overline{ADS} is asserted, \overline{RAS} is asserted. The DP8520A/21A/22A will place the row address on the VRAM address bus, guarantee the programmed value of row address hold time, place the column address on the VRAM address bus, guarantee the programmed value of column address setup time and assert \overline{CAS} . \overline{AREQ} can be tied to \overline{ADS} or can be asserted after \overline{ADS} is asserted. \overline{AREQ} negated will terminate the access.

The DP8520A/21A/22A also provides full support for VRAM transfer cycles. To begin the cycle, the input $\overline{AVSRLRQ}$, Advanced Video Shift Register Load Request, is

asserted and must precede the input \overline{VSRL} , Video Shift Register Load, asserting by enough CLK periods to guarantee any access in progress or pending refresh can finish. \overline{VSRL} asserting causes $\overline{DT}/\overline{OE}$ to transition low immediately. Both \overline{VSRL} and $\overline{DT}/\overline{OE}$ assert before \overline{RAS} and \overline{CAS} assert for the transfer. The cycle is ended by $\overline{DT}/\overline{OE}$ negating. This is caused by either \overline{VSRL} negating or by four rising edges of CLK from \overline{VSRL} asserting, whichever comes first.

The DP8520A/21A/22A have greatly expanded refresh capabilities compared to other VRAM controllers. There are three modes of refreshing available. These modes are internal automatic refreshing, externally controlled/burst refreshing, and refresh request/acknowledge refreshing. Any of these modes can be used together or separately to achieve the desired results. In any combination of these modes, the programming of $\overline{ECAS0}$ determines the use of the \overline{RFIP} (\overline{RRFQ}) pin. $\overline{ECAS0}$ asserted during programming causes this pin to function as \overline{RFIP} which will assert just prior to a refresh cycle and will negate when the refresh is completed. $\overline{ECAS0}$ negated during programming causes this pin to function as \overline{RRFQ} which indicates an internal refresh request when asserted.

When using internal automatic refreshing, the DP8520A/21A/22A will generate an internal refresh request from the refresh request clock. The DP8520A/21A/22A will arbitrate between the refresh requests and accesses. Assuming an access is not currently in progress, the DP8520A/21A/22A will grant a refresh, assert \overline{RFIP} if programmed, and on the next positive clock edge, refreshing will begin. If an access had been in progress, the refresh will begin after the access has terminated.

To use externally controlled/burst refresh, the user disables the internal refresh request by asserting the input $\overline{DISRRFSH}$. A refresh can now be externally requested by asserting the input \overline{RFSH} . The DP8520A/21A/22A will arbitrate between the external refresh request and accesses. Assuming an access is not currently in progress, the DP8520A/21A/22A will grant a refresh, assert \overline{RFIP} if programmed, and on the next positive clock edge, refreshing will begin. If an access had been in progress, the refresh would take place after the access has terminated.

With refresh request/acknowledge mode, the DP8520A/21A/22A broadcasts the internal refresh request to the system through the \overline{RRFQ} output pin. External circuitry can determine when to refresh the VRAM through the \overline{RFSH} input.

The controllers have three types of refreshing available: conventional, staggered and error scrubbing. Any refresh control mode can be used with any type of refresh. In a conventional refresh, all of the \overline{RAS} outputs will be asserted and negated at once. In a staggered refresh, the \overline{RAS} outputs will be asserted one positive clock edge apart. Error scrubbing is the same as conventional refresh except that a \overline{CAS} will be asserted during a refresh allowing the system to run that data through an EDAC chip and write it back to memory, if a single bit error has occurred. The refreshes can be extended with the $\overline{EXTEND REFRESH}$ input, $\overline{EXTNDRF}$.

The DP8520A/21A/22A have wait support available as \overline{DTACK} or \overline{WAIT} . Both are programmable. \overline{DTACK} , Data Transfer Acknowledge, is useful for processors whose wait signal is active high. \overline{WAIT} is useful for processors whose

1.0 Introduction (Continued)

wait signal is active low. The user can choose either at programming. These signals are used by the on-chip arbiter to insert wait states to guarantee the arbitration between accesses and refreshes or precharge. Both signals are independent of the access mode chosen.

\overline{DTACK} will assert a programmed number of clock edges from the event that starts the access \overline{RAS} . \overline{DTACK} will be negated, when the access is terminated, by \overline{AREQ} being negated. \overline{DTACK} can also be programmed to toggle with the \overline{ECAS} inputs during burst/page mode accesses.

\overline{WAIT} is asserted during the start of the access (ALE and CS, or \overline{ADS} and CS) and will negate a number of clock edges from the event that starts the access \overline{RAS} . After \overline{WAIT} is negated, it will stay negated until the next access. \overline{WAIT} can also be programmed to toggle with \overline{ECAS} inputs during a burst/page mode access.

Both signals can be dynamically delayed further through the \overline{WAITIN} signal to the DP8520A/21A/22A.

The DP8520A/21A/22A have address latches, used to latch the bank, row and column address inputs. Once the address is latched, a column increment feature can be used to increment the column address. The address latches can also be programmed to be fall through.

The \overline{RAS} and \overline{CAS} drivers can be configured to drive a one, two or four bank memory array up to 32 bits in width. The two \overline{ECAS} signals can then be used to select one pair of \overline{CAS} drivers for byte writing with no external logic for systems with a word length of up to 16 bits.

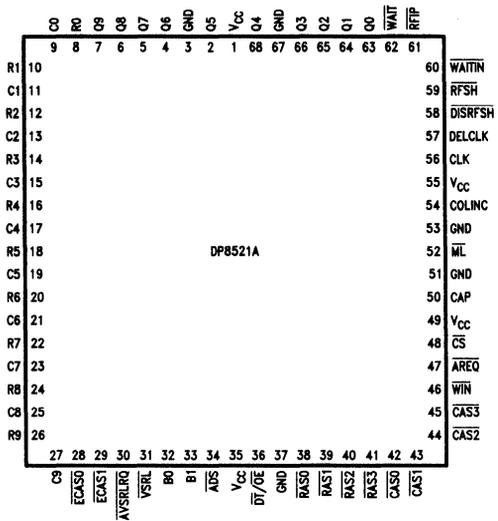
When configuring the DP8520A/21A/22A for more than one bank, memory interleaving can be used. By tying the low order address bits to the bank select lines, B0 and B1, sequential back to back accesses will not be delayed since the DP8520A/21A/22A have separate precharge counters per bank. The DP8520A/21A/22A are capable of performing address pipelining. In address pipelining, the DP8520A/21A/22A guarantee the column address hold time and switch the internal multiplexor to place the row address on the address bus. At this time, another memory access to another bank can be initiated.

The DP8522A has all the features previously mentioned. Unlike the DP8520A/21A, the DP8522A has a second port to allow a second CPU to access the memory array. This port, Port B, has two control signals to allow a CPU to access the VRAM array. These signals are access request for Port B, \overline{AREQB} , and Advanced Transfer ACKnowledge for Port B, \overline{ATACKB} . Two other signals are used by both Port A and Port B for dual accessing purposes. The signals are lock, \overline{LOCK} and grant Port B, \overline{GRANTB} . All arbitration for the two ports and refresh is done on-chip by the DP8522A through the insertion of wait states. Since the DP8522A has only one input address bus, the address lines have to be multiplexed externally. The signal \overline{GRANTB} can be used for this purpose since it is asserted when Port B has access to the VRAM array and negated when Port A has access to the VRAM array. Once a port has access to the array, the other port can be "locked out" by asserting the input \overline{LOCK} . \overline{AREQB} , when asserted, is used by Port B to request an access. \overline{ATACKB} , when asserted, signifies that access \overline{RAS} has been asserted for the requested Port B access. By using \overline{ATACKB} , the user can generate an appropriate \overline{WAIT} or \overline{DTACK} like signal for the Port B CPU.

The following explains the terminology used in this data sheet. The terms negated and asserted are used. Asserted refers to a "true" signal. Thus, " $\overline{ECAS0}$ asserted" means the $\overline{ECAS0}$ input is at a logic 0. The term "COLINC asserted" means the COLINC input is at a logic 1. The term negated refers to a "false" signal. Thus, " $\overline{ECAS0}$ negated" means the $\overline{ECAS0}$ input is at a logic 1. The term "COLINC negated" means the input COLINC is at a logic 0. The table shown below clarifies this terminology.

Signal	Action	Logic Level
Active High	Asserted	High
Active High	Negated	Low
Active Low	Asserted	Low
Active Low	Negated	High

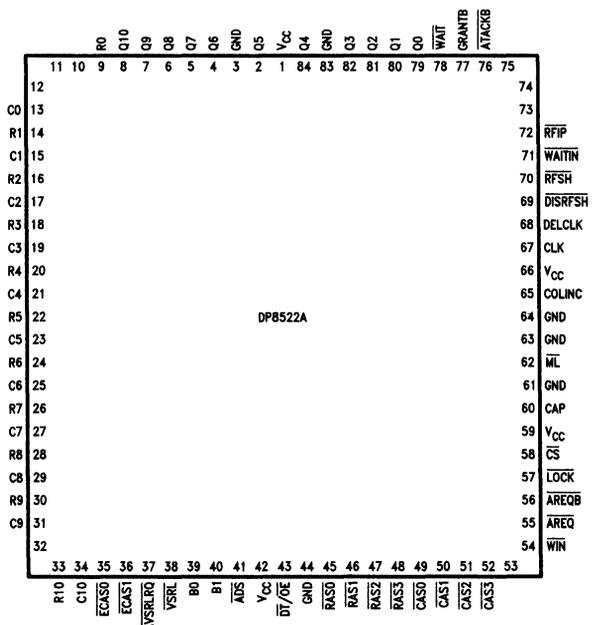
Connection Diagrams (Continued)



TL/F/9338-3

Top View
FIGURE 3

Order Number DP8521AV-20 or DP8521AV-25
See NS Package Number V68A



TL/F/9338-4

Top View
FIGURE 4

Order Number DP8522AV-20 or DP8522AV-25
See NS Package Number V84A

2.0 Signal Descriptions

Pin Name	Device (if not applicable to all)	Input/Output	Description
2.1 ADDRESS, R/W AND PROGRAMMING SIGNALS			
R0-10 R0-9	DP8522A DP8520A/21A	I I	ROW ADDRESS: These inputs are used to specify the row address during an access or refresh to the VRAM or for a VRAM transfer cycle. They are also used to program the chip when \overline{ML} is asserted (except R10).
C0-10 C0-9	DP8522A DP8520A/21A	I I	COLUMN ADDRESS: These inputs are used to specify the column address during an access to the VRAM or for a VRAM transfer cycle. They are also used to program the chip when \overline{ML} is asserted (except C10).
B0, B1		I	BANK SELECT: Depending on programming, these inputs are used to select a group of \overline{RAS} and \overline{CAS} outputs to assert during an access. They are also used to program the chip when \overline{ML} is asserted.
$\overline{ECAS0-1}$		I	ENABLE \overline{CAS}: These inputs are used to enable a single or group of \overline{CAS} outputs when asserted. In combination with the B0, B1 and the programming bits, these inputs select which \overline{CAS} output or \overline{CAS} outputs will assert during an access. $\overline{ECAS0}$ must be asserted for either $\overline{CAS0}$ or $\overline{CAS1}$ to assert during an access. $\overline{ECAS1}$ must be asserted for either $\overline{CAS2}$ or $\overline{CAS3}$ to assert during an access. The \overline{ECAS} signals can also be used to toggle a group of \overline{CAS} outputs for page/nibble mode accesses. They also can be used for byte write operations. If $\overline{ECAS0}$ is negated during programming, continuing to assert the $\overline{ECAS0}$ while negating \overline{AREQ} or \overline{AREQB} during an access, will cause the \overline{CAS} outputs to be extended while the \overline{RAS} outputs are negated (the \overline{ECASn} inputs have no effect during scrubbing refreshes).
\overline{WIN}		I	WRITE ENABLE IN: This input is used to signify a write operation to the VRAM. This input asserted will also cause \overline{CAS} to delay to the next positive clock edge if address bit C9 is asserted during programming.
COLINC (EXTNDRF)		I I	COLUMN INCREMENT: When the address latches are used, and a refresh is not in progress, this input functions as COLINC. Asserting this signal causes the column address to be incremented by one. When a refresh is in progress, this signal, when asserted, is used to extend the refresh cycle by any number of periods of CLK until it is negated.
\overline{ML}		I	MODE LOAD: This input signal, when low, enables the internal programming register that stores the programming information.
2.2 VRAM CONTROL SIGNALS			
Q0-10 Q0-9 Q0-8	DP8522A DP8521A DP8521A	O O O	VRAM ADDRESS: These outputs are the multiplexed output of the R0-9, 10 and C0-9, 10 and form the VRAM address bus. These outputs contain the refresh address whenever a refresh is in progress. They contain high capacitive drivers with 20 Ω series damping resistors.
$\overline{RAS0-3}$		O	ROW ADDRESS STROBES: These outputs are asserted to latch the row address contained on the outputs Q0-8, 9, 10 into the VRAM. When a refresh is in progress, the \overline{RAS} outputs are used to latch the refresh row address contained on the Q0-8, 9, 10 outputs in the VRAM. These outputs contain high capacitive drivers with 20 Ω series damping resistors.
$\overline{CAS0-3}$		O	COLUMN ADDRESS STROBES: These outputs are asserted to latch the column address contained on the outputs Q0-8, 9, 10 into the VRAM. These outputs have high capacitive drivers with 20 Ω series damping resistors.
$\overline{DT/OE}$		O	DATA TRANSFER/OUTPUT ENABLE: This output transitions low before \overline{RAS} goes low and transitions high before \overline{RAS} goes high during a video RAM shift register load operation (see \overline{VSRL} pin description). During normal write accesses this output is held high, and for read accesses this output is asserted after \overline{CAS} is asserted, and is negated after \overline{CAS} negates.

2.0 Signal Descriptions (Continued)

Pin Name	Device (if not applicable to all)	Input/ Output	Description
2.3 REFRESH SIGNALS			
RFP($\overline{\text{RFRQ}}$)		O	REFRESH IN PROGRESS or REFRESH REQUEST : When $\overline{\text{ECAS0}}$ is asserted during programming, this output functions as RFP, and is asserted prior to a refresh cycle and is negated when all the RAS outputs are negated for that refresh. When $\overline{\text{ECAS0}}$ is negated during programming, this output functions as RFRQ. When asserted, this pin specifies that 13 μs or 15 μs have passed. If $\overline{\text{DISRFSH}}$ is negated, the DP8520A/21A/22A will perform an internal refresh as soon as possible. If $\overline{\text{DISRFSH}}$ is asserted, RFRQ can be used to externally request a refresh through the input RFSH. This output has a high capacitive driver and a 20 Ω series damping resistor.
RFSH		I	REFRESH : This input asserted with $\overline{\text{DISRFSH}}$ already asserted will request a refresh. If this input is continually asserted, the DP8520A/21A/22A will perform refresh cycles in a burst refresh fashion until the input is negated. If RFSH is asserted with $\overline{\text{DISRFSH}}$ negated, the internal refresh address counter is cleared (useful for burst refreshes).
$\overline{\text{DISRFSH}}$		I	DISABLE REFRESH : This input is used to disable internal refreshes and must be asserted when using RFSH for externally requested refreshes.
2.4 PORT A ACCESS			
$\overline{\text{ADS}}$ (ALE)		I I	ADDRESS STROBE or ADDRESS LATCH ENABLE : Depending on programming, this input can function as $\overline{\text{ADS}}$ or ALE. In mode 0, the input functions as ALE and when asserted along with $\overline{\text{CS}}$ causes an internal latch to be set. Once this latch is set an access will start from the positive clock edge of CLK as soon as possible. In Mode 1, the input functions as $\overline{\text{ADS}}$ and when asserted along with $\overline{\text{CS}}$, causes the access RAS to assert if no other event is taking place. If an event is taking place, RAS will be asserted from the positive edge of CLK as soon as possible. In both cases, the low going edge of this signal latches the bank, row and column address if programmed to do so.
$\overline{\text{CS}}$		I	CHIP SELECT : This input signal must be asserted to enable a Port A access.
$\overline{\text{AREQ}}$		I	ACCESS REQUEST : This input signal in Mode 0 must be asserted some time after the first positive clock edge after ALE has been asserted. When this signal is negated, RAS is negated for the access. In Mode 1, this signal must be asserted before $\overline{\text{ADS}}$ can be negated. When this signal is negated, RAS is negated for the access.
WAIT (DTACK)		O O	WAIT or DTACK : This output can be programmed to insert wait states into a CPU access cycle. With R7 negated during programming, the output will function as a WAIT type output. In this case, the output will be active low to signal a wait condition. With R7 asserted during programming, the output will function as DTACK. In this case, the output will be negated to signify a wait condition and will be asserted to signify the access has taken place. Each of these signals can be delayed by a number of positive clock edges or negative clock levels of CLK to increase the microprocessor's access cycle through the insertion of wait states.
WAITIN		I	WAIT INCREASE : This input can be used to dynamically increase the number of positive clock edges of CLK until DTACK will be asserted or WAIT will be negated during a VRAM access.
2.5 PORT B ACCESS SIGNALS			
$\overline{\text{AREQB}}$	DP8522A only	I	PORT B ACCESS REQUEST : This input asserted will latch the row, column and bank address if programmed, and requests an access to take place for Port B. If the access can take place, RAS will assert immediately. If the access has to be delayed, RAS will assert as soon as possible from a positive edge of CLK.
$\overline{\text{ATACKB}}$	DP8522A only	O	ADVANCED TRANSFER ACKNOWLEDGE PORT B : This output is asserted when the access RAS is asserted for a Port B access. This signal can be used to generate the appropriate DTACK or WAIT type signal for Port B's CPU or bus.

2.0 Signal Descriptions (Continued)

Pin Name	Device (If not applicable to all)	Input/ Output	Description
2.6 COMMON DUAL PORT SIGNALS			
GRANTB	DP8522A only	O	GRANT B: This output indicates which port is currently granted access to the VRAM array. When GRANTB is asserted, Port B has access to the array. When GRANTB is negated, Port A has access to the VRAM array. This signal is used to multiplex the signals R0–8, 9, 10; C0–8, 9, 10; B0–1; WIN; LOCK and ECAS0–1 to the DP8522A when using dual accessing.
LOCK	DP8522A only	I	LOCK: This input can be used by the currently granted port to “lock out” the other port from the VRAM array by inserting wait states into the locked out port’s access cycle until LOCK is negated.
2.7 VRAM TRANSFER CYCLE SIGNALS			
AVSRLRQ		I	ADVANCED VIDEO SHIFT REGISTER LOAD REQUEST: This must precede the VSRL input going low by the amount of time necessary to guarantee that any currently executing access and pending refresh can finish. This input disables Port B and refresh requests until four CLK periods after VSRL has transitioned low. This input may be held low until the video RAM transfer cycle is completed or may be momentarily pulsed low.
VSRL		I	VIDEO SHIFT REGISTER LOAD: This input causes the $\overline{DT/OE}$ output to transition low immediately. Therefore, when executing a video RAM shift register load, VSRL transitions low before RAS goes low. The $\overline{DT/OE}$ output will transition high from VSRL going high or four CLK periods (rising clock edges) from VSRL going low, whichever occurs first. VSRL low also disables the WIN input from affecting the $\overline{DT/OE}$ logic, until the video shift register load access is over.
2.8 POWER SIGNALS AND CAPACITOR INPUT			
VCC		I	POWER: Supply Voltage.
GND		I	GROUND: Supply Voltage Reference.
CAP		I	CAPACITOR: This input is used by the internal PLL for stabilization. The value of the ceramic capacitor should be 0.1 μ F and should be connected between this input and ground.
2.9 CLOCK INPUTS			
There are two clock inputs to the DP8520A/21A/22A, CLK and DELCLK. These two clocks may both be tied to the same clock input, or they may be two separate clocks, running at different frequencies, asynchronous to each other.			
CLK		I	SYSTEM CLOCK: This input may be in the range of 0 Hz up to 25 MHz. This input is generally a constant frequency but it may be controlled externally to change frequencies or perhaps be stopped for some arbitrary period of time. This input provides the clock to the internal state machine that arbitrates between accesses and refreshes. This clock’s positive edges and negative levels are used to extend the WAIT (\overline{DTACK}) signals. This clock is also used as the reference for the RAS precharge time and RAS low time during refresh. All Port A and Port B accesses are assumed to be synchronous to the system clock CLK.
DELCLK		I	DELAY LINE CLOCK: The clock input DELCLK, may be in the range of 6 MHz to 20 MHz and should be a multiple of 2 (i.e., 6, 8, 10, 12, 14, 16, 18, 20 MHz) to have the DP8520A/21A/22A switching characteristics hold. If DELCLK is not one of the above frequencies the accuracy of the internal delay line will suffer. This is because the phase locked loop that generates the delay line assumes an input clock frequency of a multiple of 2 MHz. For example, if the DELCLK input is at 7 MHz and we choose a divide by 3 (program bits C0–2) this will produce 2.333 MHz which is 16.667% off of 2 MHz. Therefore, the DP8520A/21A/22A delay line would produce delays that are shorter (faster delays) than what is intended. If divide by 4 was chosen the delay line would be longer (slower delays) than intended (1.75 MHz instead of 2 MHz). (See Section 10 for more information.) This clock is also divided to create the internal refresh clock.

3.0 Port A Access Modes

The DP8520A/21A/22A have two general purpose access modes. With one of these modes, any microprocessor can be interfaced to VRAM. A Port A access to VRAM is initiated by two input signals: \overline{ADS} (ALE) and \overline{CS} . The access is always terminated by one signal: \overline{AREQ} . These input signals should be synchronous to the input clock, CLK. One of these access modes is selected at programming through the B1 input signal. In both modes, once an access has been requested by \overline{CS} and \overline{ADS} (ALE), the DP8522A will guarantee the following:

The DP8520A/21A/22A will have the row address valid to the VRAMs' address bus, Q0–8, 9, 10 given that the row address setup time to the DP8520A/21A/22A was met;

The DP8520A/21A/22A will bring the appropriate \overline{RAS} or \overline{RASs} low;

The DP8520A/21A/22A will guarantee the minimum row address hold time, before switching the internal multiplexer to place the column address on the VRAM address bus, Q0–8, 9, 10;

The DP8520A/21A/22A will guarantee the minimum column address setup time before asserting the appropriate \overline{CAS} or \overline{CASs} ;

The DP8520A/21A/22A will hold the column address valid the minimum specified column address hold time in address pipelining mode and will hold the column address valid the remainder of the access in non-pipelining mode.

The chip includes a \overline{WIN} pin to signify a write operation to the DP8520A/21A/22A. When asserted, \overline{WIN} will cause \overline{CAS} to delay to the next positive clock edge if address bit C9 is asserted during programming. When negated, \overline{WIN} will cause the $\overline{DT/OE}$ output to follow the \overline{CAS} outputs for a read access, if $\overline{ECAS0}$ is negated during programming. \overline{WE} , write enable, must be externally gated from the processor to the VRAM as there is no output pin from the \overline{WIN} input pin available on chip.

3.1 ACCESS MODE 0

Access Mode 0, shown in *Figure 5a*, is selected by negating the input B1 during programming. This access mode allows accesses to VRAM to always be initiated from the positive edge of the system input clock, CLK. To initiate a Mode 0 access, ALE is pulsed high and \overline{CS} is asserted. Pulsing ALE high and asserting \overline{CS} , sets an internal latch which requests an access. If the precharge time from the last access or VRAM refresh had been met and a refresh of VRAM, a Port B access, or a VRAM transfer cycle was not in progress, the \overline{RAS} or group of \overline{RASs} would be initiated from the first positive edge of CLK. If a VRAM refresh is in progress or precharge time is required, the controller will wait until these events have taken place and assert \overline{RAS} on the next positive edge of CLK.

Sometime after the first positive edge of CLK after ALE and \overline{CS} have been asserted, the input \overline{AREQ} must be asserted. In single port applications, once \overline{AREQ} has been asserted, \overline{CS} can be negated. Once \overline{AREQ} is negated, \overline{RAS} and \overline{DTACK} , if programmed, will be negated. If $\overline{ECAS0}$ is asserted during programming, \overline{CAS} will be negated with \overline{AREQ} . If $\overline{ECAS0}$ was negated during programming, a single \overline{CAS} or group of \overline{CASs} will continue to be asserted after \overline{RAS} has been negated given that the appropriate \overline{ECASs} inputs were asserted as shown in *Figure 5b*. This allows the VRAM to have data present on the data out bus while gaining \overline{RAS} precharge time. ALE can stay asserted several periods of CLK. However, ALE must be negated before or during the period of CLK in which \overline{AREQ} is negated.

When performing address pipelining, the ALE input cannot be asserted to start another access until \overline{AREQ} has been asserted for at least one clock period of CLK for the present access.

3.2 ACCESS MODE 1

Access Mode 1, shown in *Figure 6a*, is selected by asserting the input B1 during programming. This mode allows accesses, which are not delayed by precharge, Port B access, VRAM transfer cycle or refresh, to start immediately from the access request input, \overline{ADS} . To initiate a Mode 1 access, \overline{CS} is asserted followed by \overline{ADS} asserted. If the programmed precharge time from the last access or VRAM refresh had been met and a refresh of the VRAM, a Port B access to the VRAM, or a VRAM transfer cycle was not in progress, the \overline{RAS} or group of \overline{RASs} selected by programming and the bank select inputs would be asserted from \overline{ADS} being asserted. If a VRAM refresh, a Port B access, or a VRAM transfer cycle is in progress or precharge time is required, the controller will wait until these events have taken place and assert \overline{RAS} or the group of \overline{RASs} from the next positive edge of CLK.

When \overline{ADS} is asserted or sometime after, \overline{AREQ} must be asserted. At this time, \overline{ADS} can be negated and \overline{AREQ} will continue the access. Once \overline{AREQ} is negated, \overline{RAS} and \overline{DTACK} , if programmed, will be negated. If $\overline{ECAS0}$ was asserted during programming, \overline{CAS} will be negated with \overline{AREQ} . If $\overline{ECAS0}$ was negated during programming, a single \overline{CAS} or group of \overline{CASs} will continue to be asserted after \overline{RAS} has been negated given that the appropriate \overline{ECAS} inputs were asserted as shown in *Figure 6b*. This allows a VRAM to have data present on the data out bus while gaining \overline{RAS} precharge time. \overline{ADS} can continue to be asserted after \overline{AREQ} has been asserted and negated, however a new access would not be started until \overline{ADS} is negated and asserted again. \overline{ADS} and \overline{AREQ} can be tied together in applications not using address pipelining.

If address pipelining is programmed, it is possible for \overline{ADS} to be negated after \overline{AREQ} is asserted. Once \overline{AREQ} is asserted, \overline{ADS} can be asserted again to initiate a new access.

3.0 Port A Access Modes (Continued)

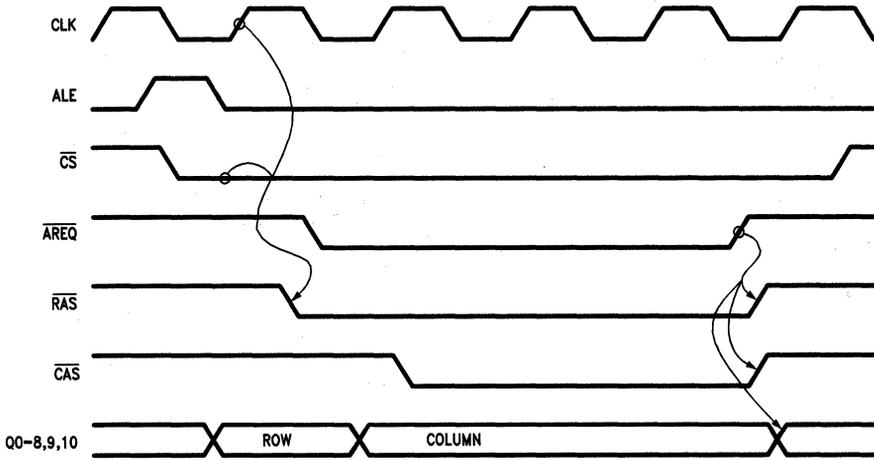


FIGURE 5a. Access Mode 0

TL/F/9338-6

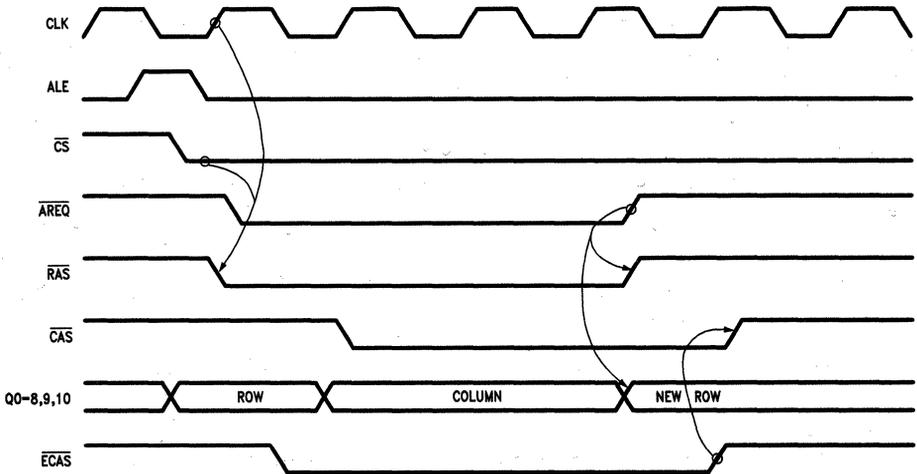


FIGURE 5b. Access Mode 0 Extending $\overline{\text{CAS}}$

TL/F/9338-7

3.0 Port A Access Modes (Continued)

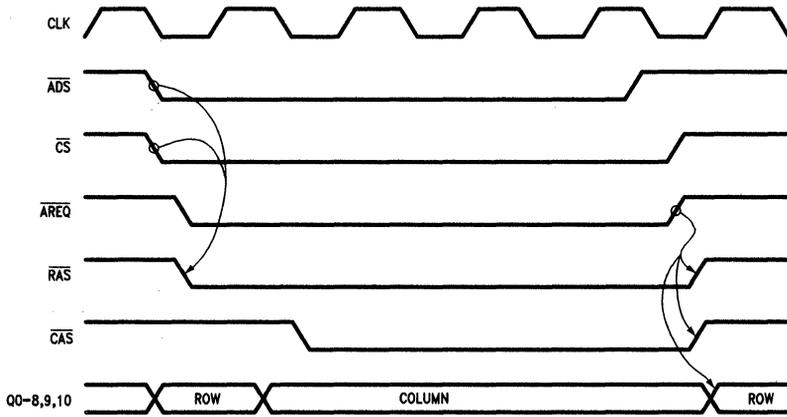


FIGURE 6a. Access Mode 1

TL/F/9338-8

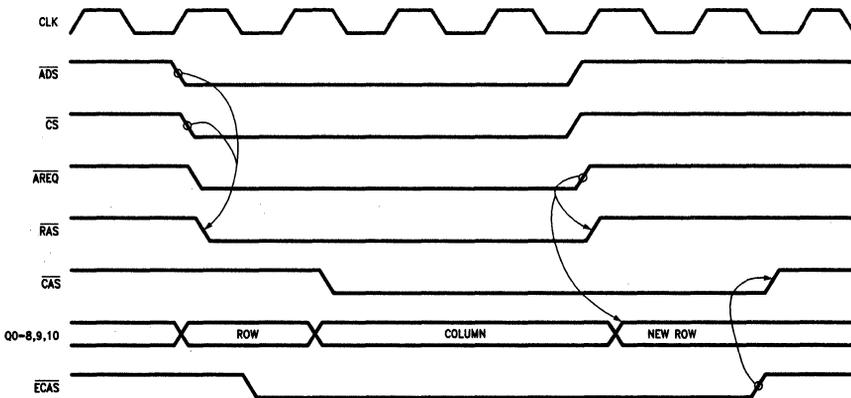


FIGURE 6b. Access Mode 1 Extending CAS

TL/F/9338-9

4.0 Refresh Options

The DP8520A/21A/22A support a wide variety of refresh control mode options including automatic internally controlled refresh, externally controlled/burst refresh, refresh request/acknowledge and any combination of the above. With each of the control modes above, different types of refreshes can be performed. These different types include all $\overline{\text{RAS}}$ refresh, staggered refresh and error scrubbing during all $\overline{\text{RAS}}$ refresh.

There are three inputs, EXTNDRF, $\overline{\text{RFSH}}$ and $\overline{\text{DISRFSH}}$, and one output, RFRQ (RFRQ), associated with refresh. There are also ten programming bits; R0–1, R9, C0–6 and ECAS0 used to program the various types of refreshing.

The two inputs, $\overline{\text{RFSH}}$ and $\overline{\text{DISRFSH}}$, are used in the externally controlled/burst refresh mode and the refresh request/acknowledge mode. The output RFRQ is used in the refresh request/acknowledge mode. The input EXTNDRF is used in all refresh modes and the output RFI $\overline{\text{P}}$ is used in all refresh modes except the refresh request/acknowledge mode. Asserting the input EXTNDRF, extends the refresh cycle single or multiple integral clock periods of CLK. The output RFI $\overline{\text{P}}$ is asserted one period of CLK before the first refresh $\overline{\text{RAS}}$ is asserted. If an access is currently in progress, RFI $\overline{\text{P}}$ will be asserted up to one period of CLK before the first refresh $\overline{\text{RAS}}$, once AREQ or AREQB is negated for the access (see *Figure 7a*).

The DP8520A/21A/22A will increment the refresh address counter automatically, independent of the refresh mode used. The refresh address counter will be incremented once all the refresh $\overline{\text{RAS}}$ s have been negated.

In every combination of refresh control mode and refresh type, the DP8520A/21A/22A is programmed to keep $\overline{\text{RAS}}$ asserted a number of CLK periods. The values of $\overline{\text{RAS}}$ low time during refresh are programmed with the programming bits R0 and R1.

4.1 REFRESH CONTROL MODES

There are three different modes of refresh control. Any of these modes can be used in combination or singularly to produce the desired refresh results. The three different modes of control are: automatic internal refresh, external/burst refresh and refresh request/acknowledge.

4.1.1 Automatic Internal Refresh

The DP8520A/21A/22A have an internal refresh clock. The period of the refresh clock is generated from the programming bits C0–3. Every period of the refresh clock, an internal refresh request is generated. As long as a VRAM access is not currently in progress and precharge time has been met, the internal refresh request will generate an automatic internal refresh. If a VRAM access is in progress, the DP8520A/21A/22A on-chip arbitration logic will wait until the access is finished before performing the refresh. The refresh/access arbitration logic can insert a refresh cycle between two address pipelined accesses. However, the refresh arbitration logic can not interrupt an access cycle to perform a refresh. To enable automatic internally controlled refreshes, the input $\overline{\text{DISRFSH}}$ must be negated.

4.1.2 Externally Controlled/Burst Refresh

To use externally controlled/burst refresh, the user must disable the automatic internally controlled refreshes by asserting the input $\overline{\text{DISRFSH}}$. The user is responsible for generating the refresh request by asserting the input $\overline{\text{RFSH}}$.

Pulsing $\overline{\text{RFSH}}$ low, sets an internal latch, that is used to produce the internal refresh request. The refresh cycle will take place on the next positive edge of CLK as shown in *Figure 7b*. If an access to VRAM is in progress or precharge time for the last access has not been met, the refresh will be delayed. Since pulsing $\overline{\text{RFSH}}$ low sets a latch, the user does not have to keep $\overline{\text{RFSH}}$ low until the refresh starts. When the last refresh $\overline{\text{RAS}}$ negates, the internal refresh request latch is cleared.

By keeping $\overline{\text{RFSH}}$ asserted past the positive edge of CLK which ends the refresh cycle as shown in *Figure 8*, the user will perform another refresh cycle. Using this technique, the user can perform a burst refresh consisting of any number of refresh cycles. Each refresh cycle during a burst refresh will meet the refresh $\overline{\text{RAS}}$ low time and the $\overline{\text{RAS}}$ precharge time (programming bits R0–1).

If the user desires to burst refresh the entire VRAM (all row addresses) he could generate an end of count signal (burst refresh finished) by looking at one of the DP8520A/21A/22A high address outputs (Q7, Q8, Q9 or Q10) and the RFI $\overline{\text{P}}$ output. The Qn outputs function as a decade of how many row addresses have been refreshed (Q7 = 128 refreshes, Q8 = 256 refreshes, Q9 = 512 refreshes, Q10 = 1024 refreshes).

4.1.3 Refresh Request/Acknowledge

The DP8520A/21A/22A can be programmed to output internal refresh requests. When the user programs ECAS0 negated during programming, the RFI $\overline{\text{P}}$ output functions as RFRQ. RFRQ will be asserted from a positive edge of CLK as shown in *Figure 9a*. Once RFRQ is asserted, it will stay asserted until the $\overline{\text{RFSH}}$ is pulsed low with $\overline{\text{DISRFSH}}$ asserted. This will cause an externally requested/burst refresh to take place. If $\overline{\text{DISRFSH}}$ is negated, an automatic internal refresh will take place as shown in *Figure 9b*.

RFRQ will go high and then assert if additional periods of the internal refresh clock have expired and neither an externally controlled refresh nor an automatically controlled internal refresh have taken place as shown in *Figure 9c*. If a time critical event, or long access like page/static column mode access can not be interrupted, RFRQ pulsing high can be used to increment a counter. The counter can be used to perform a burst refresh of the number of refreshes missed (through the $\overline{\text{RFSH}}$ input).

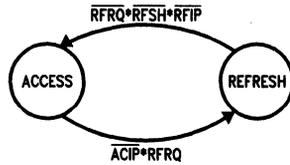
4.2 REFRESH CYCLE TYPES

Three different types of refresh cycles are available for use. The three different types are mutually exclusive and can be used with any of the three modes of refresh control. The three different refresh cycle types are: all $\overline{\text{RAS}}$ refresh, staggered $\overline{\text{RAS}}$ refresh and error scrubbing during all $\overline{\text{RAS}}$ refresh. In all refresh cycle types, the $\overline{\text{RAS}}$ precharge time is guaranteed: between the previous access $\overline{\text{RAS}}$ ending and the refresh $\overline{\text{RAS0}}$ starting; between refresh $\overline{\text{RAS3}}$ ending and access $\overline{\text{RAS}}$ beginning; between burst refresh $\overline{\text{RAS}}$ s.

4.2.1 Conventional $\overline{\text{RAS}}$ Refresh

A conventional refresh cycle causes $\overline{\text{RAS0-3}}$ to all assert from the first positive edge of CLK after RFI $\overline{\text{P}}$ is asserted as shown in *Figure 10*. $\overline{\text{RAS0-3}}$ will stay asserted until the number of positive edges of CLK programmed have passed. On the last positive edge, $\overline{\text{RAS0-3}}$, and RFI $\overline{\text{P}}$ will be negated. This type of refresh cycle is programmed by negating address bit R9 during programming.

4.0 Refresh Options (Continued)

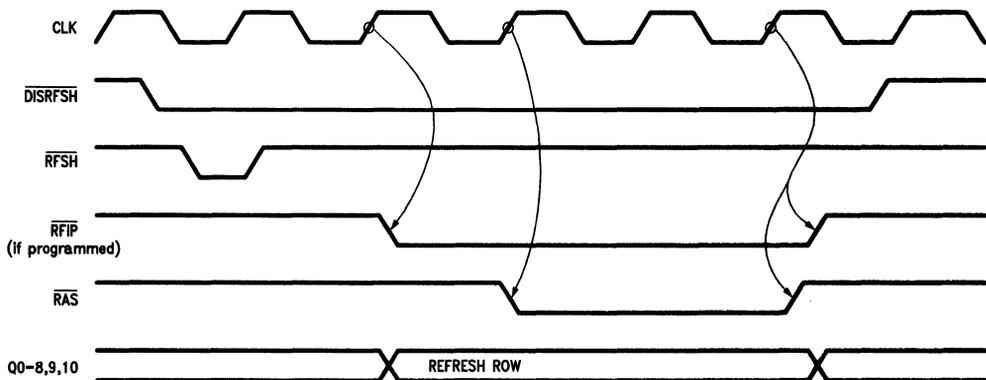


TL/F/9338-10

Explanation of Terms

- RFRQ = ReFresh ReQuest internal to the DP8520A/21A/22A.
RFRQ has the ability to hold off a pending access.
- RFSH = Externally requested ReFrESH
- RFIP = ReFresh In Progress
- ACIP = Port A or Port B (DP8522A only) ACcess In Progress.
This means that either RAS is low for an access or is in the process of transitioning low for an access.

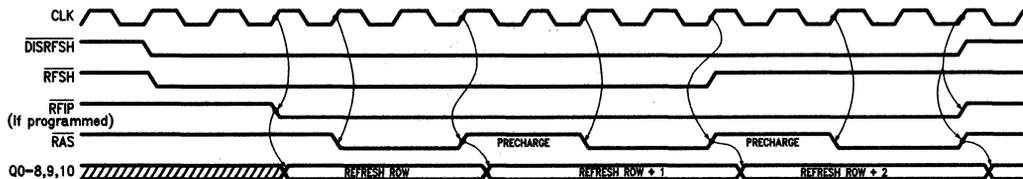
FIGURE 7a. DP8520A/21A/22A Access/Refresh Arbitration State Program



TL/F/9338-11

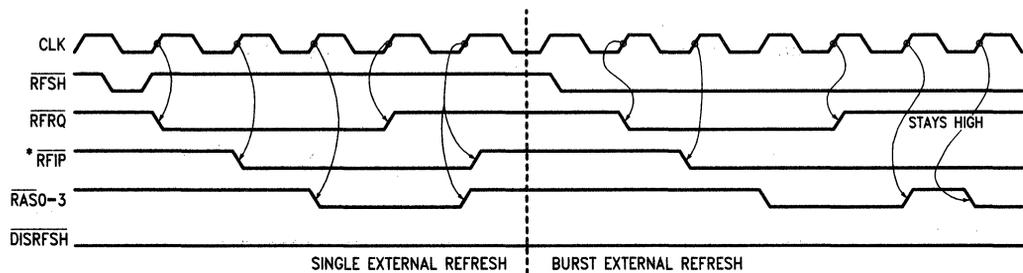
FIGURE 7b. Single External Refresh (2 Periods of \overline{RAS} Low during Refresh Programmed)

4.0 Refresh Options (Continued)



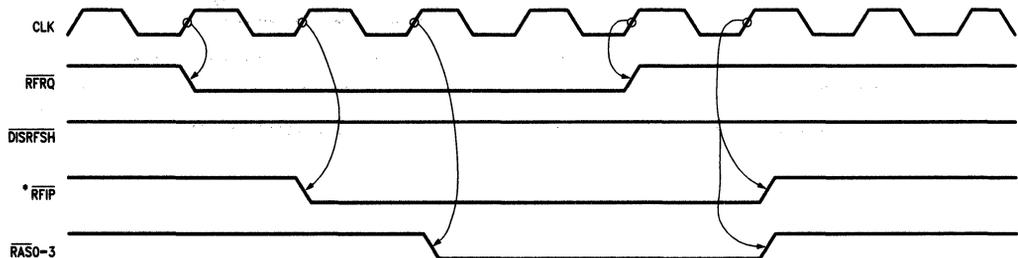
TL/F/9338-12

FIGURE 8. External Burst Refresh (2 Periods of $\overline{\text{RAS}}$ Precharge, 2 Periods of Refresh $\overline{\text{RAS}}$ Low during Refresh Programmed)



TL/F/9338-13

FIGURE 9a. Externally Controlled Single and Burst Refresh with Refresh Request ($\overline{\text{RFRQ}}$) Output (2 Periods of $\overline{\text{RAS}}$ Low during Refresh Programmed)



TL/F/9338-14

FIGURE 9b. Automatic Internal Refresh with Refresh Request (3T of $\overline{\text{RAS}}$ low during refresh programmed)

*In Figures 9a and 9b, where the DP8520A/21A/22A operate in the refresh request/acknowledge mode, the $\overline{\text{RFIP}}$ output pin functions as $\overline{\text{RFRQ}}$. An $\overline{\text{RFIP}}$ timing waveform is included in the figure solely for the purpose of simplifying the diagrams.

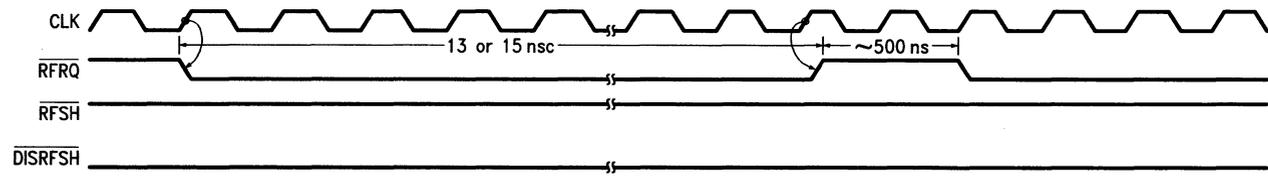


FIGURE 9c. Refresh Request Timing

TL/F/9338-15

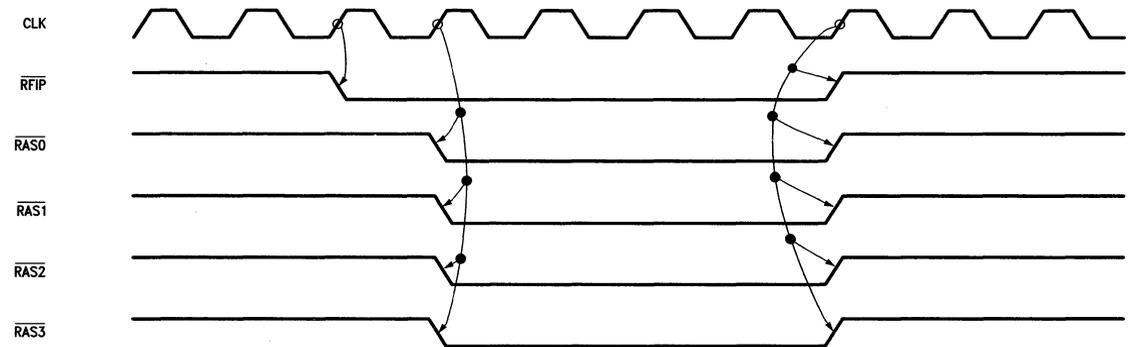


FIGURE 10. Conventional $\overline{\text{RAS}}$ Refresh

TL/F/9338-16

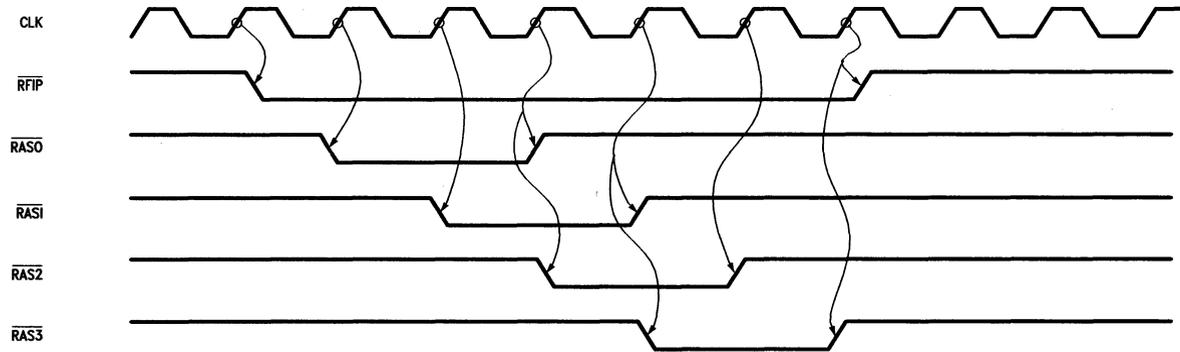


FIGURE 11. Staggered $\overline{\text{RAS}}$ Refresh

TL/F/9338-17

4.0 Refresh Options (Continued)

4.2.2 Staggered $\overline{\text{RAS}}$ Refresh

A staggered refresh staggers each $\overline{\text{RAS}}$ or group of $\overline{\text{RAS}}$ s by a positive edge of CLK as shown in *Figure 11*. The number of $\overline{\text{RAS}}$ s, which will be asserted on each positive edge of CLK, is determined by the $\overline{\text{RAS}}$, CAS configuration mode programming bits C4–C6. If single $\overline{\text{RAS}}$ outputs are selected during programming, then each $\overline{\text{RAS}}$ will assert on successive positive edges of CLK. If two $\overline{\text{RAS}}$ outputs are selected during programming then $\overline{\text{RAS}}_0$ and $\overline{\text{RAS}}_1$ will assert on the first positive edge of CLK after $\overline{\text{RFIP}}$ is asserted. $\overline{\text{RAS}}_2$ and $\overline{\text{RAS}}_3$ will assert on the second positive edge of CLK after $\overline{\text{RFIP}}$ is asserted. If all $\overline{\text{RAS}}$ outputs were selected during programming, all $\overline{\text{RAS}}$ outputs would assert on the first positive edge of CLK after $\overline{\text{RFIP}}$ is asserted. Each $\overline{\text{RAS}}$ or group of $\overline{\text{RAS}}$ s will meet the programmed RAS low time and then negate.

4.2.3 Error Scrubbing during Refresh

The DP8520A/21A/22A support error scrubbing during all $\overline{\text{RAS}}$ VRAM refreshes. Error scrubbing during refresh is selected through bits C4–C6 with bit R9 negated during programming. Error scrubbing can not be used with staggered refresh (see Section 9.0). Error scrubbing during refresh al-

lows a $\overline{\text{CAS}}$ or group of $\overline{\text{CAS}}$ s to assert during the all $\overline{\text{RAS}}$ refresh as shown in *Figure 12*. This allows data to be read from the VRAM array and passed through an Error Detection And Correction Chip, EDAC. It is important to note that while an error scrubbing during refresh access is being performed, it is the system designer's responsibility to properly control the $\overline{\text{WE}}$ input of the VRAM. $\overline{\text{WE}}$ should be high during the initial access of the VRAM, which could be accomplished by gating $\overline{\text{RFIP}}$, if programmed, with the processor access circuitry that creates $\overline{\text{WE}}$. If the EDAC determines that the data contains a single bit error and corrects that error, the refresh cycle can be extended with the input extend refresh, EXTNDRF, and a read-modify-write operation can be performed, and the corrected data can be written back to the VRAM by bringing $\overline{\text{WE}}$ low. The DP8522A has a 24-bit internal refresh address counter that contains the 11 row, 11 column and 2 bank addresses. The DP8520A/21A have a 22-bit internal refresh address counter that contains the 10 row, 10 column and 2 bank addresses. These counters are configured as bank, column, row with the row address as the least significant bits. The bank counter bits are then used with the programming selection to determine which CAS or group of CASs will assert during a refresh.

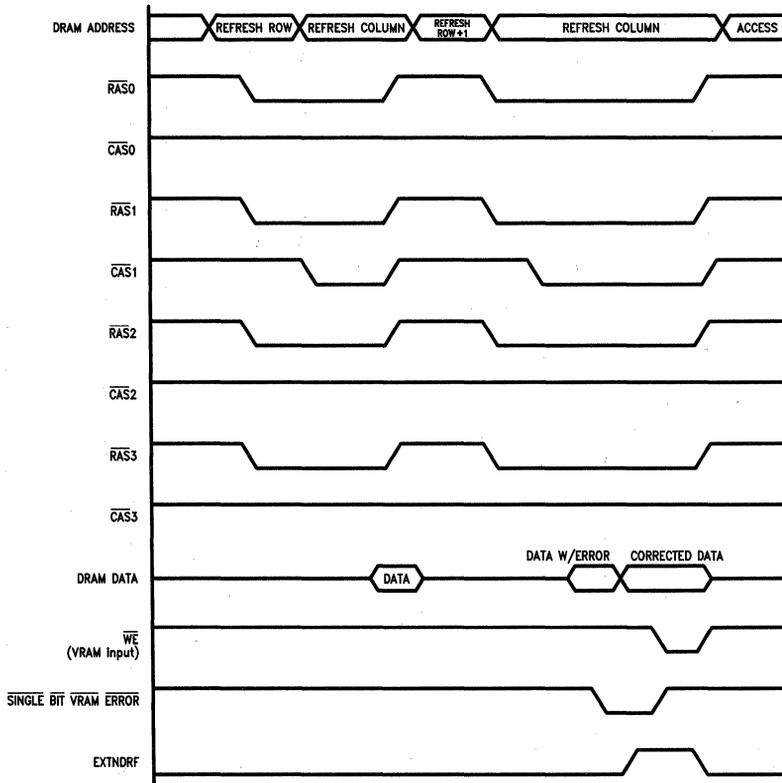


FIGURE 12. Error Scrubbing during Refresh

TL/F/9398-18

4.0 Refresh Options (Continued)

4.3 EXTENDING REFRESH

The programmed number of periods of CLK that refresh RASs are asserted can be extended by one or multiple periods of CLK. Only the all RAS (with or without error scrubbing) type of refresh can be extended. To extend a refresh cycle, the input extend refresh, EXTNDRF, must be asserted before the positive edge of CLK that would have negated all the RAS outputs during the refresh cycle and after the positive edge of CLK which starts all RAS outputs during the refresh as shown in Figure 13. This will extend the refresh to the next positive edge of CLK and EXTNDRF will be sampled again. The refresh cycle will continue until EXTNDRF is sampled low on a positive edge of CLK.

4.4 CLEARING THE REFRESH ADDRESS COUNTER

The refresh address counter can be cleared by asserting RFSH while DISRFSH is negated as shown in Figure 14a. This can be used prior to a burst refresh of the entire memo-

ry array. By asserting RFSH one period of CLK before DISRFSH is asserted and then keeping both inputs asserted, the DP8520A/21A/22A will clear the refresh address counter and then perform refresh cycles separated by the programmed value of precharge as shown in Figure 14b. An end-of-count signal can be generated from the Q VRAM address outputs of the DP8520A/21A/22A and used to negate RFSH.

4.5 CLEARING THE REFRESH REQUEST CLOCK

The refresh request clock can be cleared by negating DISRFSH and asserting RFSH for 500 ns, one period of the internal 2 MHz clock as shown in Figure 15. By clearing the refresh request clock, the user is guaranteed that an internal refresh request will not be generated for approximately 15 μ s, one refresh clock period, from the time RFSH is negated. This action will also clear the refresh address counter.

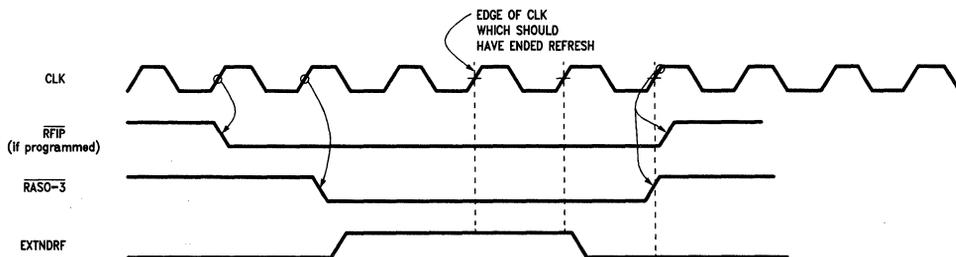


FIGURE 13. Extending Refresh with the Extend Refresh (EXTNDRF) Input

TL/F/9338-19

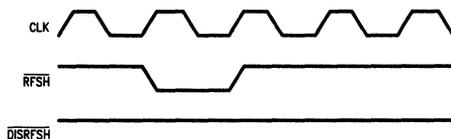


FIGURE 14a. Clearing the Refresh Address Counter

TL/F/9338-20

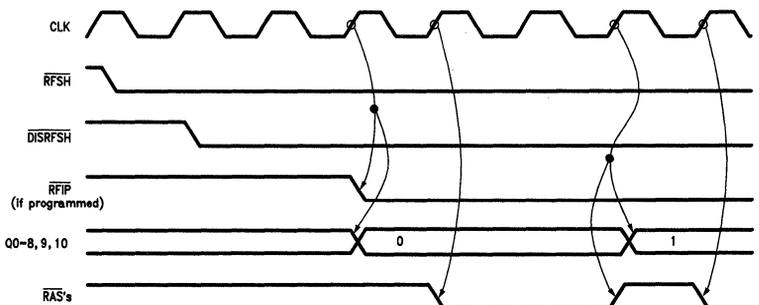


FIGURE 14b. Clearing the Refresh Counter during Burst

TL/F/9338-21



FIGURE 15. Clearing the Refresh Request Clock Counter

TL/F/9338-22

5.0 Port A Wait State Support

Wait states allow a CPU's access cycle to be increased by one or multiple CPU clock periods. By increasing the CPU's access cycle, all signals associated with that access cycle are extended. The wait or ready input is named differently by CPU manufacturers. However, any CPU's wait or ready input is compatible with either the $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$ output of the DP8520A/21A/22A. The CPU samples a wait or ready line to determine if another clock period should be inserted into the access cycle. If another clock period is inserted, the CPU will continue to sample the input every CPU clock period until the input signal changes polarity, allowing the CPU access cycle to terminate. The user determines whether to program $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$ (R7) and which value to select for $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$ (R2, R3) depending upon the CPU used and where the CPU samples its wait input during an access cycle.

The decision to terminate the CPU access cycle is directly affected by the speed of the VRAMs used. The system designer must ensure that the data from the VRAMs will be present for the CPU to sample or that the data has been written to the VRAM before allowing the CPU access cycle to terminate.

The insertion of wait states also allows a CPU's access cycle to be extended until the VRAM access has taken place. The DP8520A/21A/22A insert wait states into CPU access cycles due to; guaranteeing precharge time, refresh currently in progress, user programmed wait states, the $\overline{\text{WAITIN}}$ signal being asserted and $\overline{\text{GRANTB}}$ not being valid (DP8522A only). If one of these events is taking place and the CPU starts an access, the DP8520A/21A/22A will insert wait states into the access cycle, thereby increasing the length of the CPU's access. Once the event has been completed, the DP8520A/21A/22A will allow the access to take place and stop inserting wait states.

There are six programming bits, R2–R7; an input, $\overline{\text{WAITIN}}$; and an output that functions as $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$.

5.1 WAIT TYPE OUTPUT

With the R7 address bit negated during programming, the user selects the $\overline{\text{WAIT}}$ output. As long as $\overline{\text{WAIT}}$ is sampled asserted by the CPU, wait states (extra clock periods) are inserted into the current access cycle as shown in Figure 16. Once $\overline{\text{WAIT}}$ is sampled negated, the access cycle is completed by the CPU. $\overline{\text{WAIT}}$ is asserted at the beginning of a chip selected access and is programmed to negate a number of positive edges and/or negative levels of CLK from the event that starts the access. $\overline{\text{WAIT}}$ can also be programmed to function in page/burst mode applications.

Once $\overline{\text{WAIT}}$ is negated during an access, and the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{AREQ}}$ asserted, $\overline{\text{WAIT}}$ can be programmed to toggle, following the $\overline{\text{ECAS}}$ inputs. Once $\overline{\text{AREQ}}$ is negated, ending the access, $\overline{\text{WAIT}}$ will stay negated until the next chip selected access.

5.1.1. Wait during Single Accesses

$\overline{\text{WAIT}}$ can be programmed to delay a number of positive edges and/or negative levels of CLK. These options are programmed through address bits R2 and R3 at programming time. The user is given four options described below.

0T during non delayed and delayed accesses: $\overline{\text{WAIT}}$ will stay negated during a non-delayed access as shown in Figures 17a and 17c. During an access that is delayed, $\overline{\text{WAIT}}$ will assert at the start of the access ($\overline{\text{CS}}$ and $\overline{\text{ALE}}$ or $\overline{\text{ADS}}$) and negate from the positive edge of CLK that starts $\overline{\text{RAS}}$ for that access as shown in Figures 17b and 17d.

0T during non-delayed accesses and $\frac{1}{2}T$ during delayed accesses: $\overline{\text{WAIT}}$ will stay negated during a non-delayed access as shown in Figures 18a and 18c. During an access that is delayed, $\overline{\text{WAIT}}$ will assert at the start of the access $\overline{\text{CS}}$ and $\overline{\text{ALE}}$ or $\overline{\text{ADS}}$ and negate on the negative level of CLK after the positive edge of CLK that asserted $\overline{\text{RAS}}$ for that access as shown in Figures 18b and 18d.

$\frac{1}{2}T$ during non-delayed and delayed accesses: if mode 0 is used, $\overline{\text{WAIT}}$ will assert when $\overline{\text{ALE}}$ is asserted and $\overline{\text{CS}}$ is asserted. $\overline{\text{WAIT}}$ will then negate on the negative level of CLK after the positive edge of CLK that asserts $\overline{\text{RAS}}$ for the access as shown in Figure 19a. In Mode 1, $\overline{\text{WAIT}}$ will assert from $\overline{\text{CS}}$ asserted and $\overline{\text{ADS}}$ asserted. $\overline{\text{WAIT}}$ will then negate on the negative level of CLK after $\overline{\text{RAS}}$ has been asserted for the access as shown in Figure 19c. During delayed accesses in both modes, $\overline{\text{WAIT}}$ will assert at the start of the access and negate on the negative level of CLK after the positive edge of CLK that started $\overline{\text{RAS}}$ for that access as shown in Figures 19b and 19d.

1T during non-delayed and delayed accesses. In Mode 0, $\overline{\text{WAIT}}$ will assert from $\overline{\text{ALE}}$ asserted and $\overline{\text{CS}}$ asserted. $\overline{\text{WAIT}}$ will negate from the next positive edge of CLK that asserts $\overline{\text{RAS}}$ for the access as shown in Figure 20a. In Mode 1, $\overline{\text{WAIT}}$ will assert from $\overline{\text{ADS}}$ asserted and $\overline{\text{CS}}$ asserted. $\overline{\text{WAIT}}$ will negate from the first positive edge of CLK after $\overline{\text{ADS}}$ and $\overline{\text{CS}}$ have been asserted as shown in Figure 20c. During delayed accesses in both modes, $\overline{\text{WAIT}}$ will assert at the beginning of the access and will negate on the first positive edge of CLK after the positive edge of CLK that starts $\overline{\text{RAS}}$ for the access as shown in Figures 20b and 20d.

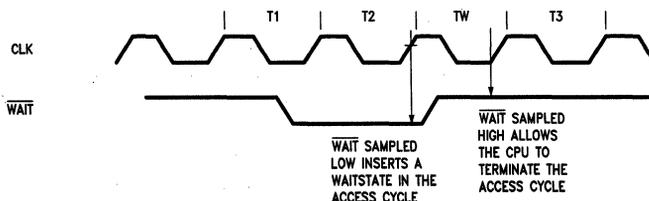
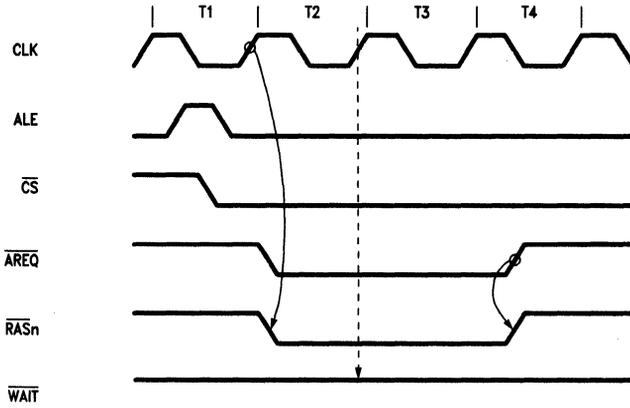


FIGURE 16. $\overline{\text{WAIT}}$ Type Output

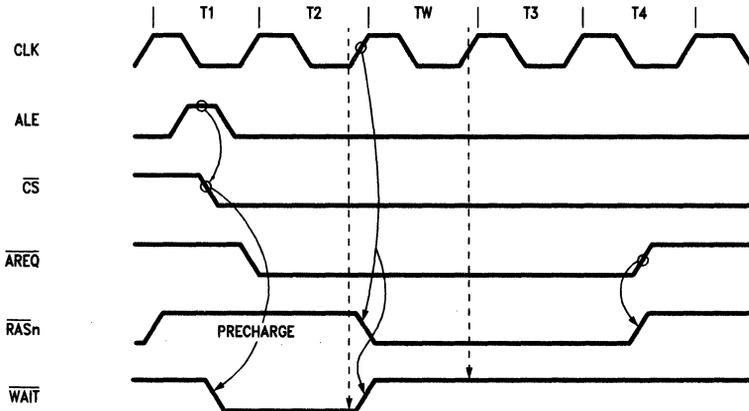
TL/F/9338-23

5.0 Port A Wait State Support (Continued)



TL/F/9338-24

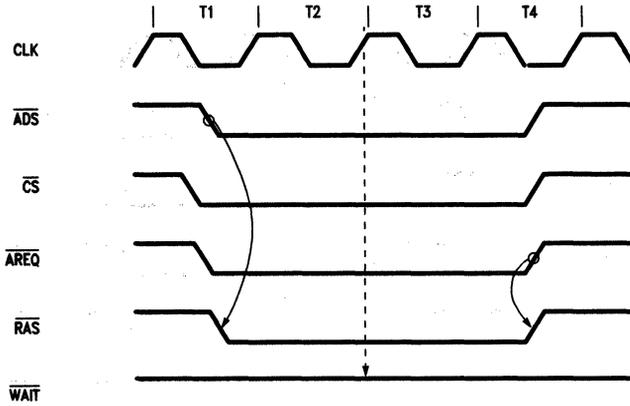
FIGURE 17a. Mode 0 Non-Delayed Access with $\overline{\text{WAIT}} 0T$ ($\overline{\text{WAIT}}$ is Sampled at the End of the "T2" Clock State)



TL/F/9338-25

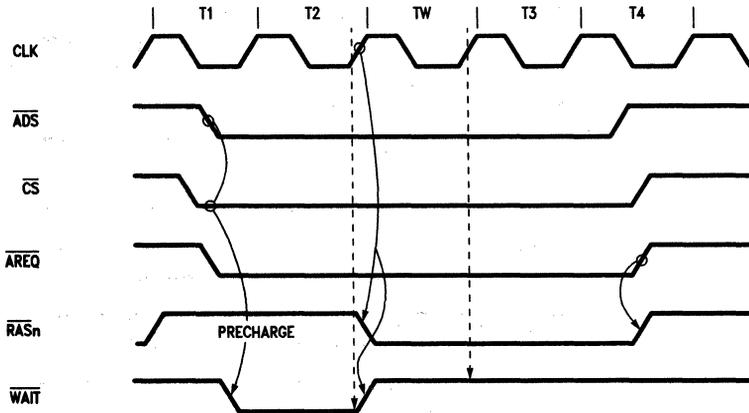
FIGURE 17b. Mode 0 Delayed Access with $\overline{\text{WAIT}} 0T$
("2T" $\overline{\text{RASn}}$ Precharge, $\overline{\text{WAIT}}$ is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)



TL/F/9338-26

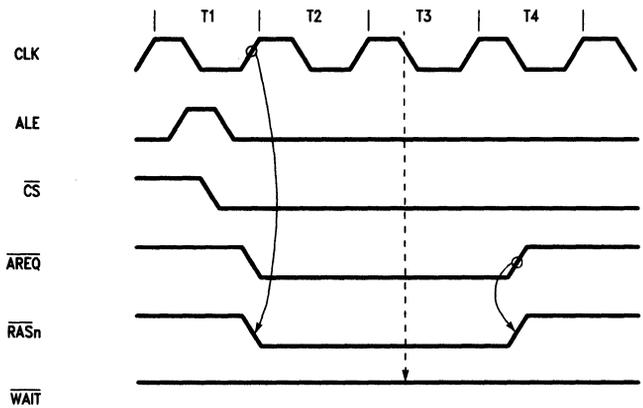
FIGURE 17c. Mode 1 Non-Delayed Access with $\overline{\text{WAIT}} = 0T$ ($\overline{\text{WAIT}}$ is Sampled at the End of the "T2" Clock State)



TL/F/9338-27

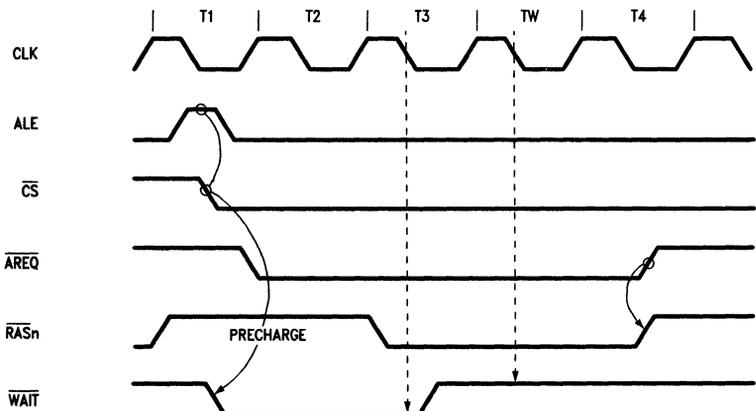
FIGURE 17d. Mode 1 Delayed Access with $\overline{\text{WAIT}} = 0T$ ($\overline{\text{WAIT}}$ is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)



TL/F/9338-28

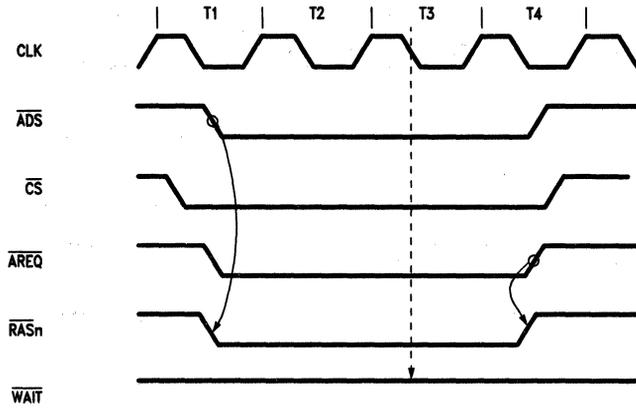
FIGURE 18a. Mode 0 Non-Delayed Access with $\overline{\text{WAIT}} 0T$ ($\overline{\text{WAIT}}$ is Sampled at the "T3" Falling Clock Edge)



TL/F/9338-29

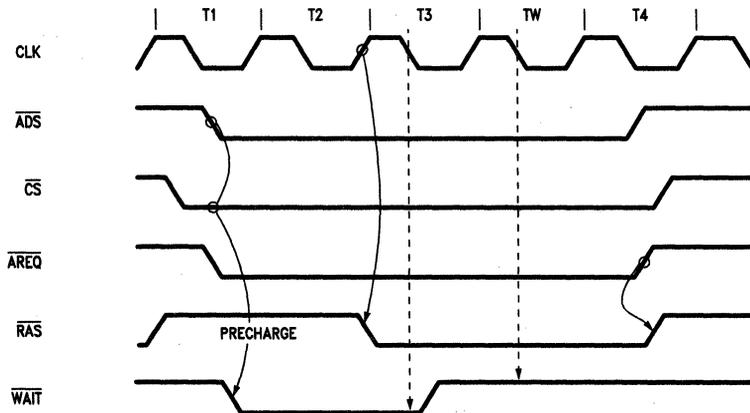
FIGURE 18b. Mode 0 Delayed Access with $\overline{\text{WAIT}} \frac{1}{2}T$ ($\overline{\text{WAIT}}$ is Sampled at the "T3" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



TL/F/9338-30

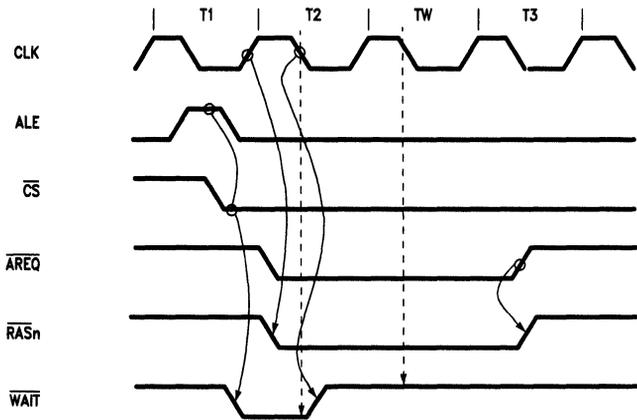
FIGURE 18c. Mode 1 Non-Delayed Access with $\overline{\text{WAIT}} 0T$ ($\overline{\text{WAIT}}$ is Sampled at the "T3" Falling Clock Edge)



TL/F/9338-31

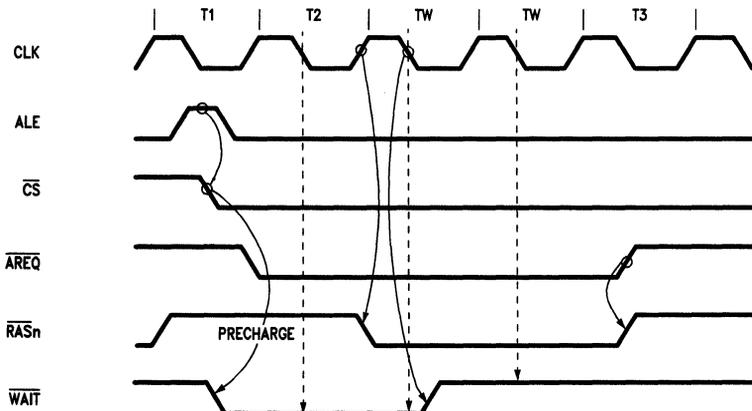
FIGURE 18d. Mode 1 Delayed Access with $\overline{\text{WAIT}} \frac{1}{2}T$ ($\overline{\text{WAIT}}$ is Sampled at the "T3" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



TL/F/9338-32

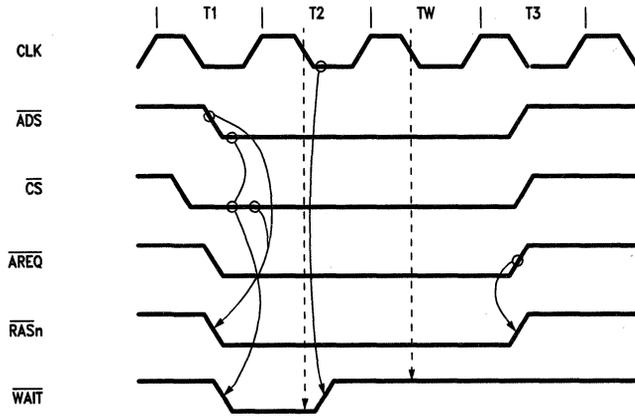
FIGURE 19a. Mode 0 Non-Delayed Access with $\overline{\text{WAIT}} \frac{1}{2}T$ ($\overline{\text{WAIT}}$ is Sampled at the "T2" Falling Clock Edge)



TL/F/9338-33

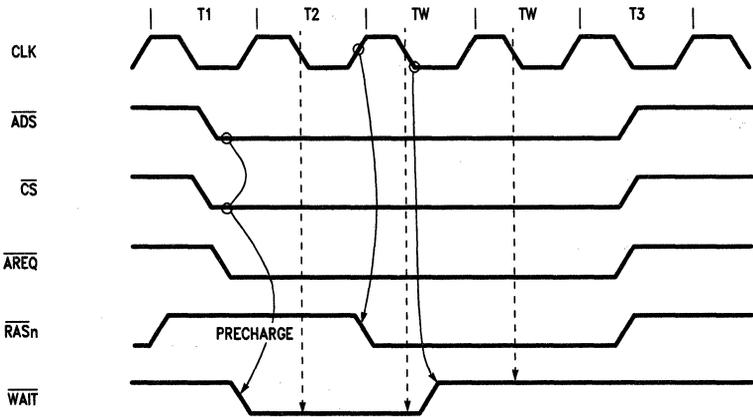
FIGURE 19b. Mode 0 Delayed Access with $\overline{\text{WAIT}} \frac{1}{2}T$ ($\overline{\text{WAIT}}$ is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



TL/F/9338-34

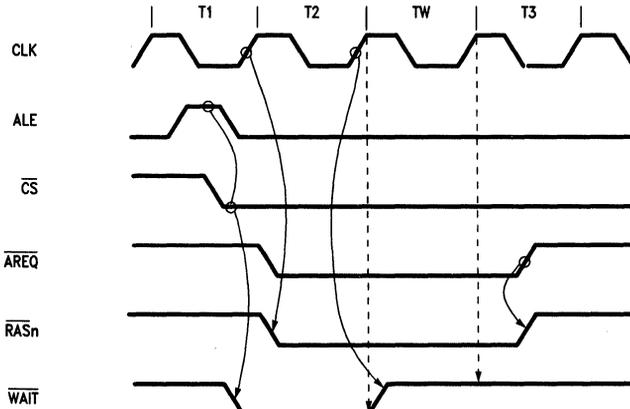
FIGURE 19c. MODE 1 Non-Delayed Access with WAIT $\frac{1}{2}T$ (WAIT is Sampled at the "T2" Falling Clock Edge)



TL/F/9338-35

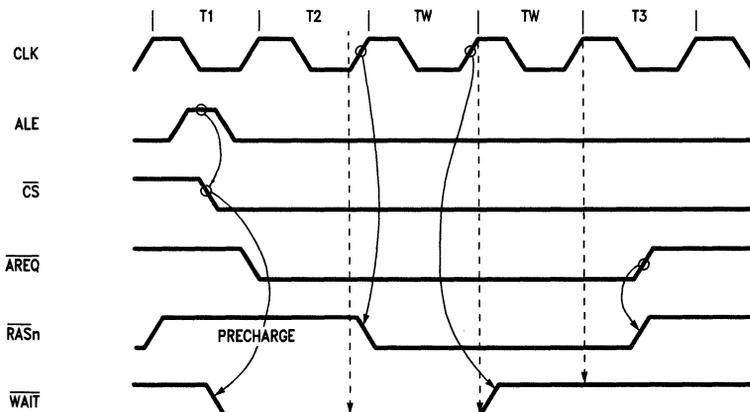
FIGURE 19d. Mode 1 Delayed Access with WAIT $\frac{1}{2}T$ (WAIT is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



TL/F/9338-36

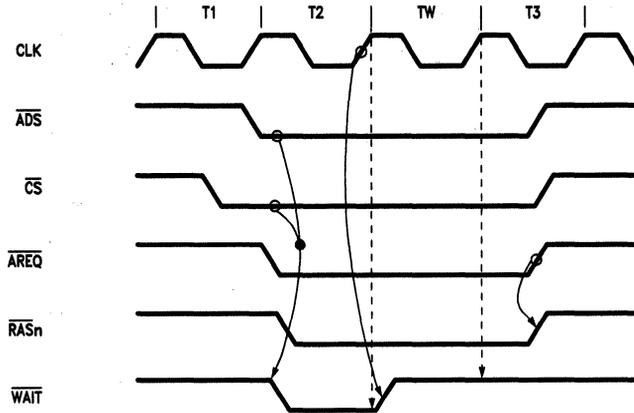
FIGURE 20a. Mode 0 Non-Delayed Access with $\overline{\text{WAIT}} 1\text{T}$ ($\overline{\text{WAIT}}$ is Sampled at the End of the "T2" Clock State)



TL/F/9338-37

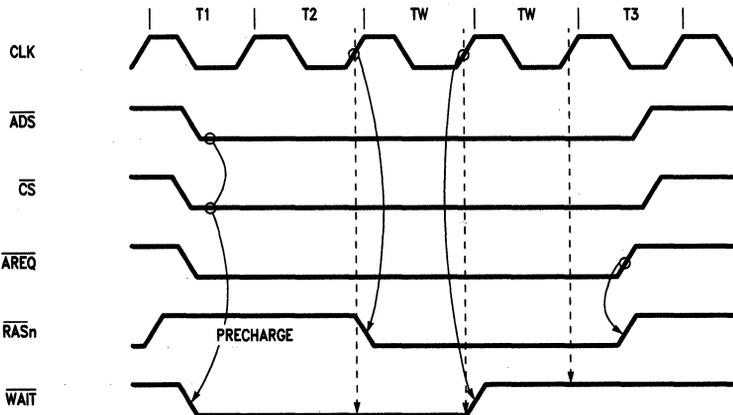
FIGURE 20b. Mode 0 Delayed Access with $\overline{\text{WAIT}} 1\text{T}$ ($\overline{\text{WAIT}}$ is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)



TL/F/9338-38

FIGURE 20c. Mode 1 Non-Delayed Access with \overline{WAIT} 1T (\overline{WAIT} is Sampled at the End of the "T2" Clock State)



TL/F/9338-39

FIGURE 20d. Mode 1 Delayed Access with \overline{WAIT} 1T (\overline{WAIT} is Sampled at the End of the "T2" Clock State)

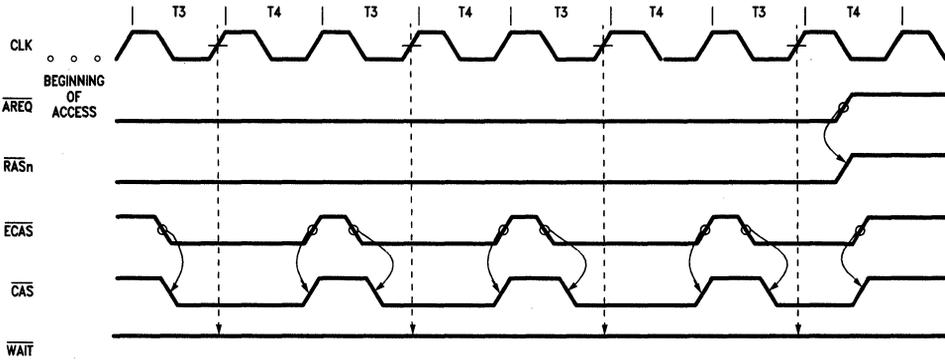
When ending \overline{WAIT} from a negative level of CLK; if \overline{RAS} is asserted while CLK is high then \overline{WAIT} will negate from the negative edge of CLK; if \overline{RAS} is asserted while CLK is low then \overline{WAIT} will negate from \overline{RAS} asserting. When ending \overline{WAIT} from a positive edge of CLK in Mode 0, the user can think of the positive edge of CLK that starts \overline{RAS} as 0T and the next positive edge of CLK as 1T. When ending \overline{WAIT} from a positive edge of CLK in Mode 1, the positive edge of CLK that \overline{ADS} is setup to can be thought of as 1T in a non-delayed access. In a delayed access, the positive edge of CLK that starts \overline{RAS} can be thought of as 0T and the next positive edge as 1T.

5.1.2 Wait during Page Burst Accesses

\overline{WAIT} can be programmed to function differently during page/burst types of accesses. During a page/burst access, the ECAS inputs will be asserted then negated while \overline{AREQ} is asserted. Through address bits R4 and R5, \overline{WAIT} can be programmed to assert and negate during this type of access. The user is given four programming options described below.

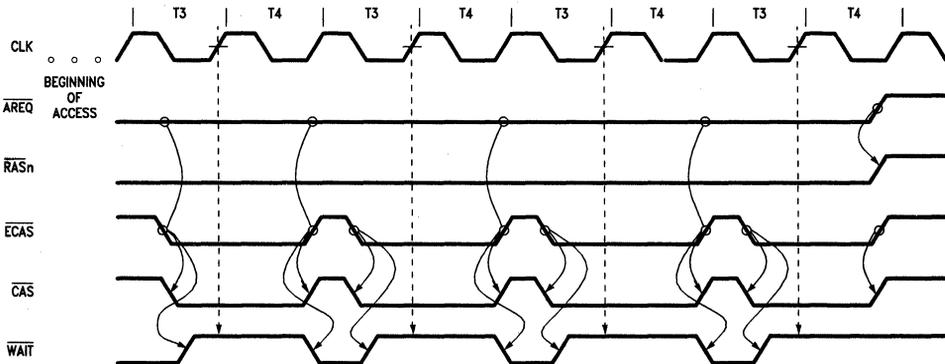
No Wait States: In this case, \overline{WAIT} will remain negated even if the ECAS inputs are toggled as shown in Figure 21.

5.0 Port A Wait State Support (Continued)



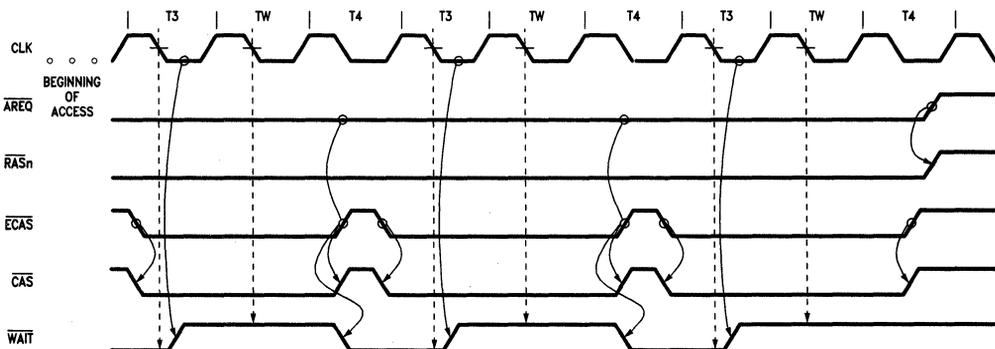
TL/F/9338-40

FIGURE 21. No Wait States during Burst (WAIT is Sampled at the End of the "T3" Clock State)



TL/F/9338-41

FIGURE 22. 0T during Burst (WAIT is Sampled at the End of the "T3" Clock State)



TL/F/9338-42

FIGURE 23. 1/2T during Burst Access (WAIT is Sampled at the "T3" Falling Clock Edge)

0T: WAIT will be asserted when the ECAS inputs are negated with AREQ remaining asserted. When one or both of the ECAS inputs are asserted, WAIT will be negated as shown in Figure 22.

1/2T: WAIT will be asserted when the ECAS inputs are negated with AREQ remaining asserted. When one or both of the ECAS inputs are asserted again, WAIT will be negated from the first negative level of CLK after one or both of the ECASs are asserted as shown in Figure 23.

5.0 Port A Wait State Support (Continued)

1T: \overline{WAIT} will be asserted when the \overline{ECAS} inputs are negated with \overline{AREQ} remaining asserted. When one or both of the \overline{ECAS} inputs are asserted again, \overline{WAIT} will be negated from the first positive edge of CLK after a single \overline{ECAS} or both of the \overline{ECAS} are asserted as shown in Figure 24.

When ending \overline{WAIT} from a negative level of CLK; if the \overline{ECAS} s are asserted while CLK is high then \overline{WAIT} will negate from the negative edge of CLK, if the \overline{ECAS} s are asserted while CLK is low then \overline{WAIT} will negate from the \overline{ECAS} s asserting. When ending \overline{WAIT} from a positive edge of CLK, the positive edge of CLK that \overline{ECAS} is setup to can be thought of as 1T.

5.2 DTACK TYPE OUTPUT

With the R7 address bit asserted during programming, the user selects the \overline{DTACK} type output. As long as \overline{DTACK} is sampled negated by the CPU, wait states are inserted into the current access cycle as shown in Figure 25. Once \overline{DTACK} is sampled asserted, the access cycle is completed by the CPU. \overline{DTACK} , which is normally negated, is programmed to assert a number of positive edges and/or negated

active levels from the event that starts \overline{RAS} for the access. \overline{DTACK} can also be programmed to function during page/burst mode accesses. Once \overline{DTACK} is asserted and the \overline{ECAS} inputs are negated with \overline{AREQ} asserted, \overline{DTACK} can be programmed to negate and assert from the \overline{ECAS} inputs toggling to perform a page/burst mode operation. Once \overline{AREQ} is negated, ending the access, \overline{DTACK} will be negated and stays negated until the next chip selected access.

5.2.1 DTACK during Single Accesses

\overline{DTACK} can be programmed to delay a number of positive edges and/or negative levels of CLK. These options are programmed through address bits R2 and R3 at programming time. The user is given four options described by the following.

0T during non-delayed accesses and delayed accesses: in Mode 0, \overline{DTACK} will assert from the positive edge of CLK which starts \overline{RAS} as shown in Figure 26a. In Mode 1, \overline{DTACK} will assert from \overline{ADS} and \overline{CS} as shown in Figure 26c. During delayed accesses in both modes, \overline{DTACK} will assert from the positive edge of CLK which starts \overline{RAS} for the access as shown in Figure 26b and 26d.

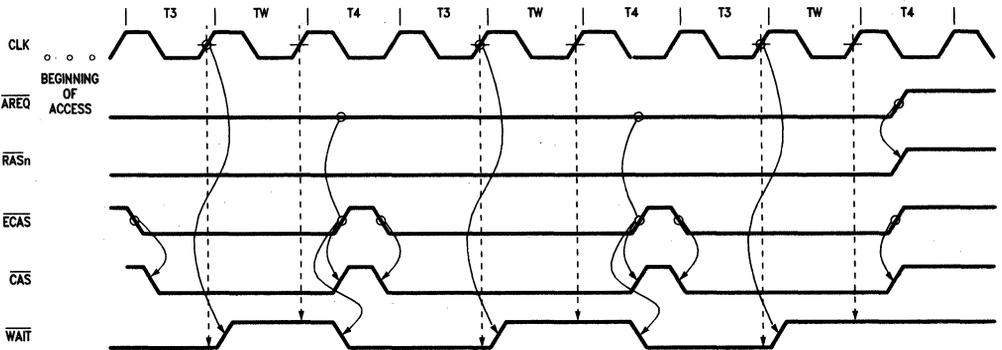


FIGURE 24. 1T during Burst Access (\overline{WAIT} is Sampled at the End of the "T3" Clock State)

TL/F/9338-43

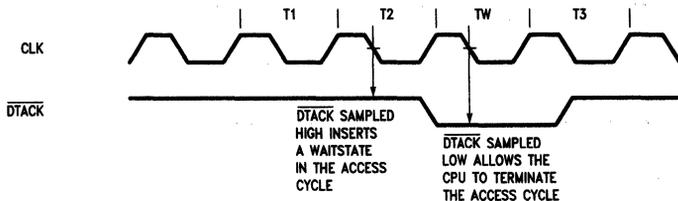
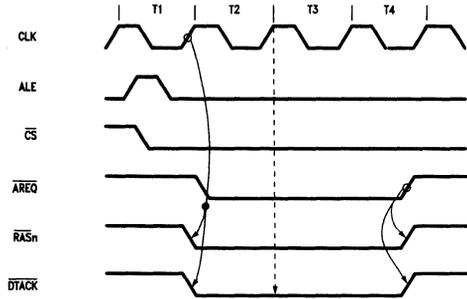


FIGURE 25. \overline{DTACK} Type Output

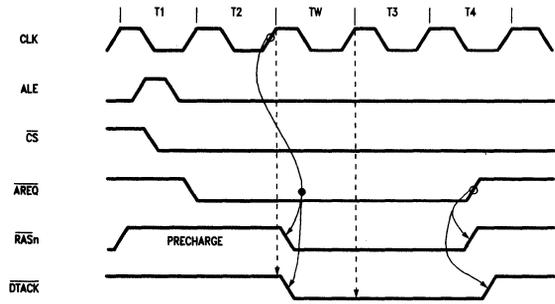
TL/F/9338-44

5.0 Port A Wait State Support (Continued)



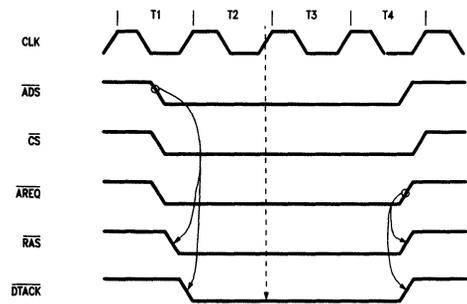
TL/F/9338-45

FIGURE 26a. Mode 0 Non-Delayed Access with \overline{DTACK} 0T (\overline{DTACK} is Sampled at the End of the "T2" Clock State)



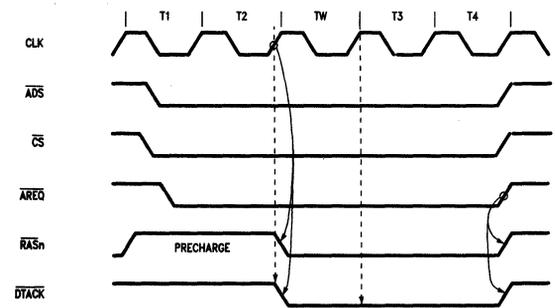
TL/F/9338-46

FIGURE 26b. Mode 0 Delayed Access with \overline{DTACK} 0T (2T Clock Periods Are Programmed for \overline{RAS} Precharge, \overline{DTACK} is Sampled at the End of the "T2" Clock State)



TL/F/9338-47

FIGURE 26c. Mode 1 Non-Delayed Access with \overline{DTACK} 0T (\overline{DTACK} is Sampled at the End of the "T2" Clock State)



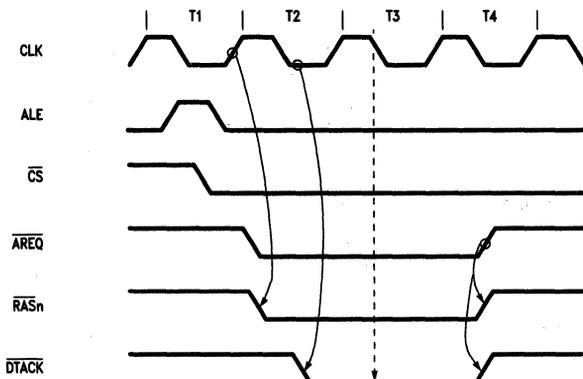
TL/F/9338-48

FIGURE 26d. Mode 1 Delayed Access with \overline{DTACK} 0T (\overline{DTACK} is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)

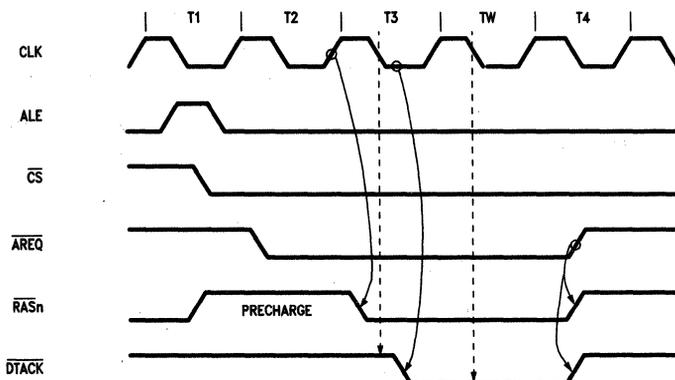
$\frac{1}{2}T$ during non-delayed and delayed accesses: In Mode 0, \overline{DTACK} will assert on the negative level of CLK after the positive edge of CLK which starts \overline{RAS} as shown in Figure 27a. In Mode 1, \overline{DTACK} will assert from the negative level of CLK after \overline{ADS} has been asserted given that \overline{RAS} is assert-

ed as shown in Figures 27c and 27d. During delayed accesses in both modes, \overline{DTACK} will assert from the negative level of CLK after the positive edge of CLK which starts \overline{RAS} for the access as shown in Figures 27b and 27e.



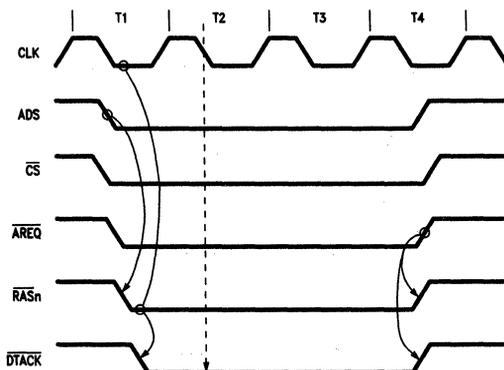
TL/F/9338-49

FIGURE 27a. Mode 0 Non-Delayed Access with \overline{DTACK} of $\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T3" Falling Clock Edge)



TL/F/9338-50

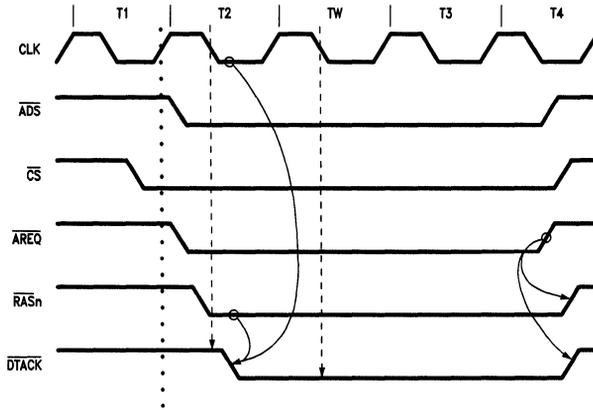
FIGURE 27b. Mode 0 Delayed Access with \overline{DTACK} of $\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T3" Falling Clock Edge)



TL/F/9338-51

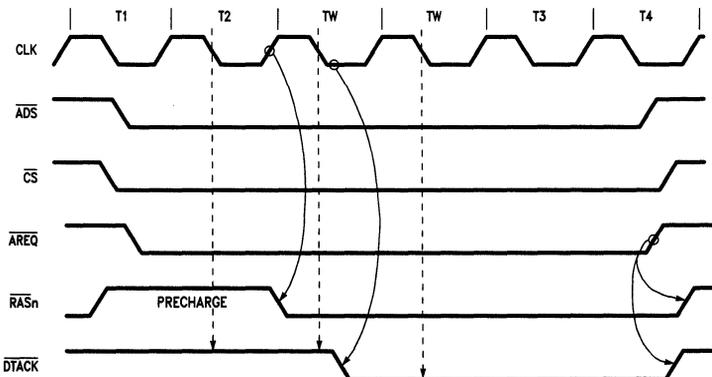
FIGURE 27c. Mode 1 Non-Delayed Access with \overline{DTACK} of $\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



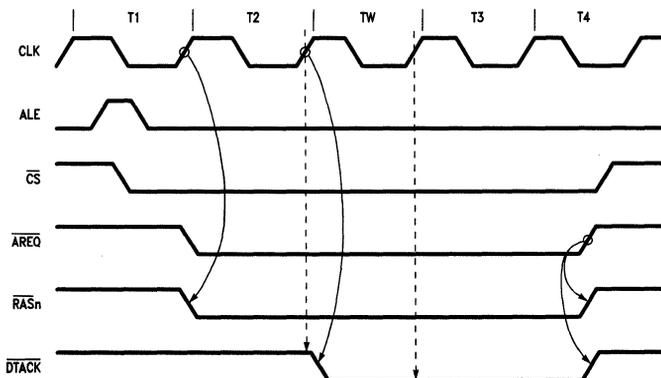
TL/F/9338-58

FIGURE 27d. Mode 1 Non-Delayed Access with \overline{DTACK} of $\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)



TL/F/9338-59

FIGURE 27e. Mode 1 Delayed Access with \overline{DTACK} of $\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)



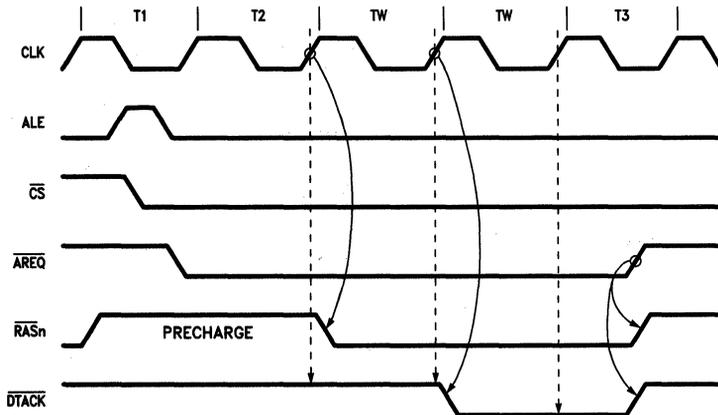
TL/F/9338-60

FIGURE 28a. Mode 0 Non-Delayed Access with \overline{DTACK} of $1T$ (\overline{DTACK} is Sampled at the End of the "T2" Clock State)

$1T$ during delayed and non-delayed accesses: In Mode 0, \overline{DTACK} will assert from the first positive edge of CLK after the positive edge of CLK which starts \overline{RAS} for the access as shown in Figure 28a. In Mode 1, \overline{DTACK} will assert from the

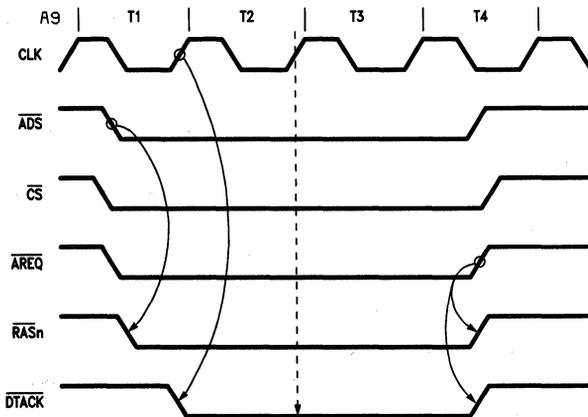
positive edge CLK after \overline{ADS} and \overline{CS} are asserted as shown in Figures 28c and 28d. During delayed accesses in both modes, \overline{DTACK} will assert from the first positive edge of CLK after the positive edge of CLK which starts \overline{RAS} for the access as shown in Figures 28b and 28e.

5.0 Port A Wait State Support (Continued)



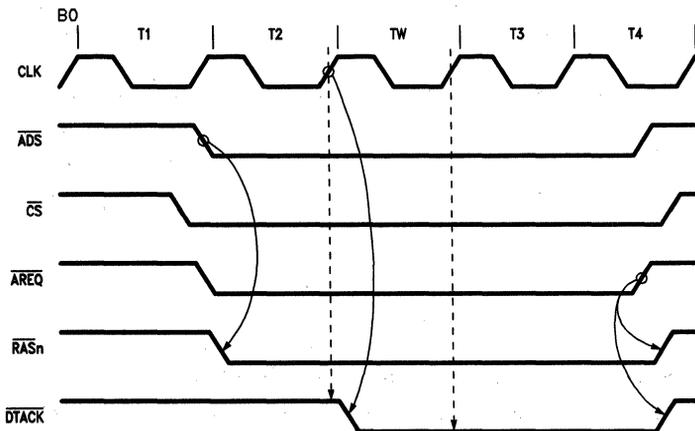
TL/F/9338-61

FIGURE 28b. Mode 0 Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)



TL/F/9338-62

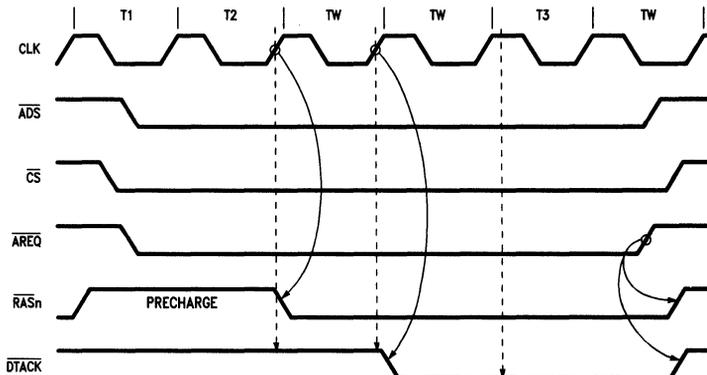
FIGURE 28c. Mode 1 Non-Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)



TL/F/9338-63

FIGURE 28d. Mode 1 Late Non-Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)

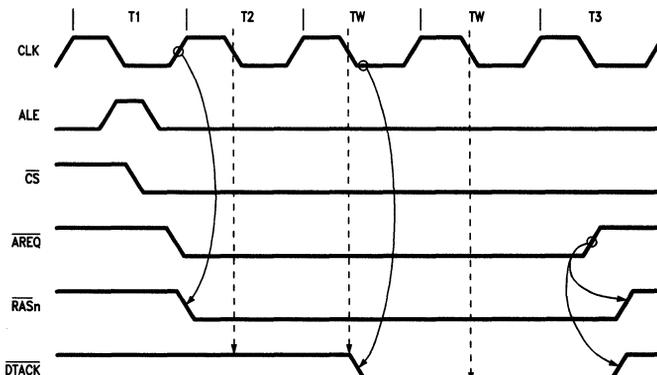


TL/F/9338-64

FIGURE 28e. Mode 1 Delayed Access with \overline{DTACK} of 1T (\overline{DTACK} is Sampled at the End of the "T2" Clock State)

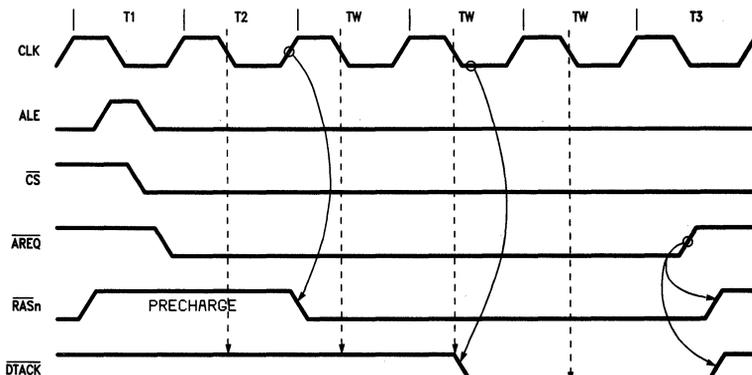
$1\frac{1}{2}T$ during delayed and non-delayed accesses: In Mode 0, \overline{DTACK} will assert from the negative level after the first positive edge of CLK after the positive edge of CLK which starts \overline{RAS} for the access as shown in Figure 29a. In Mode 1, \overline{DTACK} will assert from the negative level after the first positive edge of CLK after \overline{ADS} and \overline{CS} are asserted as shown

in Figures 29c and 29d. During delayed accesses in both modes, \overline{DTACK} will assert from the negative level after the first positive edge of CLK after the positive edge of CLK which starts \overline{RAS} for the access as shown in Figures 29b and 29e.



TL/F/9338-65

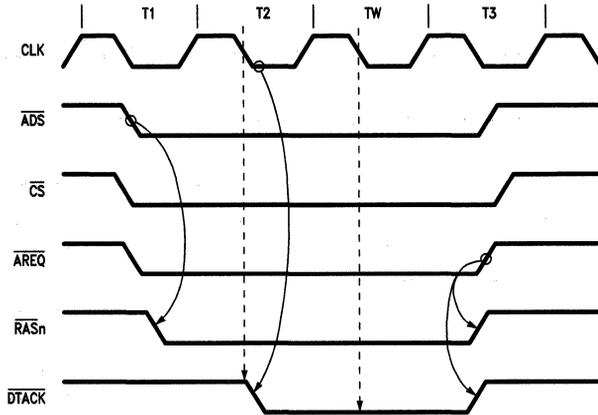
FIGURE 29a. Mode 0 Non-Delayed Access with \overline{DTACK} of $1\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)



TL/F/9338-66

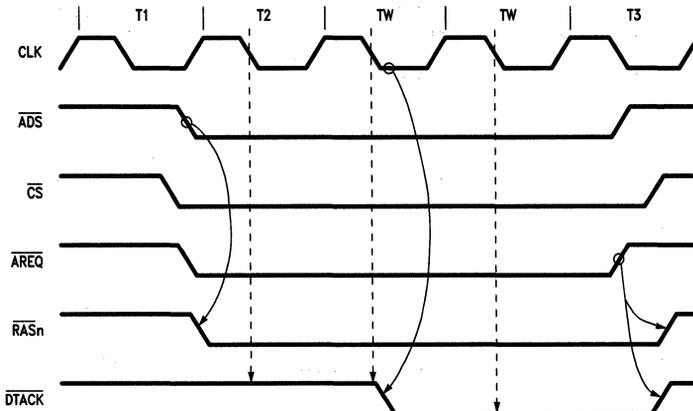
FIGURE 29b. Mode 0 Delayed Access with \overline{DTACK} of $1\frac{1}{2}T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



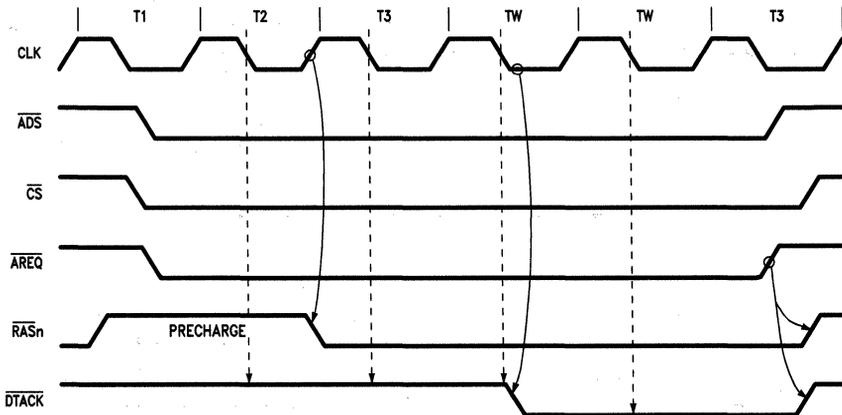
TL/F/9338-67

FIGURE 29c. Mode 1 Non-Delayed Access with \overline{DTACK} of $1/2T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)



TL/F/9338-68

FIGURE 29d. Mode 1 Non-Delayed Access with \overline{DTACK} of $1/2T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)



TL/F/9338-69

FIGURE 29e. Mode 1 Delayed Access with \overline{DTACK} of $1/2T$ (\overline{DTACK} is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)

When starting \overline{DTACK} from a negative level of CLK; if \overline{RAS} is asserted while CLK is high then \overline{DTACK} will assert from the negative edge of CLK, if \overline{RAS} is asserted while CLK is low, then \overline{DTACK} will assert from \overline{RAS} asserting. When starting \overline{DTACK} from a positive edge of CLK in Mode 0, the positive edge of CLK that starts \overline{RAS} can be thought of as 0T. In Mode 1 during non-delayed accesses, the positive edge of CLK that \overline{ADS} is setup to can be thought of as 1T. During delayed accesses, the positive edge of CLK that starts \overline{RAS} can be thought of as 0T and the next positive edge of CLK as 1T.

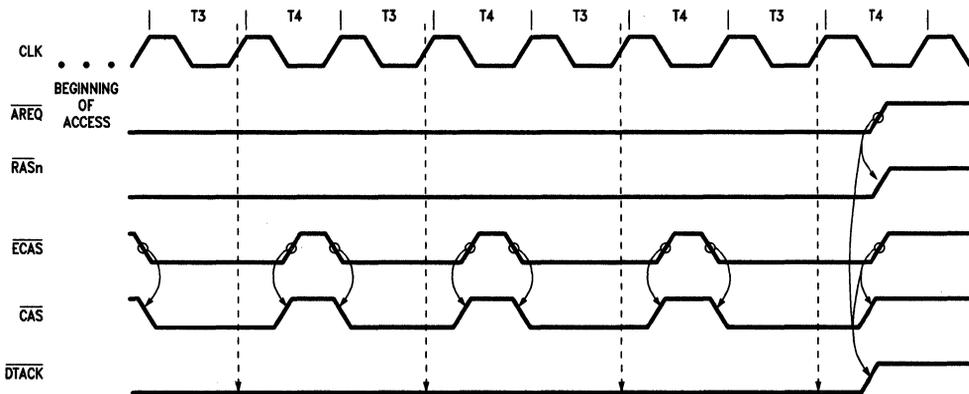
5.2.2 \overline{DTACK} during Page/Burst Accesses

\overline{DTACK} can be programmed to function differently during page/burst types of accesses. During a page/burst access,

the \overline{ECAS} inputs will be asserted then negated while \overline{AREQ} remains asserted. Through address bits R4 and R5, \overline{DTACK} can be programmed to negate and assert during this type of access. The user is given four programming options described below.

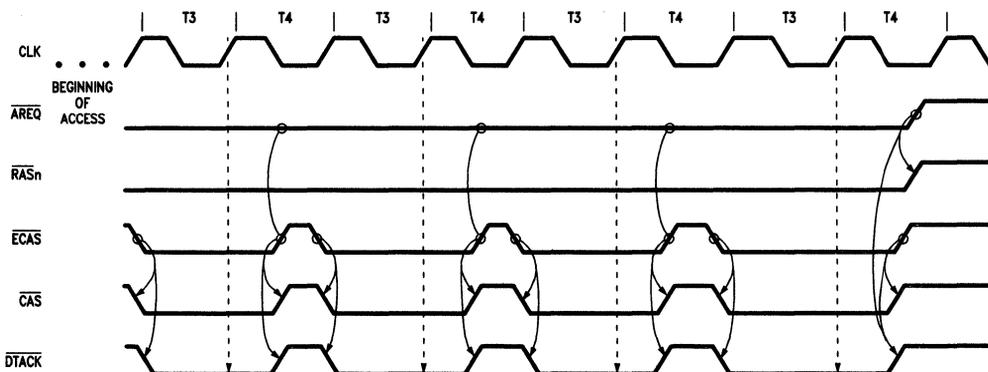
No Wait States: In this case, \overline{DTACK} will remain asserted even if the \overline{ECAS} inputs are negated with \overline{AREQ} asserted as shown in Figure 30.

0T: \overline{DTACK} will be negated when the \overline{ECAS} inputs are negated with \overline{AREQ} asserted. When one or both \overline{ECAS} inputs are asserted again, \overline{DTACK} will be asserted as shown in Figure 31.



TL/F/9338-70

FIGURE 30. No Wait States during Burst Access (\overline{DTACK} is Sampled at the End of the "T3" Clock State)



TL/F/9338-71

FIGURE 31. 0T during Burst Access (\overline{DTACK} is Sampled at the End of the "T3" Clock State)

5.0 Port A Wait State Support (Continued)

$\frac{1}{2}T$: \overline{DTACK} will be negated when the \overline{ECAS} inputs are negated with \overline{AREQ} asserted. When one or both \overline{ECAS} inputs are asserted again, \overline{DTACK} will be asserted from the first negative level of \overline{CLK} after \overline{ECAS} is asserted as shown in Figure 32.

$1T$: \overline{DTACK} will be negated when the \overline{ECAS} inputs are negated with \overline{AREQ} asserted. When one or both \overline{ECAS} inputs are asserted again, \overline{DTACK} will be asserted from the first positive edge of \overline{CLK} after \overline{ECAS} is asserted as shown in Figure 33.

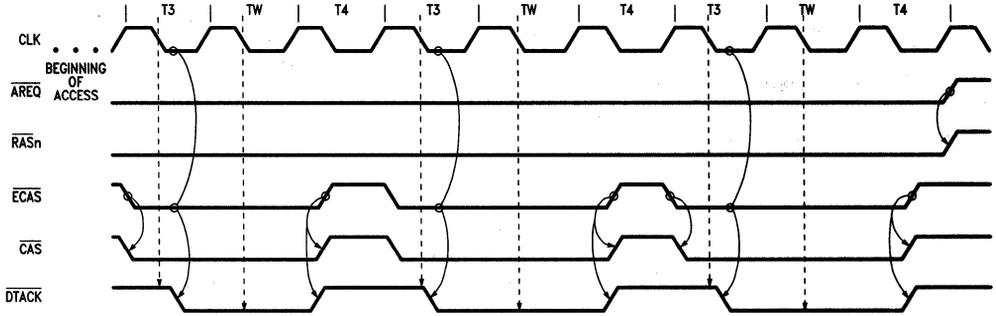


FIGURE 32. $\frac{1}{2}T$ during Burst Access (\overline{DTACK} is Sampled at the "T3" Falling Clock Edge)

TL/F/9338-72

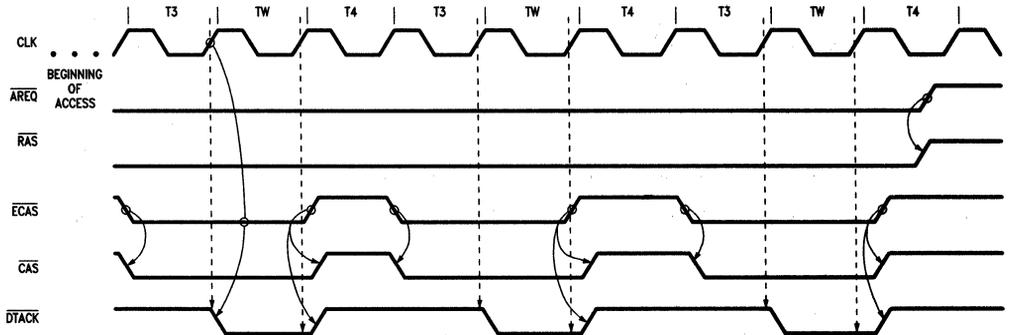


FIGURE 33. $1T$ during Burst Access (\overline{DTACK} is Sampled at the "T3" Falling Clock Edge)

TL/F/9338-73

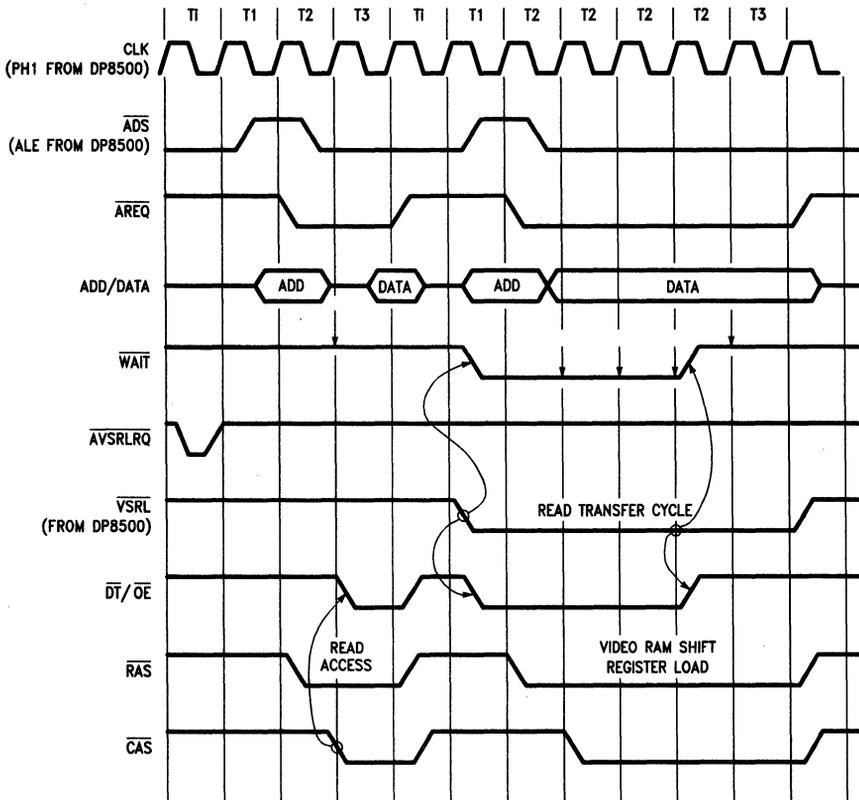
5.0 Port A Wait State Support (Continued)

When starting \overline{DTACK} from a negative level of CLK; if the \overline{ECAS} s are asserted while CLK is high then \overline{DTACK} will assert from the negative edge of CLK, if the \overline{ECAS} s are asserted while CLK is low then \overline{DTACK} will assert from the \overline{ECAS} s asserting. When starting \overline{DTACK} from a positive edge of CLK, the positive edge of CLK that \overline{ECAS} is setup to can be thought of as 1T.

5.3 WAIT STATE SUPPORT FOR VIDEO RAM SHIFT REGISTER LOAD OPERATIONS FOR PORT A

If using the DP8520A/21A/22A in a system using video VRAMs, the CPU that controls loading the Video RAM shift register must be connected to Port A. The input $\overline{AVSRLRQ}$ asserts, signaling an advanced request for a Video RAM

shift register load operation. Sometime later, the input \overline{VSRL} asserts, signifying that the transfer cycle has started, and this action causes the $\overline{DT/OE}$ output to transfer low. \overline{VSRL} asserting also asserts \overline{WAIT} (keeps \overline{DTACK} negated) and will then insert wait states into the transfer cycle. The transfer cycle is completed from either \overline{VSRL} negating or four clocks from \overline{VSRL} asserting. The first event of these two to take place causes \overline{WAIT} to negate (\overline{DTACK} to assert) immediately or one half system clock period later, depending on how the user had programmed \overline{WAIT} to end (\overline{DTACK} to start) during a non-burst type of access. The wait logic is intimately connected to the DP8520A/21A/22A graphics functions and the \overline{WAIT} output functions the same as the $\overline{DT/OE}$ output (see Figure 34).



TL/F/9398-74

FIGURE 34. Wait State Timing during a VRAM Transfer Cycle (\overline{WAIT} Programmed as OT, \overline{WAIT} Sampled at the "T3" Rising Clock Edge)

5.0 Port A Wait State Support (Continued)

5.4 DYNAMICALLY INCREASING THE NUMBER OF WAIT STATES

The user can increase the number of positive edges of CLK before DTACK is asserted or WAIT is negated. With the input WAITIN asserted, the user can delay DTACK asserting or WAIT negating either one or two more positive edges of CLK. The number of edges is programmed through address bit R6. If the user is increasing the number of positive edges in a delay that contains a negative level, the positive edges will be met before the negative level. For example if the user programmed DTACK of $\frac{1}{2}T$, asserting WAITIN, programmed as $2T$, would increase the number of positive edges resulting in DTACK of $2\frac{1}{2}T$ as shown in Figure 35a. Similarly, WAITIN can increase the number of positive edges in a page/burst access. WAITIN can be permanently asserted in systems requiring an increased number of wait states. WAITIN can also be asserted and negated, depending on the type of access. As an example, a user could invert the WRITE line from the CPU and connect the output to WAITIN. This could be used to perform write accesses with 1 wait state and read accesses with 2 wait states as shown in Figure 35b.

5.5 GUARANTEEING RAS LOW TIME AND RAS PRECHARGE TIME

The DP8520A/21A/22A will guarantee \overline{RAS} precharge time between accesses; between refreshes; and between access and refreshes. The programming bits R0 and R1 are used to program combinations of RAS precharge time and RAS low time referenced by positive edges of CLK. RAS low time is programmed for refreshes only. During an access, the system designer guarantees the time RAS is asserted through the DP8520A/21A/22A wait logic. Since inserting wait states into an access increases the length of the CPU signals which are used to create \overline{ADS} or ALE and \overline{AREQ} , the time that \overline{RAS} is asserted can be guaranteed.

Precharge time is also guaranteed by the DP8520A/21A/22A. Each RAS output has a separate positive edge of CLK counter. \overline{AREQ} is negated setup to a positive edge of CLK to terminate the access. That positive edge is $1T$. The next positive edge is $2T$. RAS will not be asserted until the programmed number of positive edges of CLK have passed as shown in Figure 36. Once the programmed precharge time has been met, RAS will be asserted from the positive edge of CLK. However, since there is a precharge counter per RAS, an access using another RAS will not be delayed. Precharge time before a refresh is always referenced from the access RAS negating before RAS0 for the refresh asserting. After a refresh, precharge time is referenced from RAS3 negating, for the refresh, to the access RAS asserting.

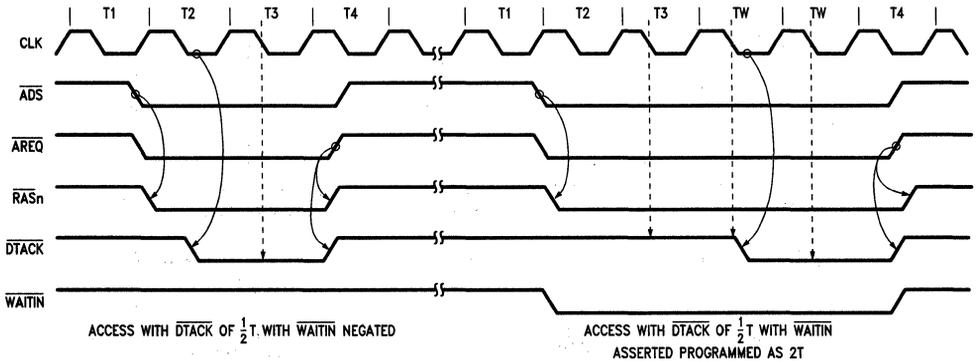
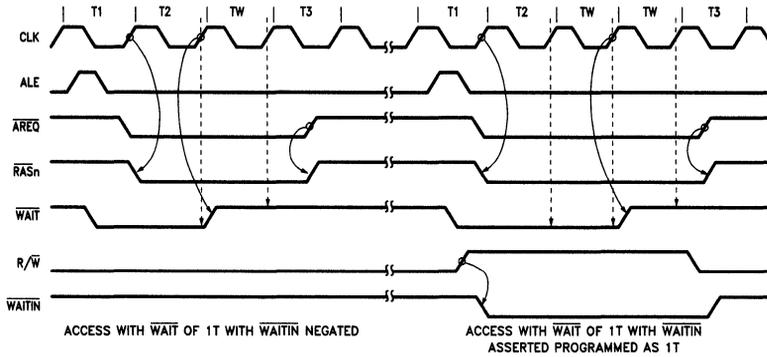


FIGURE 35a. WAITIN Example (DTACK is Sampled at the "T3" Falling Clock Edge)

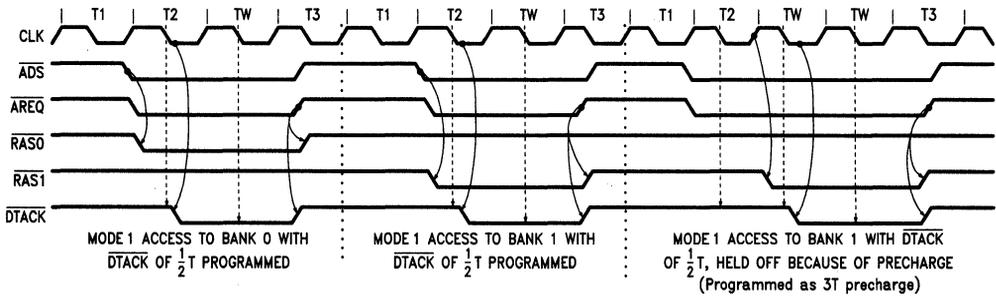
TL/F/9338-75

5.0 Port A Wait State Support (Continued)



TL/F/9338-76

FIGURE 35b. $\overline{\text{WAITn}}$ Example ($\overline{\text{WAIT}}$ is Sampled at the End of "T2")

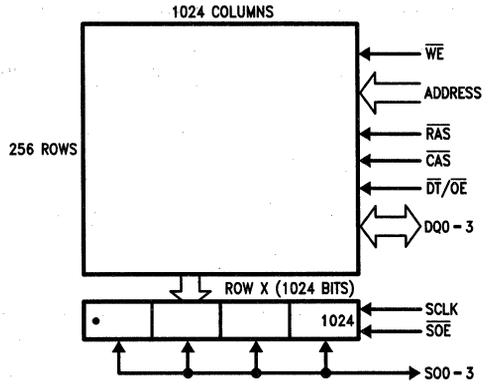


TL/F/9338-77

FIGURE 36. Guaranteeing RAS Precharge ($\overline{\text{DTACK}}$ is Sampled at the "T2" Falling Clock Edge)

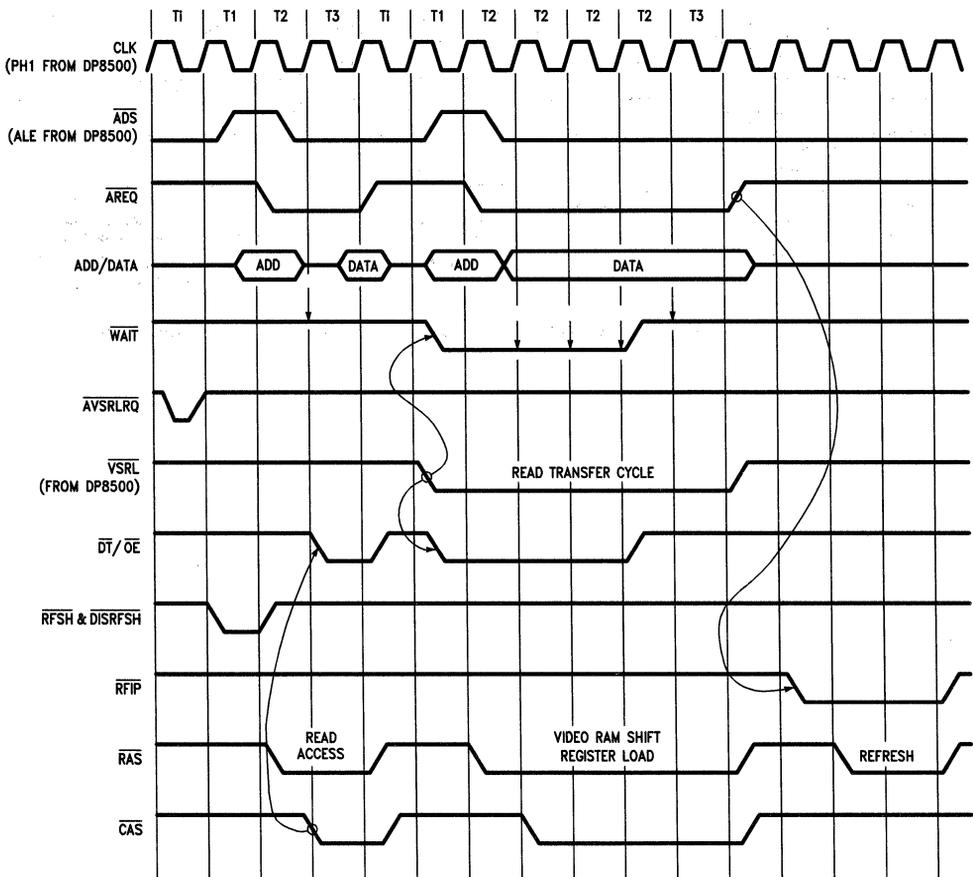
6.0 DP8520A/21A/22A Video RAM Support

The DP8520A/21A/22A provides full support for all access modes of video RAMs through the addition of three pins (AVSRLRQ, VSRL, and DT/OE) to the standard DP8420A/21A/22A. The access modes of video RAMs can be split up into two groups; video RAM transfer cycles (read with the serial port in active or in standby mode, write, and pseudo write transfer cycles), and non-transfer cycles. The DP8520A/21A/22A support of video RAMs allows the full capabilities of the National Semiconductor Advanced Graphics chip set (DP8500 Series) to be realized. See *Figures 37, 38, and 58a*.



TL/F/9338-52

FIGURE 37. The Video RAM (A Dual Ported Memory)
 Ideal solution for graphics frame buffer. Screen refresh can occur at the same time as random access to the frame buffer for screen update and manipulation.



TL/F/9338-56

FIGURE 38. The DP8500 Raster Graphics Processor Interfaced to the DP8520A/21A/22A Video RAM Controller

6.0 DP8520A/21A/22A Video RAM Support (Continued)

6.1 SUPPORT FOR VRAM TRANSFER CYCLES (TO THE SERIAL PORT OF THE VRAM)

The DP8520A/21A/22A supports VRAM transfer cycles with the serial port in the active or standby mode. Active or standby refers to whether data is or is not currently being shifted in or out of the VRAM serial port (i.e., whether the shift clock (SCLK) is currently active). The DP8520A/21A/22A support for data transfer cycles with the serial port in the active mode includes the ability to support transfer cycles with the serial port in the standby mode. Hereafter, the term VRAM transfer cycle means VRAM transfer cycle with the serial port in the active mode.

In order to support VRAM transfer cycles, the DP8520A/21A/22A must be able to guarantee timing with respect to its input CLK (which must be synchronous to VRAM shift clock), \overline{RAS} , \overline{CAS} , and $\overline{DT}/\overline{OE}$. *Figure 38* shows the timing of a graphics memory system where the DP8520A/21A/22A is being used with the National Semiconductor DP8500 Raster Graphics Processor (RGP). If the DP8520A/21A/22A is being used in a graphics frame buffer application, it has the ability to support a VRAM transfer cycle during active video time (ex. mid scan line). This is one of the very attractive features supported by the National Semiconductor Advanced Graphics chip set. Most of the commercial graphics controller chip sets available will only support VRAM transfer cycles during blanking periods (while the VRAM is in standby mode).

The DP8520A/21A/22A supports VRAM transfer cycles during active video time by being able to guarantee an exact instant during which the transfer of VRAM data to the VRAM shift register will occur. This exact instant can be guaranteed through the $\overline{AVSRLRQ}$ and \overline{VSRL} inputs.

The input $\overline{AVSRLRQ}$ disables any further internally or externally requested refreshes or Port B access requests from being executed. The $\overline{AVSRLRQ}$ input does this by making the VRAM controller arbitration logic think that a Port A access is in progress from the point where the $\overline{AVSRLRQ}$ input asserts until the VRAM shift register load operation is

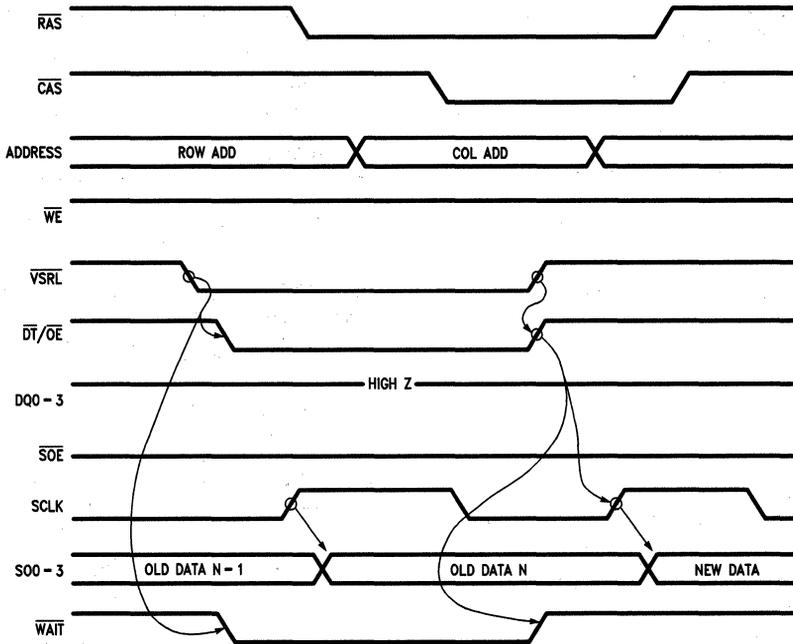
completed. *Figure 38* shows the case of an externally requested refresh being disabled, because of a previous $\overline{AVSRLRQ}$, until the VRAM shift register load has been completed.

The \overline{VSRL} input causes the $\overline{DT}/\overline{OE}$ output to assert immediately, regardless of what else may be happening in the DP8520A/21A/22A. Therefore, it is the system designer's responsibility to guarantee that all pending accesses have been completed by the time the \overline{VSRL} input asserts. The system designer can guarantee this by issuing $\overline{AVSRLRQ}$ far enough in advance to guarantee that all pending accesses have been completed by the time \overline{VSRL} asserts.

The $\overline{AVSRLRQ}$ input does not override the \overline{LOCK} input (see Section 12.0) for dual port systems, and as a result, the designer must also guarantee that Port A can be accessed by assuring that \overline{GRANTB} and \overline{LOCK} are both not asserted when $\overline{AVSRLRQ}$ is asserted.

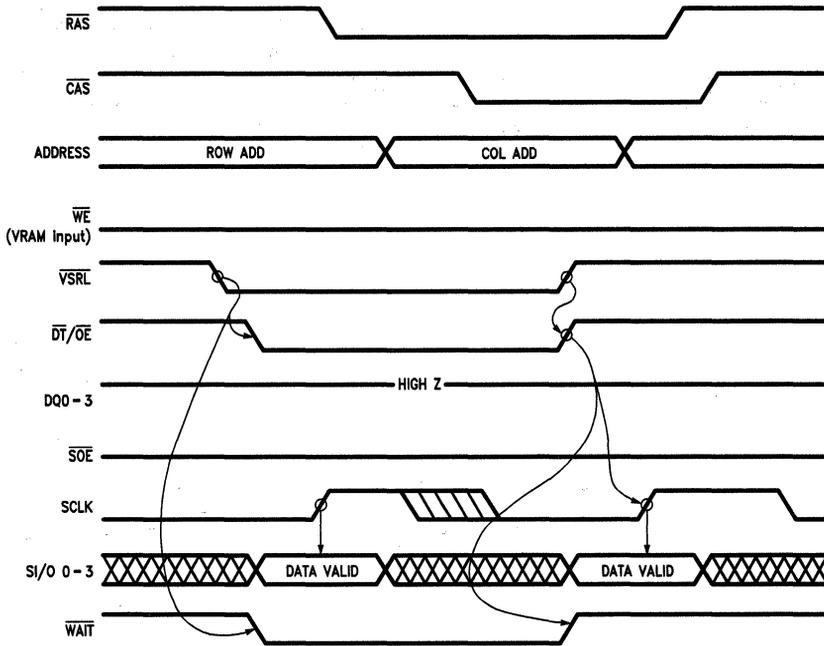
Generally, the \overline{VSRL} is the status of the upcoming access cycle (of the graphics processor). Therefore, this input precedes the inputs \overline{ADS} and \overline{AREQ} that execute the VRAM shift register load transfer cycle. This sequence of events guarantees the correct relationship of $\overline{DT}/\overline{OE}$, \overline{RAS} and \overline{CAS} (\overline{DT} preceding \overline{RAS} and \overline{CAS} when asserting and negating). The wait logic is also intimately connected to the graphics functions on the DP8520A/21A/22A. The $\overline{DT}/\overline{OE}$ (and $\overline{WAIT}/\overline{DTACK}$) relationship to \overline{VSRL} during a VRAM transfer cycle depends upon how the DP8520A/21A/22A was programmed with respect to the $\overline{ECAS0}$ input. If $\overline{ECAS0}$ was negated during programming, the $\overline{DT}/\overline{OE}$ output will follow \overline{VSRL} asserting. $\overline{DT}/\overline{OE}$ will then negate either when \overline{VSRL} negates or from the fourth rising clock edge after \overline{VSRL} asserted, whichever event takes place first. This allows \overline{DT} to negate before \overline{RAS} and \overline{CAS} negate, thus guaranteeing the correct timing relationship during the transfer cycle (see *Figure 38*). The \overline{WE} input of the VRAM determines whether the access is a read or write transfer cycle (see *Figures 39* and *40* respectively).

6.0 DP8520A/21A/22A Video RAM Support (Continued)



TL/F/9338-78

**FIGURE 39. Video RAM Timing READ Transfer Cycle, B Port Active
(Transfer VRAM Row Data into Shift Register)**



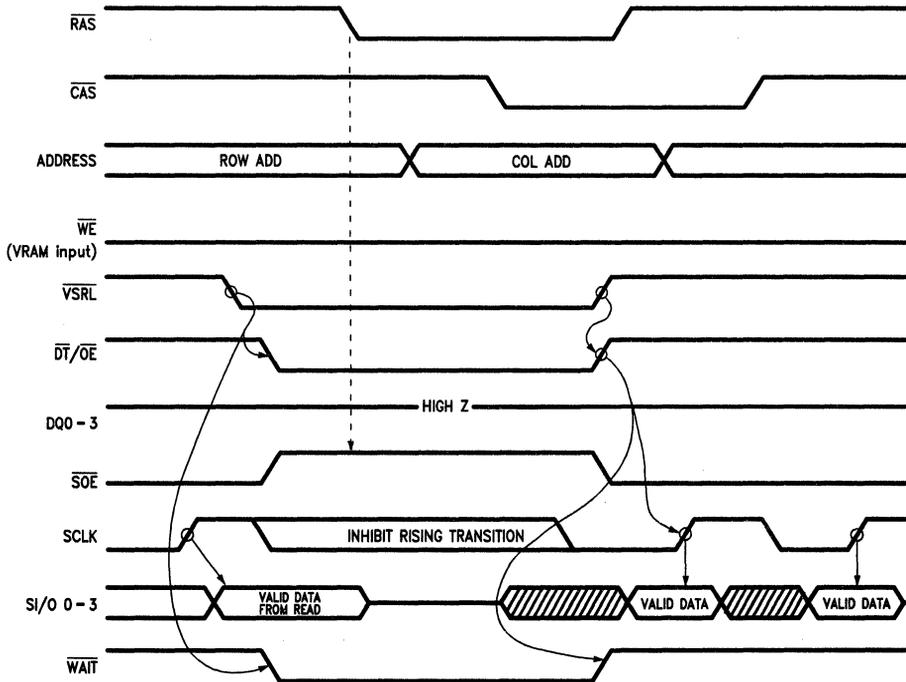
TL/F/9338-55

**FIGURE 40. Video RAM Timing WRITE Transfer Cycle, B Port Active
(Transfer Shift Register Data into VRAM Row)**

6.0 DP8520A/21A/22A Video RAM Support (Continued)

During a transfer cycle (\overline{VSRL} asserted during the access) \overline{WIN} is disabled from affecting the $\overline{DT}/\overline{OE}$ logic until the transfer cycle is completed as shown by \overline{CAS} negating. During a transfer cycle, the \overline{SOE} (Serial Output Enable) input to the VRAM is asserted and is used as an output control for a read transfer cycle and is used as a write enable control during a write transfer cycle. When \overline{SOE} is negated, serial access is disabled, and a transfer cycle cannot take place. \overline{SOE} asserted during a read enables the serial input/output bus SI/O (0-3) while the VRAM data bus (DQ0-3) is put into a high impedance state, thus allowing the transfer cycle

to take place from the serial port. In addition to both read and write transfer cycles, the DP8520A/21A/22A also supports pseudo write transfer cycles (see *Figure 41*). A pseudo write transfer cycle must be performed after a read transfer cycle if the subsequent operation is a write transfer cycle. The DP8520A/21A/22A VRAM controller is operated as if it is doing a write transfer cycle, but since the \overline{SOE} input to the VRAM is negated (disabling the serial port), a transfer doesn't take place. The purpose of this pseudo write transfer cycle is to switch the SI/O (0-3) lines of the VRAM's serial port from output mode to input mode.



TL/F/9338-79

FIGURE 41. Video RAM Timing Pseudo WRITE Transfer Cycle, B Port Active (Transfer Shift Register Data into VRAM Row)

6.0 DP8520A/21A/22A Video RAM Support (Continued)

6.2 SUPPORT FOR VRAM ACCESS CYCLES THROUGH PORT A USING THE DP8520A/21A/22A

With the DP8520A/21A/22A, the output $\overline{DT}/\overline{OE}$ will remain negated during write accesses (see *Figure 42*), but during read accesses it will assert after \overline{CASn} asserts. $\overline{DT}/\overline{OE}$ will be negated for a read access once \overline{CASn} is negated (see *Figure 43*), causing the VRAM outputs to be enabled. If \overline{CASn} toggles during a page mode read access, then the $\overline{DT}/\overline{OE}$ logic will also toggle following the \overline{CASn} input.

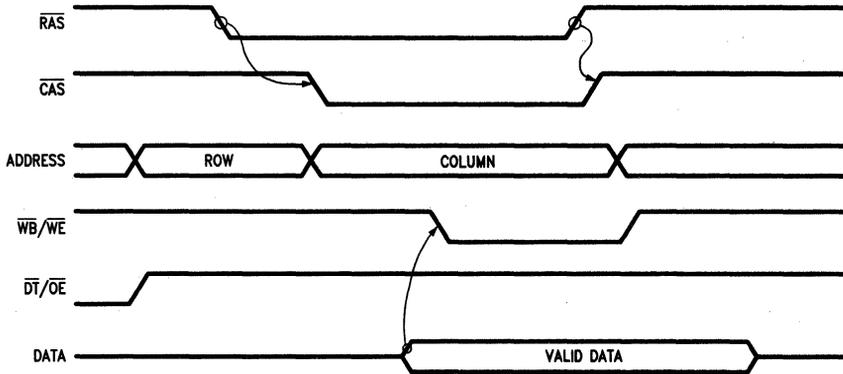


FIGURE 42. Video RAM Random Access Write Cycle

TL/F/9338-54

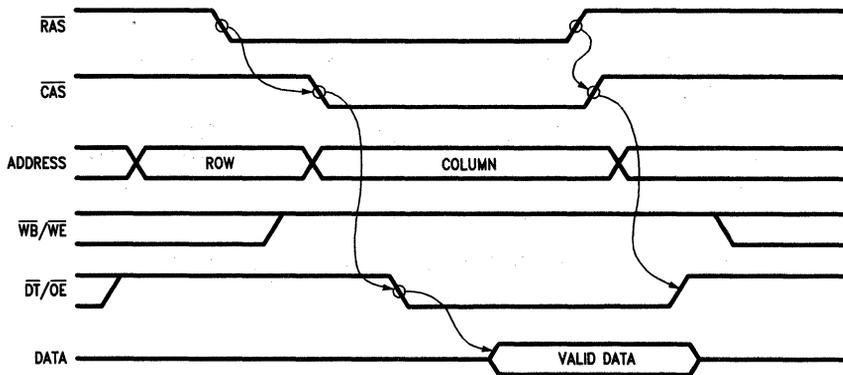


FIGURE 43. Video RAM Random Access Read Cycle

TL/F/9338-53

7.0 Additional Access Support Features

To support the different modes of accessing, the DP8520A/21A/22A have multiple access features. These features allow the user to take advantage of CPU or VRAM functions. These additional features include: address latches and column increment for page/burst mode support; address pipelining to allow a new access to start to a different bank of VRAM after \overline{CAS} has been asserted and the column address hold time has been met; and delay \overline{CAS} , to allow the user with a multiplexed bus to ensure valid data is present before \overline{CAS} is asserted.

7.1 ADDRESS LATCHES AND COLUMN INCREMENT

The address latches can be programmed, through programming bit B0, to either latch the address or remain permanently in fall-through mode. If the address latches are used to latch the address, the rising edge of ALE in Mode 0 places the latches in fall-through. Once ALE is negated, the address present on the row, column and bank inputs is latched. In Mode 1, the address latches are in fall-through mode until \overline{ADS} is asserted. \overline{ADS} asserted latches the address.

Once the address is latched, the column address can be incremented with the input COLINC. With COLINC asserted, the column address is incremented. If COLINC is asserted with all of the bits of the column address asserted, the column address will return to zero. COLINC can be used for sequential accesses of static column VRAMs. COLINC can also be used with the \overline{ECAS} inputs to support sequential accesses to page mode VRAMs as shown in *Figure 44*. COLINC should only be asserted when a refresh is not in progress as indicated by \overline{RFIP} , if programmed, being negated during an access since this input functions as an extend refresh when a refresh is in progress.

The address latches function differently with the DP8522A. The DP8522A will latch the address of the currently granted port. If Port A is currently granted, the address will be latched as described in Section 7.1. If Port A is not granted, and requests an access, the address will be latched on the first or second positive edge of CLK after GRANTB has been negated depending on the programming bits R0, R1.

For Port B, if GRANTB is asserted, the address will be latched with \overline{AREQB} asserted. If GRANTB is negated, the address will latch on the first or second positive edge of CLK after GRANTB is asserted depending on the programming bits R0, R1.

7.2 ADDRESS PIPELINING

Address pipelining is the overlapping of accesses to different banks of VRAM. If the majority of successive accesses are to a different bank, the accesses can be overlapped. Because of this overlapping, the cycle time of the VRAM accesses are greatly reduced. The DP8520A/21A/22A can be programmed to allow a new row address to be placed on the VRAM address bus after the column address hold time has been met. At this time, a new access can be initiated with \overline{ADS} or ALE, depending on the access mode, while \overline{AREQ} is used to sustain the current access. The DP8522A supports address pipelining for Port A only. This mode can not be used with page, static column or nibble modes of operations because the VRAM column address is switched back to the row address after \overline{CAS} is asserted. This mode is programmed through address bit R8 (see *Figures 45a* and *45b*).

During address pipelining in Mode 0, shown in *Figure 45c*, ALE cannot be pulsed high to start another access until \overline{AREQ} has been asserted for the previous access for at least one period of CLK. \overline{DTACK} , if programmed, will be negated once \overline{AREQ} is negated. \overline{WAIT} , if programmed to insert wait states, will be asserted once ALE and \overline{CS} are asserted.

In Mode 1, shown in *Figure 45d*, \overline{ADS} can be negated once \overline{AREQ} is asserted. After meeting the minimum negated pulse width for \overline{ADS} , \overline{ADS} can again be asserted to start a new access. \overline{DTACK} , if programmed, will be negated once \overline{AREQ} is negated. \overline{WAIT} , if programmed, will be asserted once \overline{ADS} is asserted.

In either mode with either type of wait programmed, the DP8520A/21A/22A will still delay the access for precharge if sequential accesses are to the same bank or if a refresh takes place.

7.0 Additional Access Support Features (Continued)

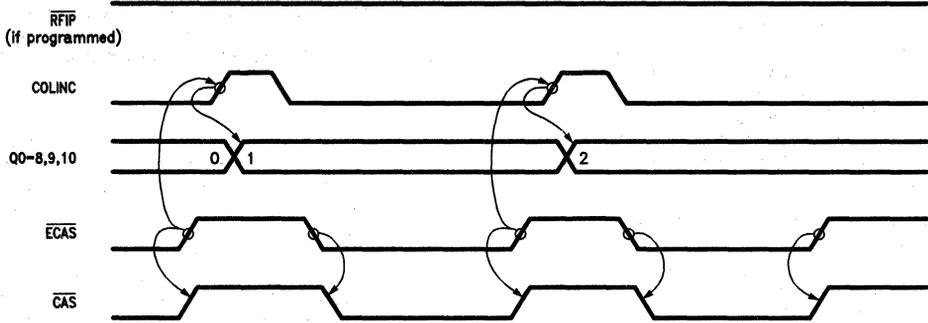


FIGURE 44. Column Increment

TL/F/9338-80

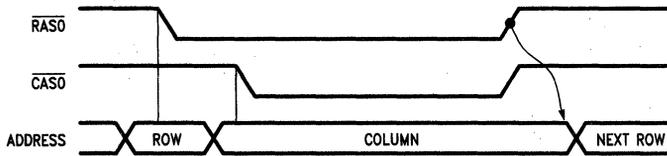


FIGURE 45a. Non-Address Pipelined Mode

TL/F/9338-81

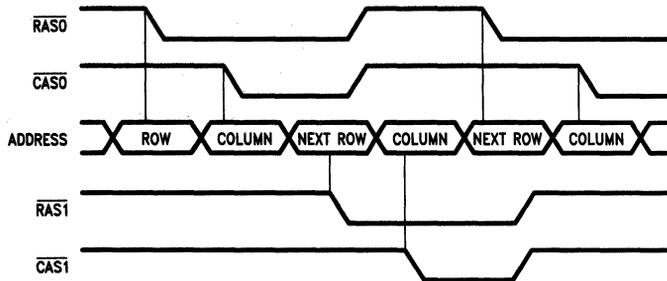


FIGURE 45b. Address Pipelined Mode

TL/F/9338-82

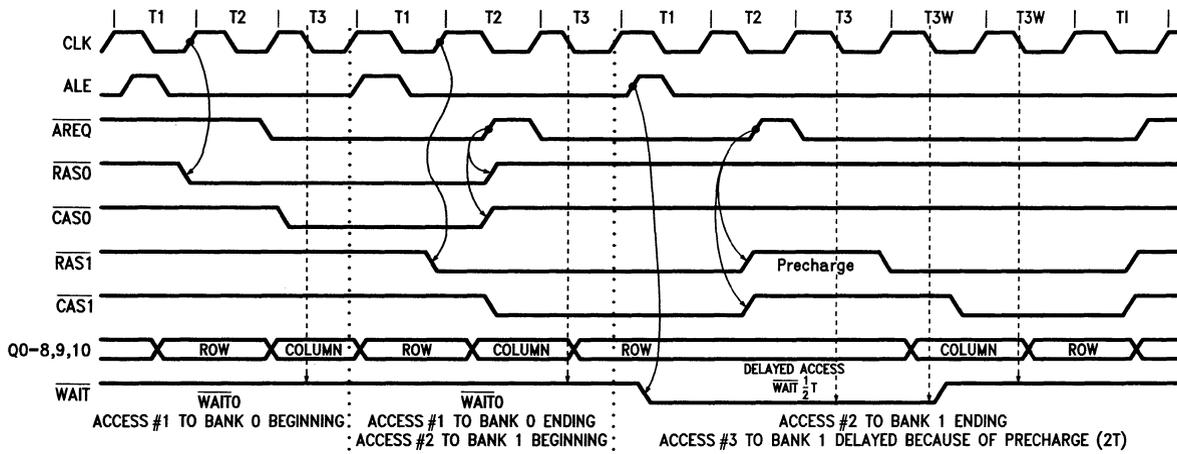


FIGURE 45c. Mode 0 Address Pipelining (WAIT of 0, 1/2T has Been Programmed. WAIT is Sampled at the "T3" Falling Clock Edge)

TL/F/9338-83

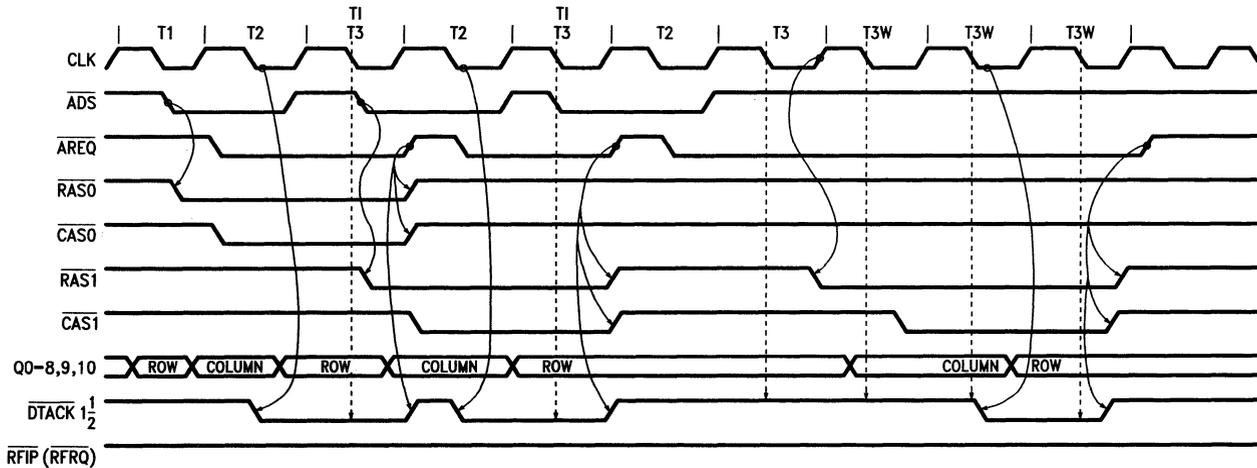


FIGURE 45d. Mode 1 Address Pipelining (DTACK 1/2T Programmed, DTACK is Sampled at the "T3" Falling Clock Edge)

TL/F/9338-84

7.0 Additional Access Support Features

7.3 DELAY $\overline{\text{CAS}}$ DURING WRITE ACCESSES

Address bit C9 asserted during programming will cause $\overline{\text{CAS}}$ to be delayed until the first positive edge of CLK after $\overline{\text{RAS}}$ is asserted when the input $\overline{\text{WIN}}$ is asserted. Delaying $\overline{\text{CAS}}$ during write accesses ensures that the data to be written to

VRAM will be setup to $\overline{\text{CAS}}$ asserting as shown in *Figures 46a and 46b*. If the possibility exists that data still may not be present after the first positive edge of CLK, $\overline{\text{CAS}}$ can be delayed further with the ECAS inputs. If address bit C9 is negated during programming, read and write accesses will be treated the same (with regard to $\overline{\text{CAS}}$).

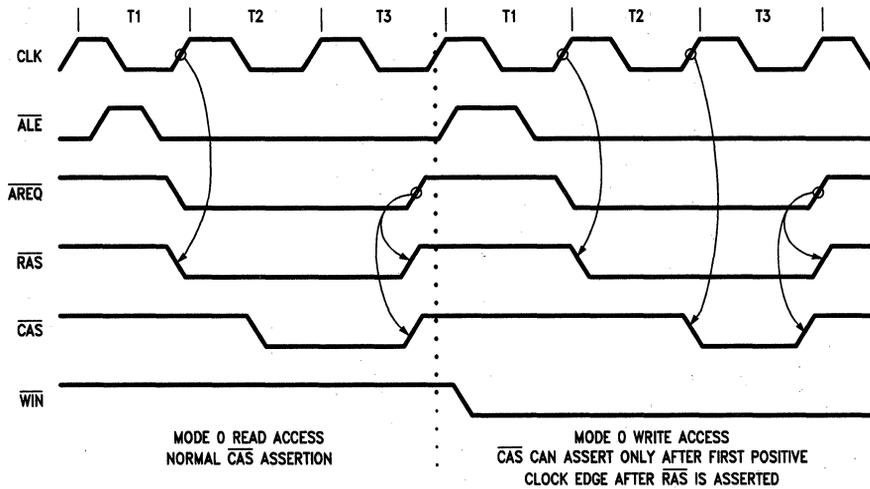


FIGURE 46a. Mode 0 Delay $\overline{\text{CAS}}$

TL/F/9338-85

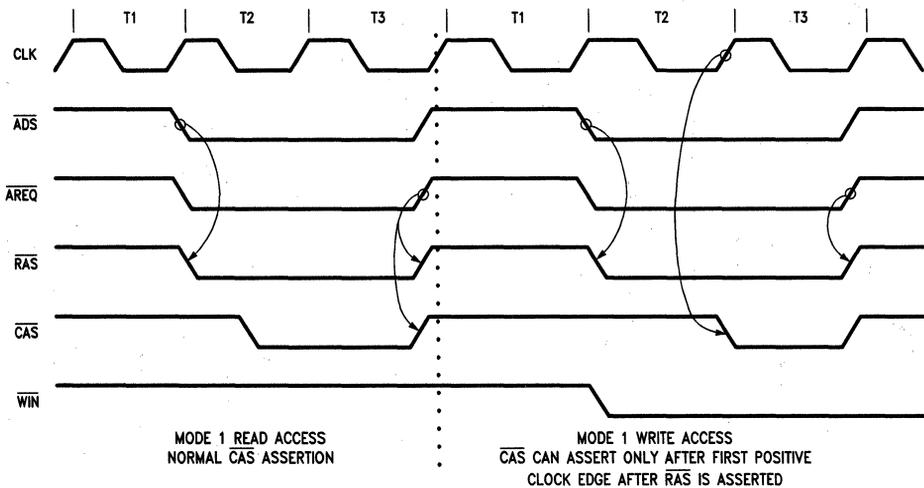


FIGURE 46b. Mode 1 Delay $\overline{\text{CAS}}$

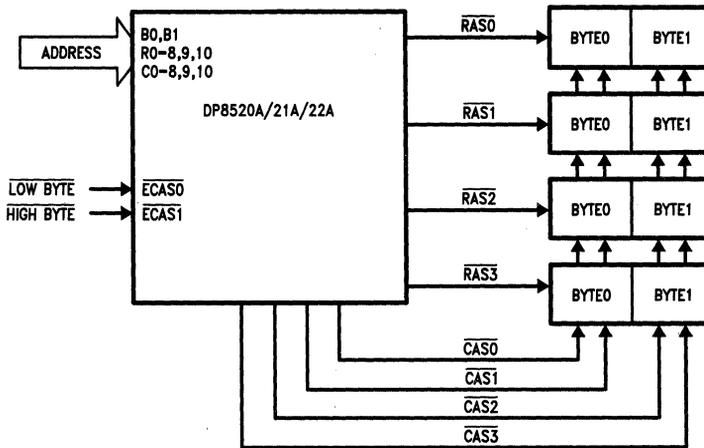
TL/F/9338-86

8.0 RAS and CAS Configuration Modes

The DP8520A/21A/22A allow the user to configure the VRAM array to contain one, two or four banks of VRAM. Depending on the functions used, certain considerations must be used when determining how to set up the VRAM array. Programming address bits C4, C5 and C6 along with bank selects, B0-1, and CAS enables, ECAS0-1, determine which RAS or group of RASs and which CAS or group of CASs will be asserted during an access. Different memory schemes are described. The DP8520A/21A/22A is specified driving a heavy load of 72 VRAMs, representing four banks of VRAM with 16-bit words and 2 parity bits. The DP8520A/21A/22A can drive more than 72 VRAMs, but the AC timing must be increased. Since the RAS and CAS outputs are configurable, all RAS and CAS outputs should be used for the maximum amount of drive.

8.1 BYTE WRITING

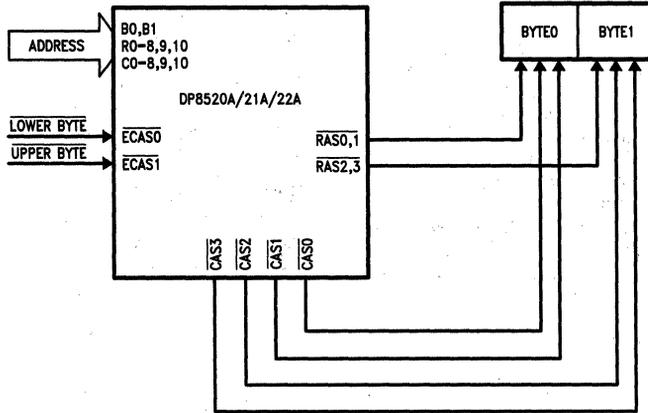
By selecting a configuration in which all CAS outputs are selected during an access, each ECAS input enables a pair of CAS outputs to select a byte in a word size of up to 16 bits. In this case, the RAS outputs are used to select which of up to 4 banks is to be used as shown in Figures 47a and 47b. The user can also configure the VRAM array into an 8 bank system as shown in Figure 47c. This setup can be used along with byte writing for an 8-bit system if the LOW BYTE and HIGH BYTE are connected to ECAS0 and ECAS1 respectively. The user can connect upper address bits to ECAS0,1 for use in an 8 bank-16-bit system, but cannot use byte writing in this case.



TL/F/9338-88

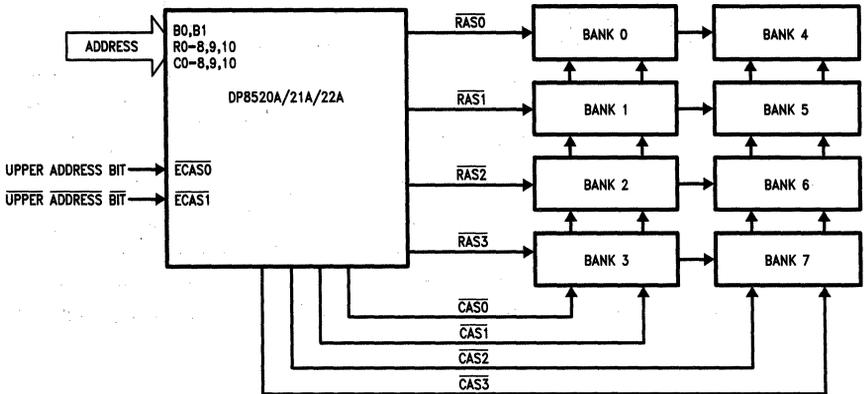
FIGURE 47a. VRAM Array Setup for 16-Bit System (C6, C5, C4 = 1, 1, 0 during Programming)

8.0 RAS and CAS Configuration Modes (Continued)



TL/F/9338-87

FIGURE 47b. VRAM Array Setup for 16-Bit, 1 Bank System (C6, C5, C4 = 0, 0, 0 Allowing Error Scrubbing or C6, C5, C4 = 0, 1, 1 No Error Scrubbing during Programming)



TL/F/9338-89

FIGURE 47c. 8 Bank VRAM Array (C6, C5, C4 = 1, 1, 0 during Programming)

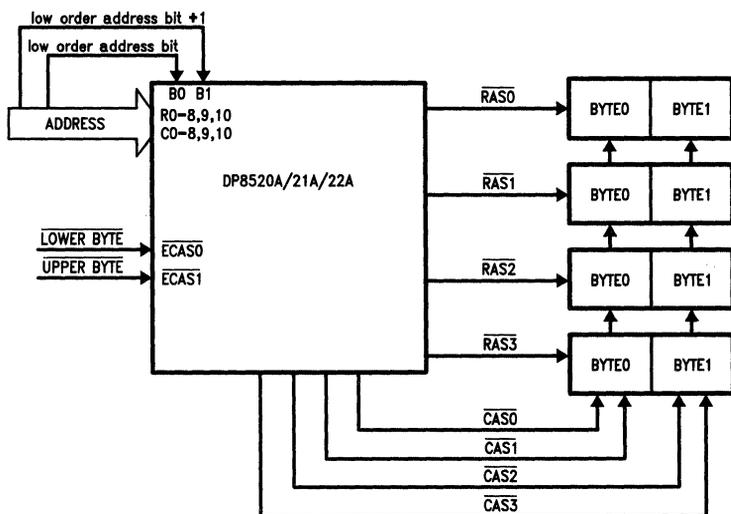
8.0 RAS and CAS Configuration Modes (Continued)

8.2 MEMORY INTERLEAVING

Memory interleaving allows the cycle time of VRAMs to be reduced by having sequential accesses to different memory banks. Since the DP8520A/21A/22A have separate pre-charge counters per bank, sequential accesses will not be delayed if the accessed banks use different RAS outputs. To ensure different RAS outputs will be used, a mode is selected where either one or two RAS outputs will be asserted during an access. The bank select or selects, B0 and B1, are then tied to the least significant address bits, causing a different group of RASs to assert during each sequential access as shown in Figure 48. In this figure there should be at least one clock period of all RASs negated between different RASs being asserted to avoid the condition of a CAS before RAS refresh cycle.

8.3 ADDRESS PIPELINING

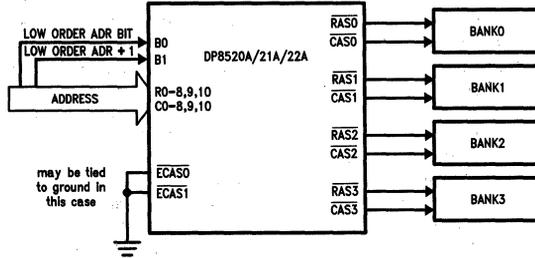
Address pipelining allows several access RASs to be asserted at once. Because RASs can overlap, each bank requires either a mode where one RAS and one CAS are used per bank as shown in Figure 49a or where two RASs and two CASs are used per bank as shown in Figure 49b. In order to perform byte writing while using address pipelining, external gating on the CAS outputs must be used. If the array is not layed out this way, a CAS to a bank can be low before RAS, which will cause a refresh of the VRAM, not an access. To take full advantage of address pipelining, memory interleaving is used. To memory interleave, the least significant address bits should be tied to the bank select inputs to ensure that all "back to back" sequential accesses are not delayed, since different memory banks are accessed.



TL/F/9398-80

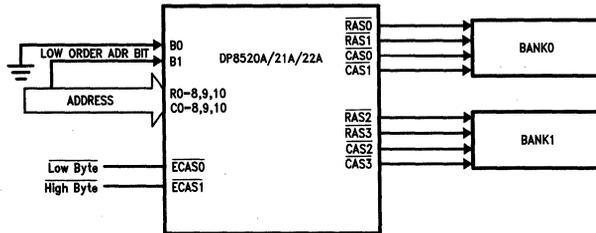
FIGURE 48. Memory Interleaving with Byte Writing Capability (C6, C5, C4 = 1, 1, 0 during Programming)

8.0 RAS and CAS Configuration Modes (Continued)



TL/F/9338-91

FIGURE 49a. VRAM Array Setup for 4 Banks Using Address Pipelining (C6, C5, C4 = 1, 1, 1 or C6, C5, C4 = 0, 1, 0 (Also Allowing Error Scrubbing) during Programming)



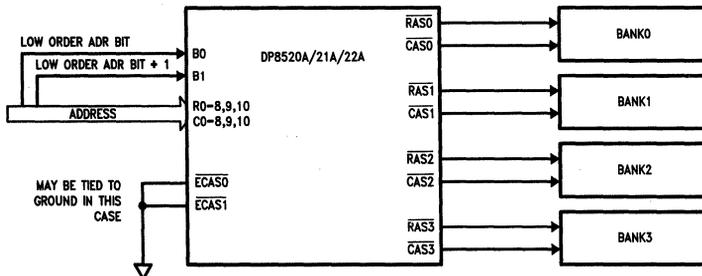
TL/F/9338-92

FIGURE 49b. VRAM Array Setup for Address Pipelining with 2 Banks (C6, C5, C4 = 1, 0, 1 or C6, C5, C4 = 0, 0, 1 (Also Allowing Error Scrubbing) during Programming)

8.4 ERROR SCRUBBING

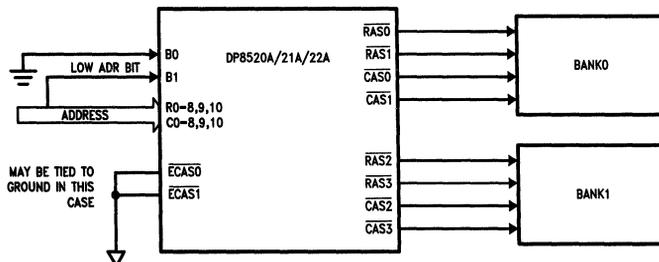
In error scrubbing during refresh, the user selects one, two or four $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ outputs per bank. When performing error detection and correction, memory is always accessed

as words. Since the $\overline{\text{CAS}}$ signals are not used to select individual bytes, their corresponding $\overline{\text{ECAS}}$ inputs can be tied low as shown in *Figures 50a* and *50b*.



TL/F/9338-93

FIGURE 50a. VRAM Array Setup for 4 Banks Using Error Scrubbing (C6, C5, C4 = 0, 1, 0 during Programming)



TL/F/9338-94

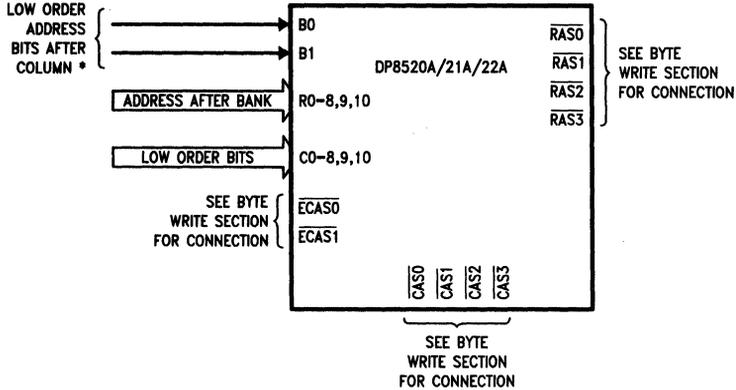
FIGURE 50b. VRAM Array Setup for Error Scrubbing with 2 Banks (C6, C5, C4 = 0, 0, 1 during Programming)

8.0 RAS and CAS Configuration Modes (Continued)

8.5 PAGE/BURST MODE

In a static column, page or burst mode system, the least significant bits must be tied to the column address in order to ensure that the page/burst accesses are to sequential memory addresses, as shown in *Figure 51*. In a nibble mode system, the two least significant address bits (A2, A3) must be tied to the highest row and column address inputs (depends on VRAM size) to ensure that the toggling bits of nibble mode VRAMs are to sequential memory addresses.

The \overline{ECAS} inputs may then be toggled with the DP8520A/21A/22A's address latches in fall-through mode, while \overline{AREQ} is asserted. The \overline{ECAS} inputs can also be used to select individual bytes. When using nibble mode VRAMs, the third and fourth address bits can be tied to the bank select inputs to perform memory interleaving. In page or static column modes, the two address bits after the page size can be tied to the bank select inputs to select a new bank if the page size is exceeded.



TL/F/9338-95

*See table below for row, column & bank address bit map. A0,A1 are used for byte addressing in this example.

Addresses	Nibble Mode*	Page Mode/Static Column Mode Page Size			
		256 Bits/Page	512 Bits/Page	1024 Bits/Page	2048 Bits/Page
Column Address	R8, C9 = A2, A3 C0-8 = X	C0-7 = A2-9 C8-10 = X	C0-8 = A2-10 C9,10 = X	C0-9 = A2-11 C10 = X	C0-10 = A2-12
Row Address	X	X	X	X	X
B0	A4	A10	A11	A12	A13
B1	A5	A11	A12	A13	A14

*Assuming 1 M-bit Vrams are being used.

Assume that the least significant address bits are used for byte addressing. Given a 32-bit system A0,A1 would be used for byte addressing.

X = DON'T CARE, the user can do as he pleases.

FIGURE 51. Page, Static Column, Nibble Mode System

9.0 Programming and Resetting

The DP8520A/21A/22A must be programmed by one of two possible programming sequences before it can be used. At power up, the DP8520A/21A/22A programming bits are in an undefined state. All internal latches and flip-flops are cleared. After programming, the DP8520A/21A/22A enters a 60 ms initialization period. During this initialization period, the DP8520A/21A/22A performs refreshes about every 15 μ s; this makes further VRAM warmup cycles unnecessary. The chip can be programmed as many times as the user wishes. After the first programming, the 60 ms initialization period will not be entered into unless the chip is reset. Refreshes occur during the 60 ms initialization period. If $\overline{ECAS0}$ was asserted during programming, the \overline{RFIP} (\overline{RFRQ}) pin will act as \overline{RFIP} and will be asserted throughout the initialization period, otherwise the pin will act like \overline{RFRQ} and toggle every 13 μ s–15 μ s in conjunction with internal refresh requests. If the user attempts an access during the initialization period, wait states will be inserted into the access cycle until the initialization period is complete and \overline{RAS} precharge time has been met. The actual initialization time period is given by the following formula:

$$T = 4096 * (\text{Clock Divisor Select}) * (\text{Refresh Clock Fine Tune}) / (\text{DELCK Frequency})$$

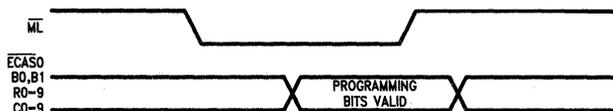
9.1 MODE LOAD ONLY PROGRAMMING

MODE LOAD, \overline{ML} , asserted enables an internal 23-bit programmable register. To use this method, the user asserts \overline{ML} , enabling the internal programming register. After \overline{ML} is asserted, a valid programming selection is placed on the address bus (and $\overline{ECAS0}$), then \overline{ML} is negated. When \overline{ML} is negated, the value on the address bus (and $\overline{ECAS0}$) is latched into the internal programming register and the DP8520A/21A/22A is programmed, as shown in Figure 52a. After \overline{ML} is negated, the DP8520A/21A/22A will enter the 60 ms initialization period only if this is the first programming after power up or reset.

Using this method, a set of transceivers on the address bus can be put at TRI-STATE[®] by the system reset signal. A combination of pull-up and pull-down resistors can be used on the address inputs of the DP8520A/21A/22A to select the programming values, as shown in Figure 52b.

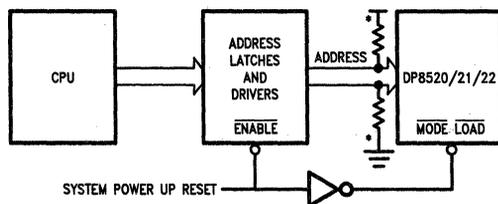
9.2 CHIP SELECTED ACCESS PROGRAMMING

The chip can also be programmed by asserting \overline{ML} and performing a chip selected access. \overline{ADS} (or ALE) is disabled internally until after programming. To program the chip using this method, \overline{ML} is asserted, \overline{CS} is asserted and a valid programming selection is placed on the address bus. When \overline{AREQ} is asserted, the chip is programmed with the programming selection on the address bus. After \overline{AREQ} is negated, \overline{ML} can be negated as shown in Figure 53a.



TL/F/9338-96

FIGURE 52a. Mode Load Only Programming



TL/F/9338-97

*Pull-Up or Pull-Down Resistors on Each Address Input

FIGURE 52b. Programming during System Reset

9.0 Programming and Resetting (Continued)

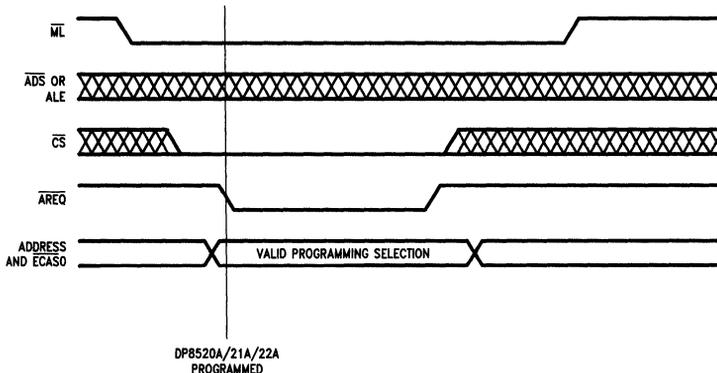


FIGURE 53a. CS Access Programming

TL/F/9338-98

Using this method, various programming schemes can be used. For example if extra upper address bits are available, an unused high order address bit can be tied to the signal \overline{ML} . Using this method, one need only write to a page of memory, thus asserting the high order bit and in turn programming the chip as shown in *Figure 53b*.

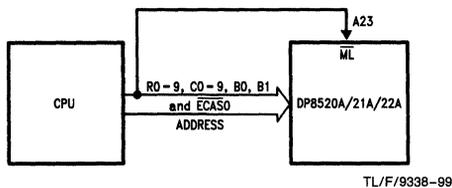


FIGURE 53b. Programming the DP8520A/21A/22A through the Address Bus Only

An I/O port can also be used to assert \overline{ML} . After \overline{ML} is asserted, a chip selected access can be performed to program the chip. After the chip selected access, \overline{ML} can be negated through the I/O port as shown in *Figure 53c*.

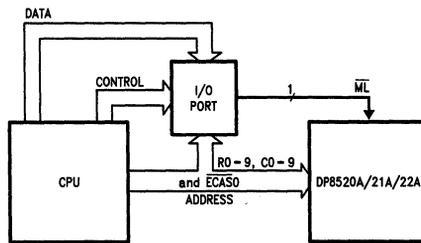


FIGURE 53c. Programming the DP8520A/21A/22A through the Address Bus and an I/O Port

Another simple way the chip can be programmed is the first write after system reset. This method requires only a flip-flop and an OR gate as shown in *Figure 53d*. At reset, the flip-flop is preset, which pulls the \overline{Q} output low. Since \overline{WR} is negated, \overline{ML} is not enabled. The first write access is used to program the chip. When \overline{WR} is asserted, \overline{ML} is asserted. \overline{WR} negated clocks the flip-flop, negates \overline{ML} , and programs the DP8520A/21A/22A with the address and \overline{ECASO} available at that time. \overline{CS} does not need to be asserted using this method.

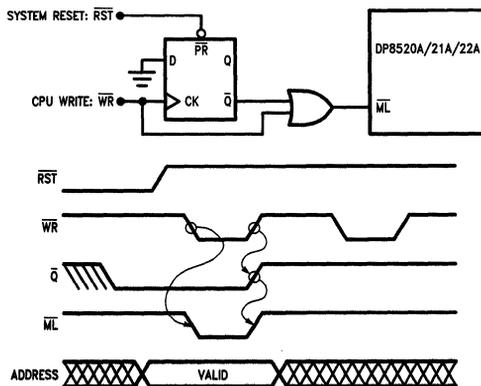


FIGURE 53d. Programming the DP8520A/21A/22A on the First CPU Write after Power Up

9.0 Programming and Resetting (Continued)

9.3 EXTERNAL RESET

At power up, all internal latches and flip-flops are cleared. The power up state can again be entered by asserting \overline{ML} and $\overline{DISRFSH}$ for 16 positive edges of CLK. After resetting if the user negates $\overline{DISRFSH}$ before negating \overline{ML} as shown in

Figure 54a, \overline{ML} negated will program the chip. If \overline{ML} is negated before or at the same time as $\overline{DISRFSH}$ as shown in *Figure 54b*, the chip will not be programmed. After the chip is programmed, the 60 ms initialization period will be entered into if this is the first programming after power up or reset.

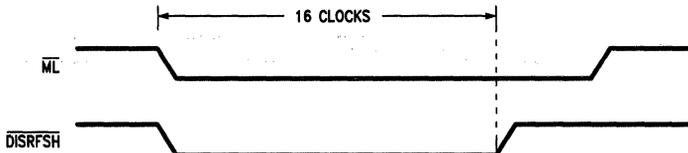


FIGURE 54a. Chip Reset and Programmed

TL/F/9338-A2

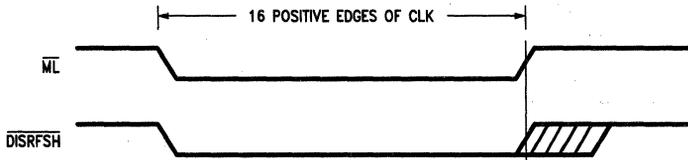


FIGURE 54b. Chip Reset but Not Programmed

TL/F/9338-A3

9.0 Programming and Resetting (Continued)

9.4 PROGRAMMING BIT DEFINITIONS

Symbol	Description
ECAS0	Extend $\overline{\text{CAS}}$/Refresh Request Select
0	The $\overline{\text{CAS}}_n$ outputs will be negated with the $\overline{\text{RAS}}_n$ outputs when $\overline{\text{AREQ}}$ (or $\overline{\text{AREQB}}$, DP8522A only) is negated. The $\overline{\text{RFIP}}$ pin will function as refresh in progress. During a video shift register load operation, the $\overline{\text{DT}}/\overline{\text{OE}}$ output will be negated by either the 4th rising clock edge after the input $\overline{\text{VSRL}}$ asserts, or by the $\overline{\text{VSRL}}$ input negating, whichever occurs first, when this mode is programmed.
1	The $\overline{\text{CAS}}_n$ outputs will be negated, during an access (Port A (or Port B, DP8522A only)) when their corresponding $\overline{\text{ECAS}}_n$ inputs are negated. This feature allows the $\overline{\text{CAS}}$ outputs to be extended beyond the $\overline{\text{RAS}}$ outputs negating. Scrubbing refreshes are NOT affected. During scrubbing refreshes the $\overline{\text{CAS}}$ outputs will negate along with the $\overline{\text{RAS}}$ outputs regardless of the state of the $\overline{\text{ECAS}}$ inputs. The $\overline{\text{RFIP}}$ output will function as ReFresh ReQuest ($\overline{\text{RFRQ}}$) when this mode is programmed. The $\overline{\text{DT}}/\overline{\text{OE}}$ output will be negated by the input $\overline{\text{VSRL}}$ negating when this mode is programmed.
B1	Access Mode Select
0	ACCESS MODE 0: ALE pulsing high sets an internal latch. On the next positive edge of CLK, the access ($\overline{\text{RAS}}$) will start. $\overline{\text{AREQ}}$ will terminate the access.
1	ACCESS MODE 1: $\overline{\text{ADS}}$ asserted starts the access ($\overline{\text{RAS}}$) immediately. $\overline{\text{AREQ}}$ will terminate the access.
B0	Address Latch Mode
0	$\overline{\text{ADS}}$ or ALE asserted for Port A or $\overline{\text{AREQB}}$ asserted for Port B with the appropriate GRANT latch the input row, column and bank address.
1	The row, column and bank latches are fall through.
C9	Delay $\overline{\text{CAS}}$ during WRITE Accesses
0	$\overline{\text{CAS}}$ is treated the same for both READ and WRITE accesses.
1	During WRITE accesses, $\overline{\text{CAS}}$ will be asserted by the event that occurs last: $\overline{\text{CAS}}$ asserted by the internal delay line or $\overline{\text{CAS}}$ asserted on the positive edge of CLK after $\overline{\text{RAS}}$ is asserted.
C8	Row Address Hold Time
0	Row Address Hold Time = 25 ns minimum
1	Row Address Hold Time = 15 ns minimum
C7	Column Address Setup Time
0	Column Address Setup Time = 10 ns minimum
1	Column Address Setup Time = 0 ns minimum
C6, C5, C4	$\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ Configuration Modes/Error Scrubbing during Refresh
0, 0, 0	$\overline{\text{RAS}}_0\text{--}3$ and $\overline{\text{CAS}}_0\text{--}3$ are all selected during an access. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted. B0 and B1 are not used during an access. Error scrubbing during refresh.
0, 0, 1	$\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ pairs are selected during an access by B1. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted. B1 = 0 during an access selects $\overline{\text{RAS}}_0\text{--}1$ and $\overline{\text{CAS}}_0\text{--}1$. B1 = 1 during an access selects $\overline{\text{RAS}}_2\text{--}3$ and $\overline{\text{CAS}}_2\text{--}3$. B0 is not used during an Access. Error scrubbing during refresh.
0, 1, 0	$\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ singles are selected during an access by B0–1. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted. B1 = 0, B0 = 0 during an access selects $\overline{\text{RAS}}_0$ and $\overline{\text{CAS}}_0$. B1 = 0, B0 = 1 during an access selects $\overline{\text{RAS}}_1$ and $\overline{\text{CAS}}_1$. B1 = 1, B0 = 0 during an access selects $\overline{\text{RAS}}_2$ and $\overline{\text{CAS}}_2$. B1 = 1, B0 = 1 during an access selects $\overline{\text{RAS}}_3$ and $\overline{\text{CAS}}_3$. Error scrubbing during refresh.
0, 1, 1	$\overline{\text{RAS}}_0\text{--}3$ and $\overline{\text{CAS}}_0\text{--}3$ are all selected during an access. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted. B1, B0 are not used during an access. No error scrubbing. ($\overline{\text{RAS}}$ only refreshing)

9.0 Programming and Resetting (Continued)

9.4 PROGRAMMING BIT DEFINITIONS (Continued)

Symbol	Description
C6, C5, C4	$\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ Configuration Modes (Continued)
1, 0, 0	<p>$\overline{\text{RAS}}$ pairs are selected by B1. $\overline{\text{CAS}}0-3$ are all selected. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted.</p> <p>B1 = 0 during an access selects $\overline{\text{RAS}}0-1$ and $\overline{\text{CAS}}0-3$.</p> <p>B1 = 1 during an access selects $\overline{\text{RAS}}2-3$ and $\overline{\text{CAS}}0-3$.</p> <p>B0 is not used during an access.</p> <p>No error scrubbing.</p>
1, 0, 1	<p>$\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ pairs are selected by B1. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted.</p> <p>B1 = 0 during an access selects $\overline{\text{RAS}}0-1$ and $\overline{\text{CAS}}0-1$.</p> <p>B1 = 1 during an access selects $\overline{\text{RAS}}2-3$ and $\overline{\text{CAS}}2-3$.</p> <p>B0 is not used during an access.</p> <p>No error scrubbing.</p>
1, 1, 0	<p>$\overline{\text{RAS}}$ singles are selected by B0-1. $\overline{\text{CAS}}0-3$ are all selected. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted.</p> <p>B1 = 0, B0 = 0 during an access selects $\overline{\text{RAS}}0$ and $\overline{\text{CAS}}0-3$.</p> <p>B1 = 0, B0 = 1 during an access selects $\overline{\text{RAS}}1$ and $\overline{\text{CAS}}0-3$.</p> <p>B1 = 1, B0 = 0 during an access selects $\overline{\text{RAS}}2$ and $\overline{\text{CAS}}0-3$.</p> <p>B1 = 1, B0 = 1 during an access selects $\overline{\text{RAS}}3$ and $\overline{\text{CAS}}0-3$.</p> <p>No error scrubbing.</p>
1, 1, 1	<p>$\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ singles are selected by B0, 1. For a particular $\overline{\text{CAS}}$ to be asserted, its corresponding $\overline{\text{ECAS}}$ input must be asserted.</p> <p>B1 = 0, B0 = 0 during an access selects $\overline{\text{RAS}}0$ and $\overline{\text{CAS}}0$.</p> <p>B1 = 0, B0 = 1 during an access selects $\overline{\text{RAS}}1$ and $\overline{\text{CAS}}1$.</p> <p>B1 = 1, B0 = 0 during an access selects $\overline{\text{RAS}}2$ and $\overline{\text{CAS}}2$.</p> <p>B1 = 1, B0 = 1 during an access selects $\overline{\text{RAS}}3$ and $\overline{\text{CAS}}3$.</p> <p>No error scrubbing.</p>
C3	Refresh Clock Fine Tune Divisor
0	Divide delay line/refresh clock further by 30 (If DELCLK/Refresh Clock Clock Divisor = 2 MHz = 15 μs refresh period).
1	Divide delay line/refresh clock further by 26 (If DELCLK/Refresh Clock Clock Divisor = 2 MHz = 13 μs refresh period).
C2, C1, C0	Delay Line/Refresh Clock Divisor Select
0, 0, 0	Divide DELCLK by 10 to get as close to 2 MHz as possible.
0, 0, 1	Divide DELCLK by 9 to get as close to 2 MHz as possible.
0, 1, 0	Divide DELCLK by 8 to get as close to 2 MHz as possible.
0, 1, 1	Divide DELCLK by 7 to get as close to 2 MHz as possible.
1, 0, 0	Divide DELCLK by 6 to get as close to 2 MHz as possible.
1, 0, 1	Divide DELCLK by 5 to get as close to 2 MHz as possible.
1, 1, 0	Divide DELCLK by 4 to get as close to 2 MHz as possible.
1, 1, 1	Divide DELCLK by 3 to get as close to 2 MHz as possible.
R9	Refresh Mode Select
0	$\overline{\text{RAS}}0-3$ will all assert and negate at the same time during a refresh.
1	Staggered Refresh. $\overline{\text{RAS}}$ outputs during refresh are separated by one positive clock edge. Depending on the configuration mode chosen, either one or two $\overline{\text{RAS}}$ s will be asserted.
R8	Address Pipelining Select
0	Address pipelining is selected. The VRAM controller will switch the VRAM column address back to the row address after guaranteeing the column address hold time.
1	Non-address pipelining is selected. The VRAM controller will hold the column address on the VRAM address bus until the access $\overline{\text{RAS}}$ s are negated.

9.0 Programming and Resetting (Continued)

9.4 PROGRAMMING BIT DEFINITIONS (Continued)

Symbol	Description
R7	WAIT or DTACK Select
0	WAIT type output is selected.
1	DTACK (Data Transfer ACKnowledge) type output is selected.
R6	Add Wait States to the Current Access if WAITIN is Low
0	WAIT or DTACK will be delayed by one additional positive edge of CLK.
1	WAIT or DTACK will be delayed by two additional positive edges of CLK.
R5, R4	WAIT/DTACK during Burst (See Section 5.1.2 or 5.2.2)
0, 0	NO WAIT STATES; If R7 = 0 during programming, WAIT will remain negated during burst portion of access. If R7 = 1 programming, DTACK will remain asserted during burst portion of access.
0, 1	1T; If R7 = 0 during programming, WAIT will assert when the ECAS inputs are negated with AREQ asserted. WAIT will negate from the positive edge of CLK after the ECASs have been asserted. If R7 = 1 during programming, DTACK will negate when the ECAS inputs are negated with AREQ asserted. DTACK will assert from the positive edge of CLK after the ECASs have been asserted.
1, 0	1/2T; If R7 = 0 during programming, WAIT will assert when the ECAS inputs are negated with AREQ asserted. WAIT will negate on the negative level of CLK after the ECASs have been asserted. If R7 = 1 during programming, DTACK will negate when the ECAS inputs are negated with AREQ asserted. DTACK will assert from the negative level of CLK after the ECASs have been asserted.
1, 1	0T; If R7 = 0 during programming, WAIT will assert when the ECAS inputs are negated. WAIT will negate when the ECAS inputs are asserted. If R7 = 1 during programming, DTACK will negate when the ECAS inputs are negated. DTACK will assert when the ECAS inputs are asserted.
R3, R2	WAIT/DTACK Delay Times (See Section 5.1.1 or 5.2.1)
0, 0	NO WAIT STATES; If R7 = 0 during programming, WAIT will remain high during non-delayed accesses. WAIT will negate when RAS is negated during delayed accesses. NO WAIT STATES; If R7 = 1 during programming, DTACK will be asserted when RAS is asserted.
0, 1	1/2T; If R7 = 0 during programming, WAIT will negate on the negative level of CLK, after the access RAS. 1T; If R7 = 1 during programming, DTACK will be asserted on the positive edge of CLK after the access RAS.
1, 0	NO WAIT STATES, 1/2T; If R7 = 0 during programming, WAIT will remain high during non-delayed accesses. WAIT will negate on the negative level of CLK, after the access RAS, during delayed accesses. 1/2T; If R7 = 1 during programming, DTACK will be asserted on the negative level of CLK after the access RAS.
1, 1	1T; If R7 = 0 during programming, WAIT will negate on the positive edge of CLK after the access RAS. 1 1/2T; If R7 = 1 during programming, DTACK will be asserted on the negative level of CLK after the positive edge of CLK after the access RAS.
R1, R0	RAS Low and RAS Precharge Time
0, 0	RAS asserted during refresh = 2 positive edges of CLK. RAS precharge time = 1 positive edge of CLK. RAS will start from the first positive edge of CLK after GRANTB transitions (DP8522A).
0, 1	RAS asserted during refresh = 3 positive edges of CLK. RAS precharge time = 2 positive edges of CLK. RAS will start from the second positive edge of CLK after GRANTB transitions (DP8522A).
1, 0	RAS asserted during refresh = 2 positive edges of CLK. RAS precharge time = 2 positive edges of CLK. RAS will start from the first positive edge of CLK after GRANTB transitions (DP8522A).
1, 1	RAS asserted during refresh = 4 positive edges of CLK. RAS precharge time = 3 positive edges of CLK. RAS will start from the second positive edge of CLK after GRANTB transitions (DP8522A).

Note 1: The configuration modes allow RASs and CASs to be grouped such that each RAS and CAS will drive one-fourth of the total VRAM array whether the array is organized as 1, 2, or 4 banks.

Note 2: In order for a CAS output to go low during an access, it must be both selected and enabled. ECAS0–1 are used to enable the CAS outputs (ECAS0 enables CAS0,1; ECAS1 enables CAS2,3). Selection is determined by the configuration mode and B1, B0.

10.0 Test Mode

Staggered refresh in combination with the error scrubbing mode places the DP8520A/21A/22A in test mode. In this mode, the 24-bit refresh counter is divided into a 13-bit and 11-bit counter. During refreshes both counters are incremented to reduce test time.

11.0 VRAM Critical Timing Parameters

The two critical timing parameters, shown in *Figure 55*, that must be met when controlling the access timing to a VRAM are the row address hold time, t_{RAH} , and the column address setup time, t_{ASC} . Since the DP8520A/21A/22A contain a precise internal delay line, the values of these parameters can be selected at programming time. These values will also increase and decrease if DELCLK varies from 2 MHz.

11.1 PROGRAMMABLE VALUES OF t_{RAH} AND t_{ASC}

The DP8520A/21A/22A allow the values of t_{RAH} and t_{ASC} to be selected at programming time. For each parameter, two choices can be selected. t_{RAH} , the row address hold time, is measured from \overline{RAS} asserted to the row address starting to change to the column address. The two choices for t_{RAH} are 15 ns and 25 ns, programmable through address bit C8.

t_{ASC} , the column address setup time, is measured from the column address valid to \overline{CAS} asserted. The two choices for t_{ASC} are 0 ns and 10 ns, programmable through address bit C7.

11.2 CALCULATION OF t_{RAH} AND t_{ASC}

There are two clock inputs to the DP8520A/21A/22A. These two clocks, DELCLK and CLK can either be tied together to the same clock or be tied to different clocks running asynchronously at different frequencies.

The clock input, DELCLK, controls the internal delay line and refresh request clock. DELCLK should be a multiple of 2 MHz. If DELCLK is not a multiple of 2 MHz, t_{RAH} and t_{ASC} will change. The new values of t_{RAH} and t_{ASC} can be calculated by the following formulas:

If t_{RAH} was programmed to equal 15 ns then $t_{RAH} = 30 * (((DELCLK \text{ Divisor}) * 2 \text{ MHz} / (DELCLK \text{ Frequency})) - 1) + 15 \text{ ns}$.

If t_{RAH} was programmed to equal 25 ns then $t_{RAH} = 30 * (((DELCLK \text{ Divisor}) * 2 \text{ MHz} / (DELCLK \text{ Frequency})) - 1) + 25 \text{ ns}$.

If t_{ASC} was programmed to equal 0 ns then $t_{ASC} = 15 * ((DELCLK \text{ Divisor}) * 2 \text{ MHz} / (DELCLK \text{ Frequency})) - 15 \text{ ns}$.

If t_{ASC} was programmed to equal 10 ns then $t_{ASC} = 25 * ((DELCLK \text{ Divisor}) * 2 \text{ MHz} / (DELCLK \text{ Frequency})) - 15 \text{ ns}$.

Since the values of t_{RAH} and t_{ASC} are increased or decreased, the time to \overline{CAS} asserted will also increase or decrease. These parameters can be adjusted by the following formula:

Delay to \overline{CAS} = Actual Spec. + Actual t_{RAH} - Programmed t_{RAH} + Actual t_{ASC} - Programmed t_{ASC} .

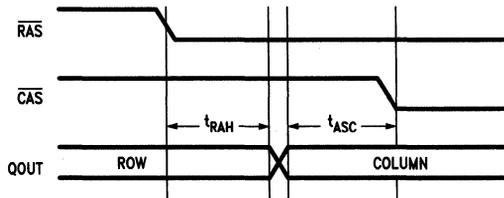


FIGURE 55. t_{RAH} and t_{ASC}

TL/F/9338-A4

12.0 Dual Accessing Functions (DP8522A)

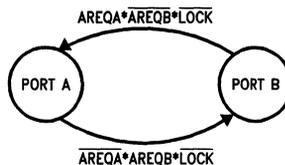
The DP8522A has all the functions previously described. In addition to those features, the DP8522A also has the capabilities to arbitrate among refresh, Port A and a second port, Port B. This allows two CPUs to access a common VRAM array. VRAM refresh has the highest priority followed by the currently granted port. The ungranted port has the lowest priority. The last granted port will continue to stay granted even after the access has terminated, until an access request is received from the ungranted port (see *Figure 56a*). The dual access configuration assumes that both Port A and Port B are synchronous to the system clock. If they are not synchronous to the system clock they should be externally synchronized (Ex. By running the access requests through several Flip-Flops, see *Figure 58a*).

12.1 PORT B ACCESS MODES (DP8522A)

Port B accesses are initiated from a single input, $\overline{\text{AREQB}}$. When $\overline{\text{AREQB}}$ is asserted, an access request is generated. If GRANTB is asserted and a refresh is not taking place or precharge time is not required, $\overline{\text{RAS}}$ will be asserted when $\overline{\text{AREQB}}$ is asserted. Once $\overline{\text{AREQB}}$ is asserted, it must stay asserted until the access is over. $\overline{\text{AREQB}}$ negated, negates $\overline{\text{RAS}}$ as shown in *Figure 56b*. Note that if $\text{ECAS0} = 1$ during programming the $\overline{\text{CAS}}$ outputs may be held asserted (beyond $\overline{\text{RASn}}$ negating) by continuing to assert the appropriate ECAS input (the same as Port A accesses). If Port B is not granted, the access will begin on the first or second positive edge of CLK after GRANTB is asserted (See R0, R1 programming bit definitions) as shown in *Figure 56c*, as-

suming that Port A is not accessing the VRAM ($\overline{\text{CS}}$, $\overline{\text{ADS/ALE}}$ and $\overline{\text{AREQ}}$) and $\overline{\text{RAS}}$ precharge for the particular bank has completed. It is important to note that for GRANTB to transition to Port B, Port A must **not** be requesting an access at a rising clock edge (or locked) and Port B must be requesting an access at that rising clock edge. Port A can request an access through $\overline{\text{CS}}$ and $\overline{\text{ADS/ALE}}$ or $\overline{\text{CS}}$ and $\overline{\text{AREQ}}$. Therefore during an interleaved access where $\overline{\text{CS}}$ and $\overline{\text{ADS/ALE}}$ become asserted before $\overline{\text{AREQ}}$ from the previous access is negated, Port A will retain $\text{GRANTB} = 0$ whether $\overline{\text{AREQB}}$ is asserted or not.

Since there is no chip select for Port B, $\overline{\text{AREQB}}$ must incorporate this signal. This mode of accessing is similar to Mode 1 accessing for Port A.



TL/F/9338-A6

Explanation of Terms

AREQA = Chip Selected access request from Port A

AREQB = Chip Selected access request from Port B

LOCK = Externally controlled LOCKing of the Port that is currently GRANTed.

FIGURE 56a. DP8522A PORT A/PORT B ARBITRATION STATE DIAGRAM. This arbitration may take place during the "ACCESS" or "REFRESH" state (see *Figure 7a*).

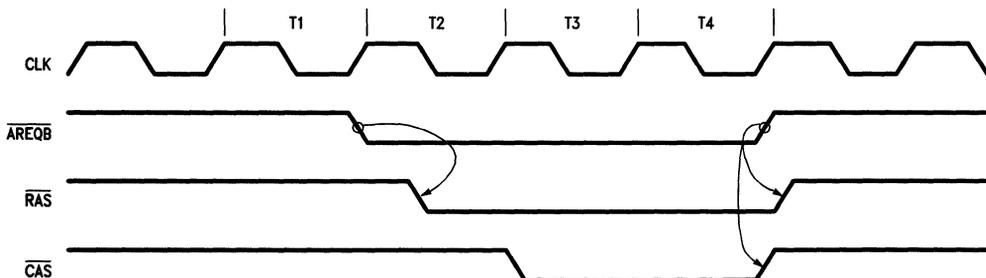


FIGURE 56b. Access Request for Port B

TL/F/9338-A6

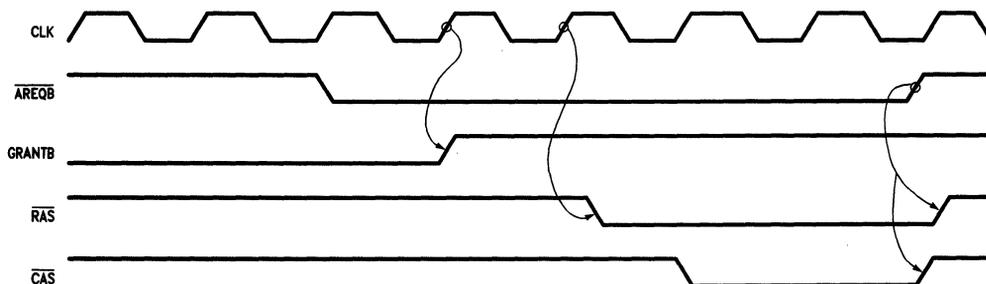


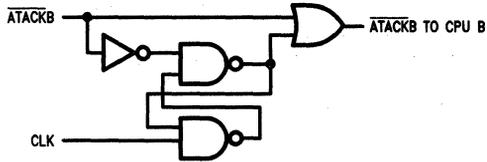
FIGURE 56c. Delayed Port B Access

TL/F/9338-A7

12.0 Dual Accessing Functions (DP8522A) (Continued)

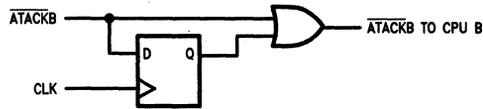
12.2 PORT B WAIT STATE SUPPORT (DP8522A)

Advanced transfer acknowledge for Port B, \overline{ATACKB} , is used for wait state support for Port B. This output will be asserted when \overline{RAS} for the Port B access is asserted, as shown in Figures 57a and 57b. Once asserted, this output will stay asserted until \overline{AREQB} is negated. With external logic, \overline{ATACKB} can be made to interface to any CPU's wait input as shown in Figure 57c.



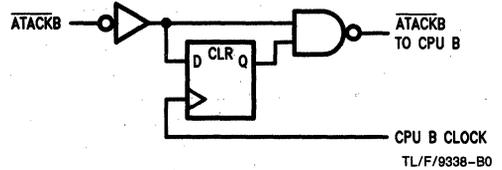
TL/F/9338-A8

A) Extend \overline{ATACK} to $1/2T$ ($1/2$ Clock) after \overline{RAS} goes low.



TL/F/9338-A9

B) Extend \overline{ATACK} to $1T$ after \overline{RAS} goes low.



C) Synchronize \overline{ATACKB} to CPU B Clock. This is useful if CPU B runs asynchronous to the DP8522.

FIGURE 57c. Modifying Wait Logic for Port B

12.3 COMMON PORT A AND PORT B DUAL PORT FUNCTIONS

An input, \overline{LOCK} , and an output, \overline{GRANTB} , add additional functionality to the dual port arbitration logic. \overline{LOCK} allows Port A or Port B to lock out the other port from the VRAM. When a Port is locked out of the VRAM, wait states will be inserted into its access cycle until it is allowed to access memory. \overline{GRANTB} is used to multiplex the input control signals and addresses to the DP8522A.

12.3.1 \overline{GRANTB} Output

The output \overline{GRANTB} determines which port has current access to the VRAM array. \overline{GRANTB} asserted signifies Port B has access. \overline{GRANTB} negated signifies Port A has access to the VRAM array.

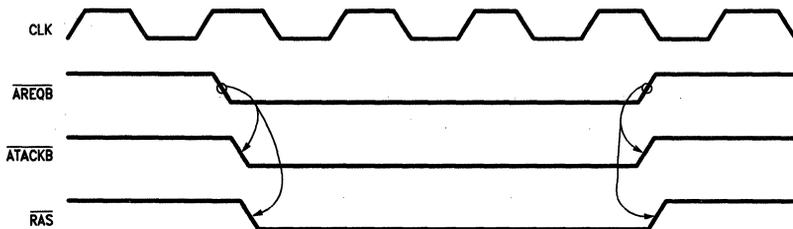


FIGURE 57a. Non-Delayed Port B Access

TL/F/9338-B1

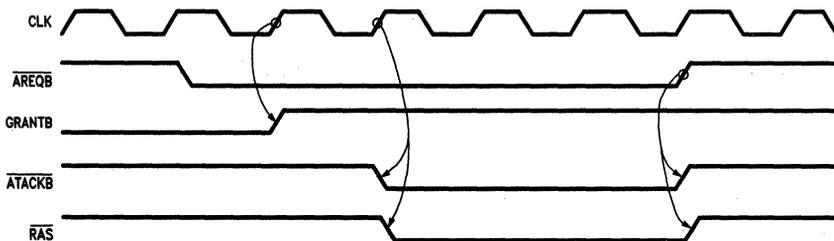


FIGURE 57b. Delayed Port B Access

TL/F/9338-B2

12.0 Dual Accessing Functions (DP8522A) (Continued)

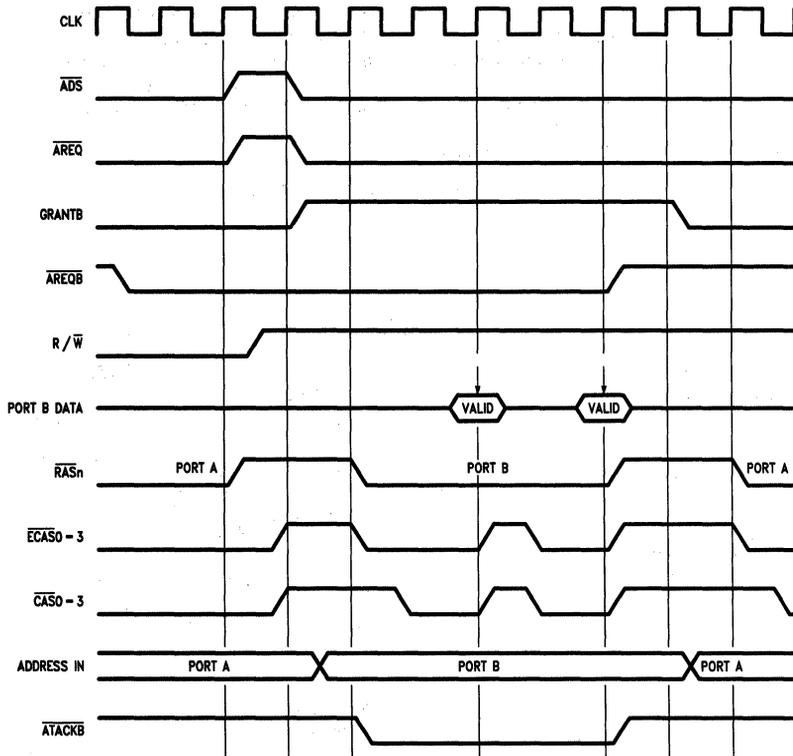


FIGURE 58b. Wait States during a Port B Access

TL/F/9338-B3

12.3.2 LOCK Input

When the \overline{LOCK} input is asserted, the currently granted port can "lock out" the other port through the insertion of wait states to that port's access cycle. \overline{LOCK} does not disable refreshes, it only keeps \overline{GRANTB} in the same state even if the other port requests an access, as shown in Figure 59. \overline{LOCK} can be used by either port. The user must guarantee that \overline{LOCK} is not asserted with Port B granted when $\overline{AVSRLRQ}$ is asserted to request a VRAM transfer cycle.

12.4 DUAL PORTING AND VRAM TRANSFER CYCLES

The input $\overline{AVSRLRQ}$ disables any further internally or externally requested refreshes or Port B access requests from being executed. The $\overline{AVSRLRQ}$ input does this by making the VRAM controller arbitration logic think that a Port A access is in progress from the point where the $\overline{AVSRLRQ}$ input goes low until the VRAM shift register load operation is completed.

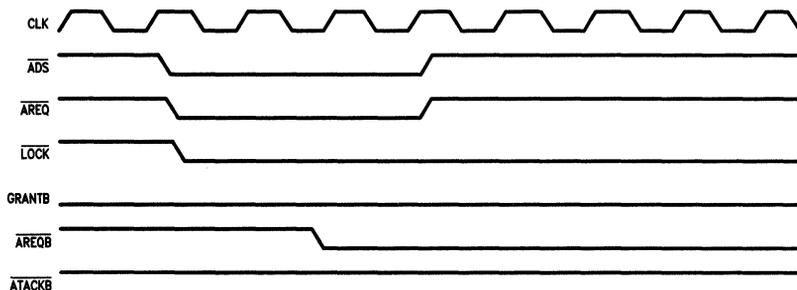


FIGURE 59. LOCK Function

TL/F/9338-B4

13.0 Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature under Bias 0°C to +70°C
 Storage Temperature -65°C to +150°C

All Input or Output Voltage
 with Respect to GND -0.5V to +7V
 Power Dissipation @ 20 MHz 0.5W
 ESD Rating to be determined.

14.0 DC Electrical Characteristics $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 10\%$, $\text{GND} = 0\text{V}$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	Logical 1 Input Voltage	Tested with a Limited Functional Pattern	2.0		$V_{CC} + 0.5$	V
V_{IL}	Logical 0 Input Voltage	Tested with a Limited Functional Pattern	-0.5		0.8	V
V_{OH1}	Q and \overline{WE} Outputs	$I_{OH} = -10\text{ mA}$	$V_{CC} - 1.0$			V
V_{OL1}	Q and \overline{WE} Outputs	$I_{OL} = 10\text{ mA}$			0.5	V
V_{OH2}	All Outputs except Qs, \overline{WE}	$I_{OH} = -3\text{ mA}$	$V_{CC} - 1.0$			V
V_{OL2}	All Outputs except Qs, \overline{WE}	$I_{OL} = 3\text{ mA}$			0.5	V
I_{IN}	Input Leakage Current	$V_{IN} = V_{CC}$ or GND	-10		10	μA
I_{IL_ML}	\overline{ML} Input Current (Low)	$V_{IN} = \text{GND}$			200	μA
I_{CC1}	Standby Current	CLK at 8 MHz ($V_{IN} = V_{CC}$ or GND)		6	15	mA
I_{CC1}	Standby Current	CLK at 20 MHz ($V_{IN} = V_{CC}$ or GND)		8	17	mA
I_{CC1}	Standby Current	CLK at 25 MHz ($V_{IN} = V_{CC}$ or GND)		10	20	mA
I_{CC2}	Supply Current	CLK at 8 MHz (Inputs Active) ($I_{LOAD} = 0$) ($V_{IN} = V_{CC}$ or GND)		20	40	mA
I_{CC2}	Supply Current	CLK at 20 MHz (Inputs Active) ($I_{LOAD} = 0$) ($V_{IN} = V_{CC}$ or GND)		40	75	mA
I_{CC2}	Supply Current	CLK at 25 MHz (Inputs Active) ($I_{LOAD} = 0$) ($V_{IN} = V_{CC}$ or GND)		50	95	mA
C_{IN}^*	Input Capacitance	f_{IN} at 1 MHz			10	pF

*Note: C_{IN} is not 100% tested.

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A

Two speed selections are given, the DP8520A/21A/22A-20 and the DP8520A/21A/22A-25. The differences between the two parts are the maximum operating frequencies of the input CLKs and the maximum delay specifications. Low frequency applications may use the "-25" part to gain improved timing.

The AC timing parameters are grouped into sectional numbers as shown below. These numbers also refer to the timing diagrams.

- 1-38 Common parameters to all modes of operation
- 50-56 Difference parameters used to calculate;
 - \overline{RAS} low time,
 - \overline{RAS} precharge time,
 - \overline{CAS} high time and
 - \overline{CAS} low time
- 100-121 Common dual access parameters used for Port B accesses and inputs and outputs used only in dual accessing
- 200-212 Refresh parameters

- 300-315 Mode 0 access parameters used in both single and dual access applications
 - 400-416 Mode 1 access parameters used in both single and dual access applications
 - 450-455 Special Mode 1 access parameters which supersede the 400-416 parameters when dual accessing
 - 500-506 Programming parameters
 - 600-605 Graphics parameters for VRAM transfer cycles
- Unless otherwise stated $V_{CC} = 5.0\text{V} \pm 10\%$, $0 < T_A < 70^\circ\text{C}$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

- $C_L = 50\text{ pF}$ loads on all outputs except
- $C_L = 150\text{ pF}$ loads on Q0-8, 9, and 10; or
- $C_H = 50\text{ pF}$ loads on all outputs except
- $C_H = 125\text{ pF}$ loads on \overline{RAS} -3 and \overline{CAS} -3 and
- $C_H = 380\text{ pF}$ loads on Q0-8, 9, and 10.

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Number	Symbol	Common Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C _L		C _H		C _L		C _H	
			Min	Max	Min	Max	Min	Max	Min	Max
1	fCLK	CLK Frequency	0	20	0	20	0	25	0	25
2	tCLKP	CLK Period	50		50		40		40	
3, 4	tCLKPW	CLK Pulse Width	15		15		12		12	
5	fDCLK	DELCLK Frequency	5	20	5	20	5	20	5	20
6	tDCLKP	DELCLK Period	50	200	50	200	50	200	50	200
7, 8	tDCLKPW	DELCLK Pulse Width	15		15		12		12	
9a	tPRASCAS0	\overline{RAS} Asserted to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 0 ns)	30		30		30		30	
9b	tPRASCAS1	\overline{RAS} Asserted to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 10 ns)	40		40		40		40	
9c	tPRASCAS2	\overline{RAS} Asserted to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 0 ns)	40		40		40		40	
9d	tPRASCAS3	\overline{RAS} Asserted to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 10 ns)	50		50		50		50	
10a	tRAH	Row Address Hold Time (tRAH = 15)	15		15		15		15	
10b	tRAH	Row Address Hold Time (tRAH = 25)	25		25		25		25	
11a	tASC	Column Address Setup Time (tASC = 0)	0		0		0		0	
11b	tASC	Column Address Setup Time (tASC = 10)	10		10		10		10	
12	tPCKRAS	CLK High to \overline{RAS} Asserted following Precharge		27		32		22		26
13	tPARQRAS	\overline{AREQ} Negated to \overline{RAS} Negated		38		43		31		35
14	tPENCL	$\overline{ECAS0-1}$ Asserted to \overline{CAS} Asserted		23		31		20		27
15	tPENCH	$\overline{ECAS0-1}$ Negated to \overline{CAS} Negated		25		33		20		27
16	tPARQCAS	\overline{AREQ} Negated to \overline{CAS} Negated		60		68		47		54
17	tPCLKWH	CLK to \overline{WAIT} Negated		39		39		31		31
18	tPCLKDLO	CLK to \overline{DTACK} Asserted (Programmed as DTACK of 1/2, 1, 1½ or if \overline{WAITIN} is Asserted)		33		33		28		28
19	tPEWL	\overline{ECAS} Negated to \overline{WAIT} Asserted during a Burst Access		42		42		34		34
20	tSECK	\overline{ECAS} Asserted Setup to CLK High to Recognize the Rising Edge of CLK during a Burst Access	24		24		19		19	

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0–8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on RAS0–3 and CAS0–3 and

$C_H = 380$ pF loads on Q0–8, 9 and 10.

Number	Symbol	Common Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
21	tPEDL	ECAS Asserted to DTACK Asserted during a Burst Access (Programmed as DTACK0)		48		48		38		38
22	tPEDH	ECAS Negated to DTACK Negated during a Burst Access		49		49		38		38
23	tSWCK	WAITIN Asserted Setup to CLK	5		5		5		5	
26	tPAQ	Row, Column Address Valid to Q0–8, 9, 10 Valid		29		38		26		35
27	tPCINCQ	COLINC Asserted to Q0–8, 9, 10 Incremented		34		43		30		39
28	tSCINEN	COLINC Asserted Setup to ECAS Asserted to Ensure tASC = 0 ns	16		17		15		17	
29a	tSARQCK1	AREQ Negated Setup to CLK High with 1 Period of Precharge	43		43		34		34	
29b	tSARQCK2	AREQ Negated Setup to CLK High with > 1 Period of Precharge Programmed	19		19		15		15	
30	tPAREQDH	AREQ Negated to DTACK Negated		34		34		27		27
31	tPCKCAS	CLK High to CAS Asserted when Delayed by WIN		31		39		25		32
32	tSCADEN	Column Address Setup to ECAS Asserted to Guarantee tASC = 0	14		15		14		16	
33	tWCINC	COLINC Pulse Width	20		20		20		20	
34a	tPCKCL0	CLK High to CAS Asserted following Precharge (tRAH = 15 ns, tASC = 0 ns)		81		89		72		79
34b	tPCKCL1	CLK High to CAS Asserted following Precharge (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
34c	tPCKCL2	CLK High to CAS Asserted following Precharge (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
34d	tPCKCL3	CLK High to CAS Asserted following Precharge (tRAH = 25 ns, tASC = 10 ns)		101		109		92		99
35	tCAH	Column Address Hold Time (Interleave Mode Only)	32		32		32		32	
36	tPCQR	CAS Asserted to Row Address Valid (Interleave Mode Only)		90		90		90		90
37	tPCASDTH	CAS Asserted to DT/OE Asserted for a VRAM Read Access		15		25		14		24
38	tPCASDTL	CAS Negated to DT/OE Negated for a VRAM Read Access		15		25		14		24

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0-8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on $\overline{RAS0-3}$ and $\overline{CAS0-3}$ and

$C_H = 380$ pF loads on Q0-8, 9 and 10.

Number	Symbol	Difference Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
50	tD1	(\overline{AREQ} or \overline{AREQB} Negated to \overline{RAS} Negated) Minus (CLK High to \overline{RAS} Asserted)		16		16		14		14
51	tD2	(CLK High to Refresh \overline{RAS} Negated) Minus (CLK High to \overline{RAS} Asserted)		13		13		11		11
52	tD3a	(\overline{ADS} Asserted to \overline{RAS} Asserted (Mode 1)) Minus (\overline{AREQ} Negated to \overline{RAS} Negated)		4		4		4		4
53	tD3b	(CLK High to \overline{RAS} Asserted (Mode 0)) Minus (\overline{AREQ} Negated to \overline{RAS} Negated)		4		4		4		4
54	tD4	(\overline{ECAS} Asserted to \overline{CAS} Asserted) Minus (\overline{ECAS} Negated to \overline{CAS} Negated)	-7	7	-7	7	-7	7	-7	7
55	tD5	(CLK to Refresh \overline{RAS} Asserted) Minus (CLK to Refresh \overline{RAS} Negated)		5		5		5		5
56	tD6	(\overline{AREQ} Negated to \overline{RAS} Negated) Minus (\overline{ADS} Asserted to \overline{RAS} Asserted ((Mode 1)))		12		12		10		10

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0–8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on RAS0–3 and CAS0–3 and

$C_H = 380$ pF loads on Q0–8, 9 and 10.

Number	Symbol	Common Dual Access Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
100	tHCKARQB	\overline{AREQB} Negated Held from CLK High	3		3		3		3	
101	tSARQBCK	\overline{AREQB} Asserted Setup to CLK High	8		8		7		7	
102	tPAQBRASL	\overline{AREQB} Asserted to \overline{RAS} Asserted		37		42		30		34
103	tPAQBRASH	\overline{AREQB} Negated to \overline{RAS} Negated		37		42		29		33
105	tPCKRASG	CLK High to \overline{RAS} Asserted for Pending Port B Access		48		53		38		42
106	tPAQBATKBL	\overline{AREQB} Asserted to \overline{ATACKB} Asserted		48		48		38		38
107	tPCKATKB	CLK High to \overline{ATACKB} Asserted for Pending Access		59		59		47		47
108	tPCKGH	CLK High to GRANTB Asserted		38		38		30		30
109	tPCKGL	CLK High to GRANTB Negated		32		32		26		26
110	tSADDCKG	Row Address Setup to CLK High That Asserts \overline{RAS} following a GRANTB Change to Ensure tASR = 0 ns for Port B	11		15		11		16	
111	tSLOCKCK	\overline{LOCK} Asserted Setup to CLK Low to Lock Current Port	5		5		5		5	
112	tPAQATKBH	\overline{AREQ} Negated to \overline{ATACKB} Negated		26		26		21		21
113	tPAQBCASH	\overline{AREQB} Negated to \overline{CAS} Negated		59		67		47		54
114	tSADAQB	Address Valid Setup to \overline{AREQB} Asserted	7		11		7		12	
116	tHCKARQG	\overline{AREQ} Negated Held from CLK High	3		3		3		3	
117	tWAQB	\overline{AREQB} High Pulse Width to Guarantee tASR = 0 ns	31		35		26		31	
118a	tPAQBCAS0	\overline{AREQB} Asserted to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 0 ns)		95		103		81		88
118b	tPAQBCAS1	\overline{AREQB} Asserted to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 10 ns)		105		113		91		98
118c	tPAQBCAS2	\overline{AREQB} Asserted to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 0 ns)		105		113		91		98
118d	tPAQBCAS3	\overline{AREQB} Asserted to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 10 ns)		115		123		101		108
120a	tPCKCASG0	CLK High to \overline{CAS} Asserted for Pending Port B Access (tRAH = 15 ns, tASC = 0 ns)		101		109		86		93
120b	tPCKCASG1	CLK High to \overline{CAS} Asserted for Pending Port B Access (tRAH = 15 ns, tASC = 0 ns)		111		119		96		103
120c	tPCKCASG2	CLK High to \overline{CAS} Asserted for Pending Port B Access (tRAH = 25 ns, tASC = 0 ns)		111		119		96		103
120d	tPCKCASG3	CLK High to \overline{CAS} Asserted for Pending Port B Access (tRAH = 25 ns, tASC = 10 ns)		121		129		106		113
121	tSBADDCKG	Bank Address Valid Setup to CLK High That Starts \overline{RAS} for Pending Port B Access	10		10		10		10	

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0–8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on RAS0–3 and CAS0–3 and

$C_H = 380$ pF loads on Q0–8, 9 and 10.

Number	Symbol	Refresh Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
200	tSRFCK	\overline{RFSH} Asserted Setup to CLK High	17		17		15		15	
201	tSDRFCK	$\overline{DISRFSH}$ Asserted Setup to CLK High	18		18		15		15	
202	tSXRFC	EXTENDRF Setup to CLK High	15		15		12		12	
204	tPCKRFL	CLK High to \overline{RFIP} Asserted		42		42		34		34
205	tPARQRF	\overline{AREQ} Negated to \overline{RFIP} Asserted		62		62		50		50
206	tPCKRFH	CLK High to \overline{RFIP} Negated		65		65		51		51
207	tPCKRFRASH	CLK High to Refresh \overline{RAS} Negated		35		40		29		33
208	tPCKRFRASL	CLK High to Refresh \overline{RAS} Asserted		28		33		23		27
209a	tPCKCLO	CLK High to \overline{CAS} Asserted during Error Scrubbing (tRAH = 15 ns, tASC = 0 ns)		82		90		73		80
209b	tPCKCL1	CLK High to \overline{CAS} Asserted during Error Scrubbing (tRAH = 15 ns, tASC = 10 ns)		92		100		83		90
209c	tPCKCL2	CLK High to \overline{CAS} Asserted during Error Scrubbing (tRAH = 25 ns, tASC = 0 ns)		92		100		83		90
209d	tPCKCL3	CLK High to \overline{CAS} Asserted during Error Scrubbing (tRAH = 25 ns, tASC = 10 ns)		102		110		83		100
210	tWRFSH	\overline{RFSH} Pulse Width	15		15		15		15	
211	tPCKRQL	CLK High to \overline{RFRQ} Asserted		46		46		40		40
212	tPCKRQH	CLK High to \overline{RFRQ} Negated		50		50		40		40

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

 Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^{\circ}C < T_A < 70^{\circ}C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

 $C_L = 50$ pF loads on all outputs except

 $C_L = 150$ pF loads on Q0-8, 9 and 10; or

 $C_H = 50$ pF loads on all outputs except

 $C_H = 125$ pF loads on $\overline{RAS0-3}$ and $\overline{CAS0-3}$ and

 $C_H = 380$ pF loads on Q0-8, 9 and 10.

Number	Symbol	Mode 0 Access Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
300	tSCSCK	\overline{CS} Asserted to CLK High	14		14		13		13	
301a	tSALECKNL	ALE Asserted Setup to CLK High Not Using On-Chip Latches or if Using On-Chip Latches and B0, B1, Are Constant, Only 1 Bank	16		16		15		15	
301b	tSALECKL	ALE Asserted Setup to CLK High, if Using On-Chip Latches if B0, B1 Can Change, More Than One Bank	29		29		29		29	
302	tWALE	ALE Pulse Width	18		18		13		13	
303	tSBADDCK	Bank Address Valid Setup to CLK High	20		20		18		18	
304	tSADDCK	Row, Column Valid Setup to CLK High to Guarantee tASR = 0 ns	11		15		11		16	
305	tHASRCB	Row, Column, Bank Address Held from ALE Negated (Using On-Chip Latches)	10		10		8		8	
306	tSRCBAS	Row, Column, Bank Address Setup to ALE Negated (Using On-Chip Latches)	3		3		2		2	
307	tPCKRL	CLK High to \overline{RAS} Asserted		27		32		22		26
308a	tPCKCL0	CLK High to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 0 ns)		81		89		72		79
308b	tPCKCL1	CLK High to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 10 ns)		91		99		82		89
308c	tPCKCL2	CLK High to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
308d	tPCKCL3	CLK High to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 10 ns)		101		109		92		99
309	tHCKALE	ALE Negated Hold from CLK High	0		0		0		0	
310	tSWINCK	\overline{WIN} Asserted Setup to CLK High to Guarantee \overline{CAS} is Delayed	-5		-5		-5		-5	
311	tPCSWL	\overline{CS} Asserted to \overline{WAIT} Asserted		26		26		22		22
312	tPCSWH	\overline{CS} Negated to \overline{WAIT} Negated		26		26		22		22
313	tPCKDL1	CLK High to \overline{DTACK} Asserted (Programmed as $\overline{DTACK0}$)		41		41		33		33
314	tPALEWL	ALE Asserted to \overline{WAIT} Asserted (\overline{CS} is Already Asserted)		48		48		39		39
315		\overline{AREQ} Negated to CLK High That Starts Access \overline{RAS} to Guarantee tASR = 0 ns (Non-Interleaved Mode Only)	41		45		34		39	

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0-8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on $\overline{RAS0-3}$ and $\overline{CAS0-3}$ and

$C_H = 380$ pF loads on Q0-8, 9 and 10.

Number	Symbol	Mode 1 Access Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
400a	tSADSK1	\overline{ADS} Asserted Setup to CLK High	8		8		7		7	
400b	tSADSKW	\overline{ADS} Asserted Setup to CLK (to Guarantee Correct WAIT or $\overline{DTACK0}$ Output; Doesn't Apply for $\overline{DTACK0}$)	31		31		25		25	
401	tSCSADS	\overline{CS} Setup to \overline{ADS} Asserted	6		6		5		5	
402	tPADSRL	\overline{ADS} Asserted to \overline{RAS} Asserted		30		35		25		29
403a	tPADSCL0	\overline{ADS} Asserted to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 0 ns)		86		94		75		82
403b	tPADSCL1	\overline{ADS} Asserted to \overline{CAS} Asserted (tRAH = 15 ns, tASC = 10 ns)		96		104		85		92
403c	tPADSCL2	\overline{ADS} Asserted to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 10 ns)		96		104		85		92
403d	tPADSCL3	\overline{ADS} Asserted to \overline{CAS} Asserted (tRAH = 25 ns, tASC = 10 ns)		106		114		95		102
404	tSADDADS	Row Address Valid Setup to \overline{ADS} Asserted to Guarantee tASR = 0 ns	9		13		9		14	
405	tHCKADS	\overline{ADS} Negated Held from CLK High	0		0		0		0	
406	tSWADS	WAITN Asserted Setup to \overline{ADS} Asserted to Guarantee $\overline{DTACK0}$ Is Delayed	0		0		0		0	
407	tSBADAS	Bank Address Setup to \overline{ADS} Asserted	11		11		11		11	
408	tHASRCB	Row, Column, Bank Address Held from \overline{ADS} Asserted (Using On-Chip Latches)	10		10		8		8	
409	tSRCBAS	Row, Column, Bank Address Setup to \overline{ADS} Asserted (Using On-Chip Latches)	3		3		2		2	
410	tWADSH	\overline{ADS} Negated Pulse Width	12		16		12		17	
411	tPADSD	\overline{ADS} Asserted to \overline{DTACK} Asserted (Programmed as $\overline{DTACK0}$)		43		43		35		35
412	tSWINADS	WIN Asserted Setup to \overline{ADS} Asserted (to Guarantee CAS Delayed during Writes Accesses)	-10		-10		-10		-10	
413	tPADSWL0	\overline{ADS} Asserted to \overline{WAIT} Asserted (Programmed as WAIT0, Delayed Access)		46		46		37		37
414	tPADSWL1	\overline{ADS} Asserted to \overline{WAIT} Asserted (Programmed \overline{WAIT} 1/2 or 1)		38		38		31		31
415	tPCKDL1	CLK High to \overline{DTACK} Asserted (Programmed as $\overline{DTACK0}$, Delayed Access)		41		41		33		33
416		\overline{AREQ} Negated to \overline{ADS} Asserted to Guarantee tASR = 0 ns (Non Interleaved Mode Only)	38		42		31		36	

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

 Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

 $C_L = 50$ pF loads on all outputs except

 $C_L = 150$ pF loads on Q0–8, 9 and 10; or

 $C_H = 50$ pF loads on all outputs except

 $C_H = 125$ pF loads on \overline{RAS} 0–3 and \overline{CAS} 0–3 and

 $C_H = 380$ pF loads on Q0–8, 9 and 10.

Number	Symbol	Mode 1 Dual Access Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
450	tSADDCKG	Row Address Setup to CLK High That Asserts \overline{RAS} following a GRANTB Port Change to Ensure tASR = 0 ns	11		15		11		16	
451	tPCKRASG	CLK High to \overline{RAS} Asserted for Pending Access		48		53		38		42
452	tPCLKDL2	CLK to \overline{DTACK} Asserted for Delayed Accesses (Programmed as DTACK0)		53		53		43		43
453a	tPCKCASG0	CLK High to \overline{CAS} Asserted for Pending Access (tRAH = 15 ns, tASC = 0 ns)		101		109		86		93
453b	tPCKCASG1	CLK High to \overline{CAS} Asserted for Pending Access (tRAH = 15 ns, tASC = 10 ns)		111		119		96		103
453c	tPCKCASG2	CLK High to \overline{CAS} Asserted for Pending Access (tRAH = 25 ns, tASC = 0 ns)		111		119		96		103
453d	tPCKCASG3	CLK High to \overline{CAS} Asserted for Pending Access (tRAH = 25 ns, tASC = 10 ns)		121		129		106		113
454	tSBADDCKG	Bank Address Valid Setp to CLK High That Asserts \overline{RAS} for Pending Access	5		5		4		4	
455	tSADSCK0	\overline{ADS} Asserted Setup to CLK High	12		12		11		11	

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^\circ C < T_A < 70^\circ C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0-8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on $\overline{RAS}0-3$ and $\overline{CAS}0-3$ and

$C_H = 380$ pF loads on Q0-8, 9 and 10.

Number	Symbol	Programming Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
500	tHMLADD	Mode Address Held from \overline{ML} Negated	6		6		5		5	
501	tSADDML	Mode Address Setup to \overline{ML} Negated	6		6		6		6	
502	tWML	\overline{ML} Pulse Width	15		15		15		15	
503	tSADAQML	Mode Address Setup to \overline{AREQ} Asserted	0		0		0		0	
504	tHADAQML	Mode Address Held from \overline{AREQ} Asserted	39		39		29		29	
505	tSCSARQ	\overline{CS} Asserted Setup to \overline{AREQ} Asserted	6		6		6		6	
506	tSMLARQ	\overline{ML} Asserted Setup to \overline{AREQ} Asserted	10		10		10		10	

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

Unless otherwise stated $V_{CC} = 5.0V \pm 10\%$, $0^{\circ}C < T_A < 70^{\circ}C$, the output load capacitance is typical for 4 banks of 18 VRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

$C_L = 50$ pF loads on all outputs except

$C_L = 150$ pF loads on Q0–8, 9 and 10; or

$C_H = 50$ pF loads on all outputs except

$C_H = 125$ pF loads on FAS0–3 and CAS0–3 and

$C_H = 380$ pF loads on Q0–8, 9 and 10.

Number	Symbol	Programming Parameter Description	8520A/21A/22A-20				8520A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
600	tSCKVSRL	\overline{VSRL} Low Setup to CLK Rising Edge to guarantee counting \overline{VSRL} as being Low (used to determine when to end Graphics Shift load access)	8		8		7		7	
601	tHVSRLCK	\overline{VSRL} Low from CLK High (to guarantee \overline{VSRL} is not counted as being Low until the next rising clock edge)	4		4		3		3	
602	tSCKAVSRL	$\overline{AVSRLRQ}$ Low before CLK Rising Edge to guarantee locking the VRAM to only Port A accesses	13		13		12		12	
603	tPVSRLDTL	\overline{VSRL} Low to $\overline{DT}/\overline{OE}$ Low during Graphics Shift Register Load access		25		35		23		33
604	tPVSRLDTH	\overline{VSRL} Negated to $\overline{DT}/\overline{OE}$ Negated		24		34		22		32
605	tPCKDTL	CLK to $\overline{DT}/\overline{OE}$ Negated		30		40		27		37

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Input pulse 0V to 3V; $t_R = t_F = 2.5$ ns. Input reference point on AC measurements is 1.5V. Output reference points are 2.4V for High and 0.8V for Low.

Note 3: AC Production testing is done at 50 pF.

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

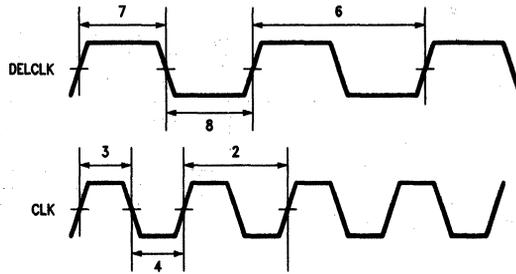


FIGURE 60. Clock, DELCLK Timing

TL/F/9338-B5

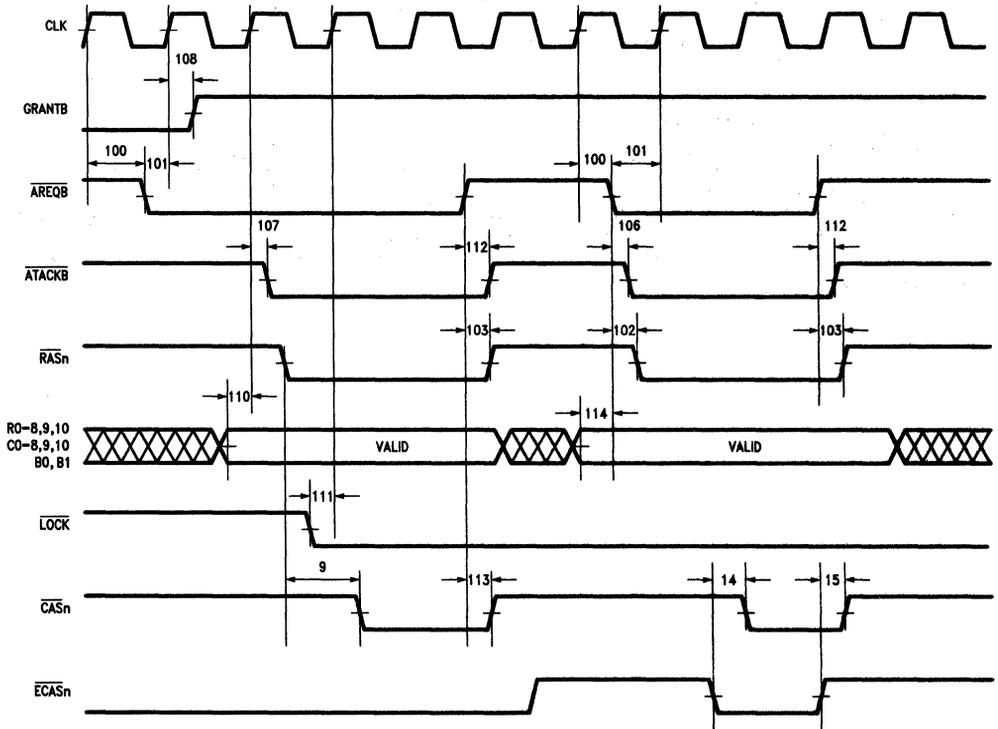


FIGURE 61. 100: Dual Access Port B

TL/F/9338-B6

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

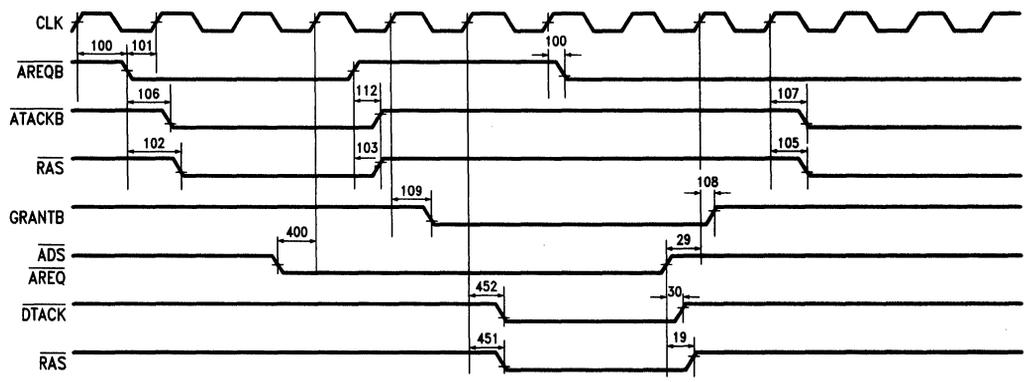


FIGURE 62. 100: Port A and Port B Dual Access

TL/F/9338-B7

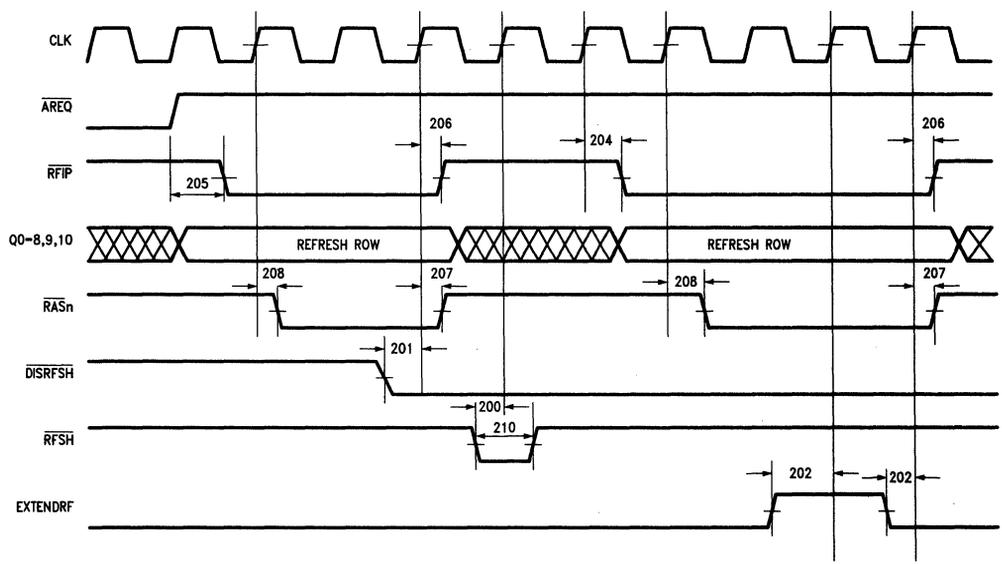


FIGURE 63. 200: Refresh Timing

TL/F/9338-B8

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

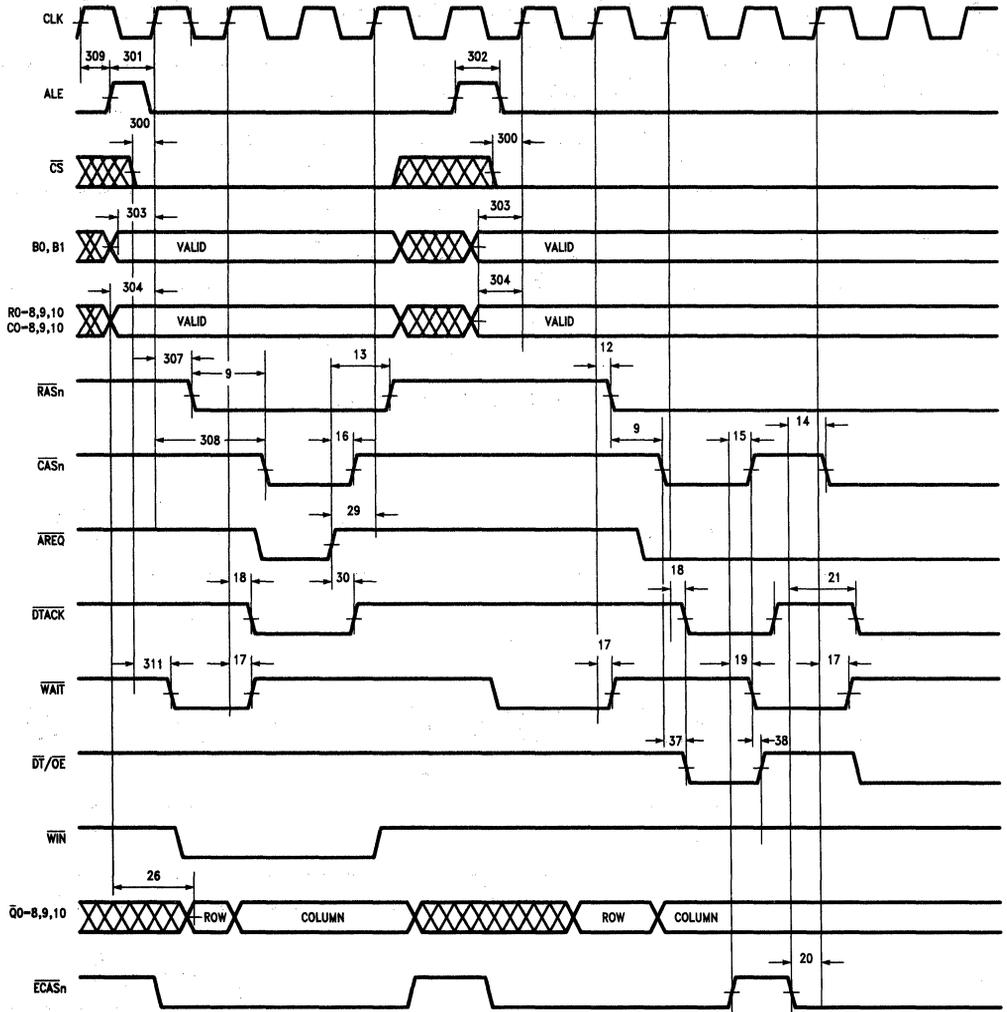
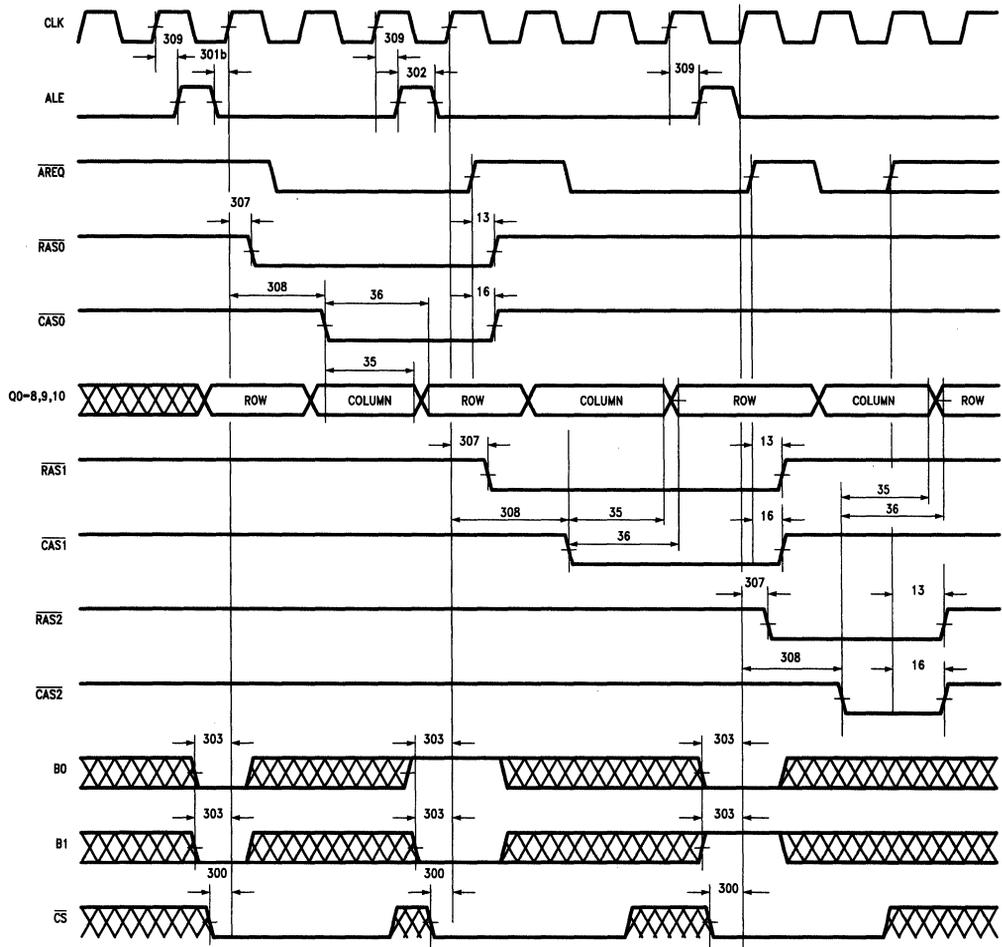


FIGURE 64. 300: Mode 0 Timing

TL/F/9338-B9

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)



TL/F/9338-C0

(Programmed as C4 = 1, C5 = 1, C6 = 1)
FIGURE 65. 300: Mode 0 Interleaving

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

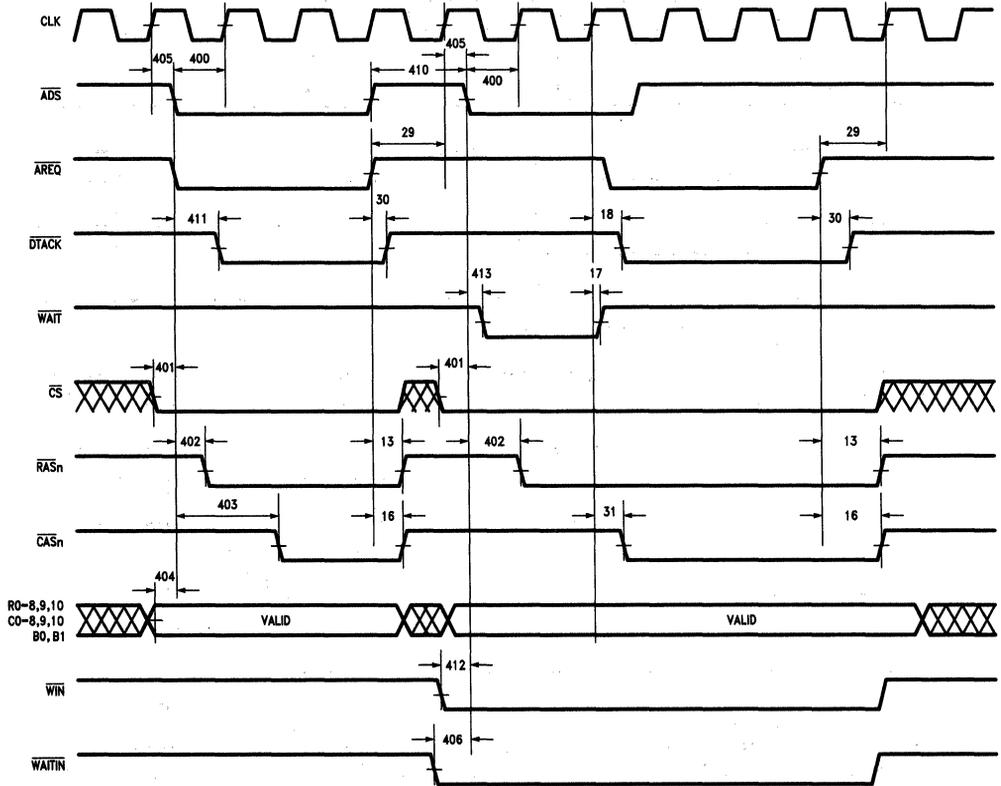


FIGURE 66. 400: Mode 1 Timing

TL/F/9338-C1

15.0 AC Timing Parameters: DP8520A/DP8521A/DP8522A (Continued)

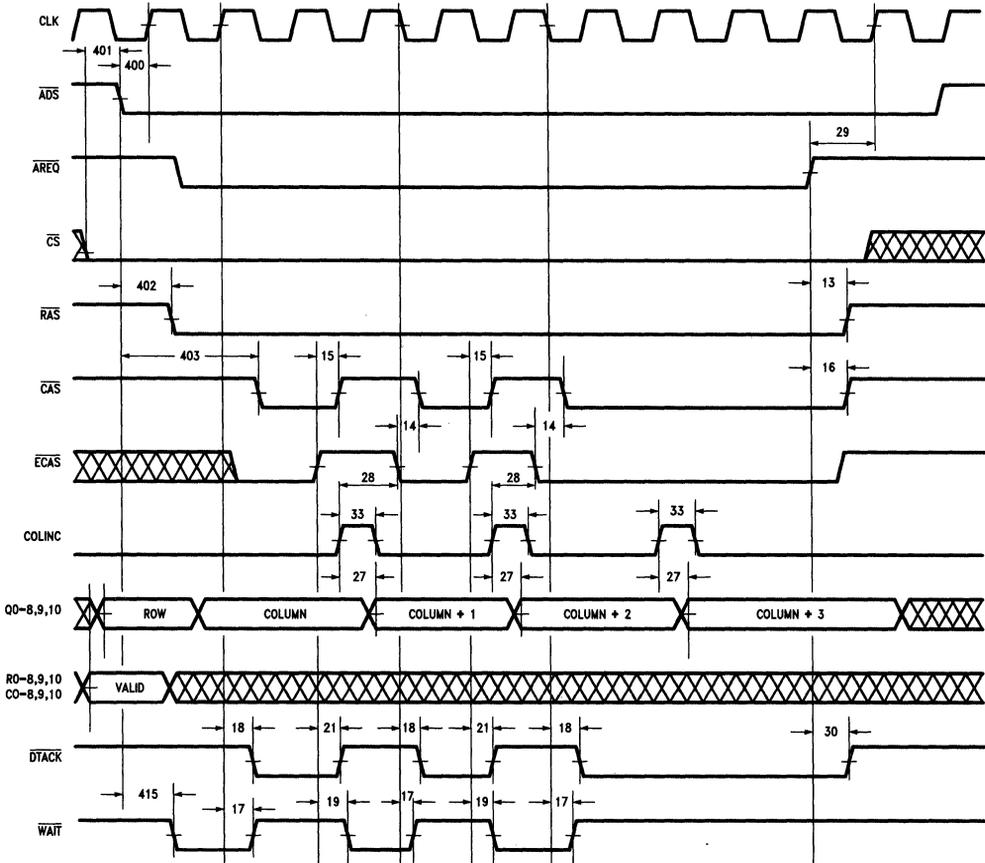


FIGURE 67. 400: COLINC Page/Static Column Access Timing

TL/F/9338-C2

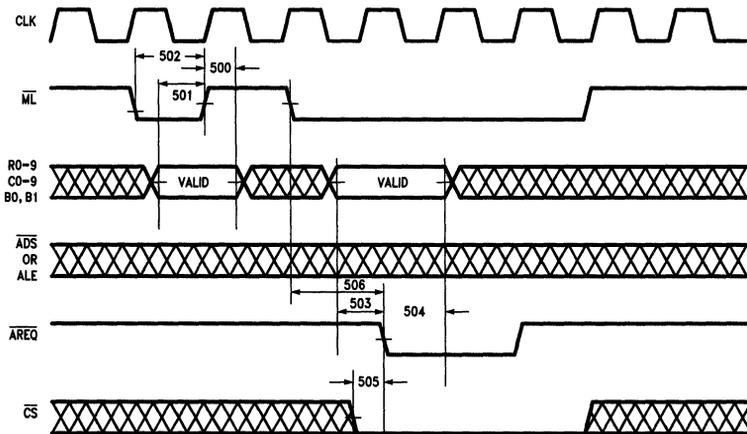


FIGURE 68. 500: Programming

TL/F/9338-C3

15.0 AC Timing Parameters: DP8520A/21A/22A (Continued)

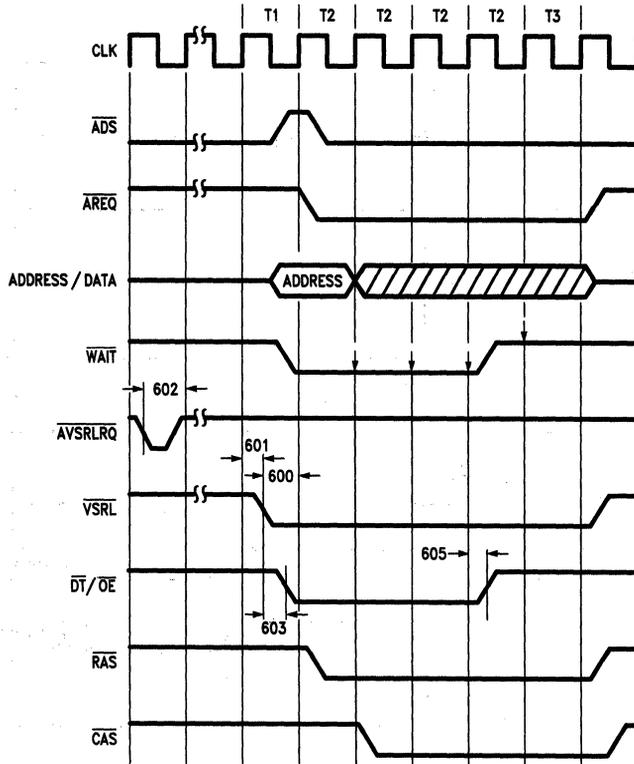


FIGURE 69. Graphics Timing Diagram

TL/F/9338-57

16.0 Functional Differences between the DP8520A/21A/22A and the DP8520/21/22

1. Extending the Column Address Strobe ($\overline{\text{CAS}}$)

$\overline{\text{CAS}}$ can be extended indefinitely after $\overline{\text{AREQ}}$ transitions high in non-interleaved mode only, providing that the user program the DP8520A/21A/22A with the $\overline{\text{ECAS0}}$ (negated) during programming. To extend $\overline{\text{CAS}}$, the user continues to assert any or multiple $\overline{\text{ECAS}}$ s after negating $\overline{\text{AREQ}}$. Extending $\overline{\text{CAS}}$ with $\overline{\text{RAS}}$ negated can be used to gain $\overline{\text{RAS}}$ precharge time. By negating $\overline{\text{AREQ}}$, $\overline{\text{RAS}}$ will be negated. The user can then continue to assert a one or both of the $\overline{\text{ECAS}}$ s, which will keep $\overline{\text{CAS}}$ asserted. By keeping $\overline{\text{CAS}}$ asserted with $\overline{\text{RAS}}$ negated, the VRAM will keep the data valid until $\overline{\text{CAS}}$ is negated. Even though $\overline{\text{CAS}}$ will be extended, $\overline{\text{DTACK}}$ output will always end from $\overline{\text{AREQ}}$ negated.

2. Extending $\overline{\text{DT/OE}}$ Functionality

The $\overline{\text{DT/OE}}$ output will follow the $\overline{\text{CAS}}$ output during a VRAM read access, and will remain negated during a VRAM write access. For the DP8520/21/22, the $\overline{\text{DT/OE}}$ output remained negated for all VRAM access cycles. This will allow the VRAM to drive the data bus. There are 2 options for the function of the $\overline{\text{DT/OE}}$ output during a video shift register load operation. With $\overline{\text{ECAS0}}$ negated during programming, the $\overline{\text{DT/OE}}$ output will follow the $\overline{\text{VSRL}}$ input during video shift register load operations. With the $\overline{\text{ECAS0}}$ asserted during programming, $\overline{\text{VSRL}}$ will assert $\overline{\text{DT/OE}}$. $\overline{\text{VSRL}}$ negated before four rising clock edges will cause $\overline{\text{DT/OE}}$ to be negated. $\overline{\text{VSRL}}$ asserted more than four rising clock edges will cause $\overline{\text{DT/OE}}$ to be negated from the fourth rising clock edge.

3. Dual Accessing

$\overline{\text{RAS}}$ will be asserted either one or two clock periods after $\overline{\text{GRANTB}}$ has been asserted. The amount of $\overline{\text{RAS}}$ low and high time, programmed by bits R0 and R1, determines the number of clock periods after $\overline{\text{GRANTB}}$ changes before $\overline{\text{RAS}}$ will start. This is shown in the table below.

R0, R1	$\overline{\text{RAS}}$ Precharge Time	$\overline{\text{RAS}}$ Asserted During Refresh	$\overline{\text{RAS}}$ Asserted from $\overline{\text{GRANTB}}$ Change
0, 0	1T	2T	1 Rising Clock Edge
0, 1	2T	2T	1 Rising Clock Edge
1, 0	2T	3T	2 Rising Clock Edges
1, 1	3T	4T	2 Rising Clock Edges

4. Refresh Clock Counter

The refresh clock counter will count and assert $\overline{\text{RFRQ}}$ externally when it is time to do a refresh. This will occur even when internal refreshes are disabled. This allows the user to run the chip in a request/acknowledge mode for refreshing. $\overline{\text{ECAS0}}$ is used to program the $\overline{\text{RFIP}}$ output to act as either refresh request ($\overline{\text{RFRQ}}$) or $\overline{\text{RFIP}}$. $\overline{\text{ECAS0}}$ asserted during programming causes the $\overline{\text{RFIP}}$ output to function as $\overline{\text{RFIP}}$. $\overline{\text{ECAS0}}$ negated during programming causes the $\overline{\text{RFIP}}$ output to function as $\overline{\text{RFRQ}}$.

5. Clearing the Refresh Clock

The refresh clock counter is cleared by negating $\overline{\text{DISRFSH}}$ and asserting $\overline{\text{RFSH}}$ for at least 500 ns.

17.0 DP8520A/21A/22A User Hints

- All inputs to the DP8520A/21A/22A should be tied high, low or the output of some other device.

Note: One signal is active high. $\overline{\text{COLINC}}$ ($\overline{\text{EXTNDRF}}$) should be tied low to disable.

- Each ground on the DP8520A/21A/22A must be decoupled to the closest on-chip supply (V_{CC}) with 0.1 μF ceramic capacitor. This is necessary because these grounds are kept separate inside the DP8520A/21A/22A. The decoupling capacitors should be placed as close as possible with short leads to the ground and supply pins of the DP8520A/21A/22A.
- The output called "CAP" should have a 0.1 μF capacitor to ground.
- The DP8520A/21A/22A has 20 Ω series damping resistors built into the output drivers of $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, address and $\overline{\text{DT/OE}}$. Space should be provided for external damping resistors on the printed circuit board (or wire-wrap board) because they may be needed. The value of these damping resistors (if needed) will vary depending upon the output, the capacitance of the load, and the characteristics of the trace as well as the routing of the trace. The value of the damping resistor also may vary between the wire-wrap board and the printed circuit board. To determine the value of the series damping resistor it is recommended to use an oscilloscope and look at the furthest VRAM from the DP8520A/21A/22A. The undershoot of $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, $\overline{\text{DT/OE}}$ and the addresses should be kept to less than 0.5V below ground by varying the value of the damping resistor. The damping resistors should be placed as close as possible with short leads to the driver outputs of the DP8520A/21A/22A.
- The circuit board must have a good V_{CC} and ground plane connection. If the board is wire-wrapped, the V_{CC} and ground pins of the DP8520A/21A/22A, the VRAM associated logic and buffer circuitry must be soldered to the V_{CC} and ground planes.
- The traces from the DP8520A/21A/22A to the VRAM should be as short as possible.
- $\overline{\text{ECAS0}}$ should be held low during programming if the user wishes that the DP8520A/21A/22A be compatible with a DP8520/21/22 design.

18.0 Description of a DP8522A/DP8500 System Interface

Several simple block and timing diagrams are inserted to help the user design a system interface between the DP8520A/21A/22A VRAM controller and the Raster Graphics Processor DP8500 (as shown in *Figure 70*). For accessing the VRAM, the DP8520A/21A/22A uses the RGP's PHI 2 clock as an input clock and it runs in Mode 1 (asynchronous mode). This allows the user to guarantee row, column and bank address set up times to a rising clock edge (as shown in timing calculations provided). This system design uses a PAL[®] to interface the access request logic and the wait logic between the DP8522A and the RGP. External logic is also needed for plane control.

18.0 Description of a DP8522A/DP8500 System Interface (Continued)

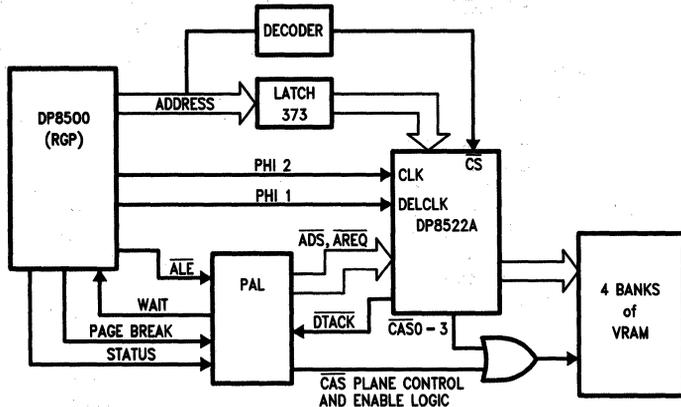


FIGURE 70. DP8422A/DP8500 (RGP) Interface Block Diagram

TL/F/9338-C4

The main idea of the block diagram in *Figure 73* is to cause the video DRAM shift register load operation to happen correctly. Once the DP8500 (RGP) issues the Display Refresh REQuest signal (\overline{DRREQ}) the system knows that the video shift register load operation should occur at a certain defined time later. In the block diagram this time is controlled by the counter device. This counter determines when \overline{VSRL} transitions high, thereby causing the video shift register load operation.

In the upper part of the block diagram is the "REFRESH" output that is used as the "VSRL" input of the DP8522A and is also used to create "REFRESH NOT DONE". To create the "REFRESH" output a NAND latch is used. This latch is set when a screen refresh is in progress, shown by the RGP outputs ALE, B0, and B1 all being high. If the status of the RGP is anything other than screen refresh the latch is reset. The latch is also reset during a screen refresh when the load shift register counter times out. This counter determines when the "VSRL" input of the DP8522A transitions high, causing the video DRAMs to load a row of data into their shift registers.

The "REFRESH" signal along with the load shift register counter output "NOT DONE" are used to create the "REFRESH NOT DONE" signal. This output is used for two purposes. One of which is to hold the "WAIT" output low, thereby inserting WAIT states into the RGP video shift register load access. The other purpose is to hold " $\overline{DT}/\overline{OE}$ " low until " \overline{VSRL} " transitions high.

Important setup timing parameters which must be met for a DP8500(RGP)-DP8522A system (assuming RGP is running at 20 MHz ($T_{CP} = 50$ ns)).

1. Address Setup to \overline{ADS} Asserted

$$= 1 T_{CP} - \#t_{ALV} + \text{Derating the DP8500 Spec for Light Load} - t_{PF373} + \text{Min PAL Delay to Produce ADS \& AREQ}$$

$$= 50 \text{ ns} - 38 \text{ ns} + 5 \text{ ns} - 8 \text{ ns} + 2 \text{ ns}$$

$$= 11 \text{ ns}$$

(Using Light Load Timing Specs, the DP8520A/21A/22A Needs)

$$9 \text{ ns Setup for Row Address to } \overline{ADS} \text{ Asserted}$$

$$11 \text{ ns Setup for Bank Address to } \overline{ADS} \text{ Asserted}$$

2. \overline{ADS} Setup to Clock Rising Edge

$$= 1 T_{CP} - \#t_{ALEV} - \text{Max PAL Delay}$$

$$= 50 \text{ ns} - 26 \text{ ns} - 10 \text{ ns}$$

$$= 14 \text{ ns}$$

(DP8520A/21A/22A Needs 7 ns)

3. \overline{WAIT} Negated Setup to Clock

$$= 1 T_{CP} - \$18 - \text{Max PAL Delay}$$

$$= 50 \text{ ns} - 28 \text{ ns} - 10 \text{ ns}$$

$$= 12 \text{ ns}$$

(The DP8500 Needs 5 ns Setup Time)

Note 1: "\$" symbol refers to a DP8520A/21A/22A timing parameter.

Note 2: "#" symbol refers to a DP8500 timing parameter.

Note 3: ALE asserted by the RGP (DP8500) should use the system PAL to assert WAIT in order to guarantee proper setup time. \overline{DTACK} low should then be used to negate the WAIT signal through PAL equations.

18.0 Description of a DP8522A/DP8500 System Interface (Continued)

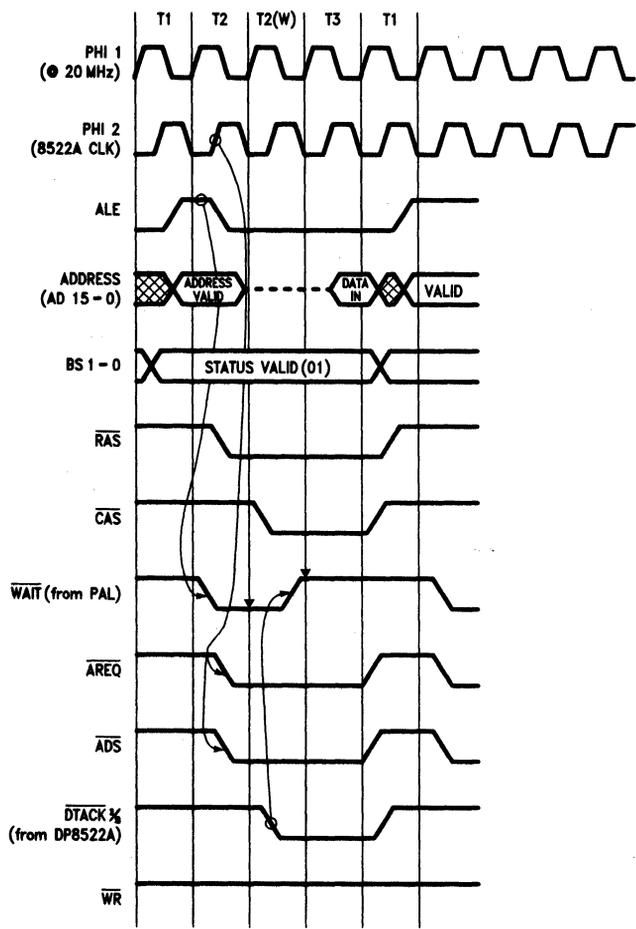


FIGURE 71. DP8522A/DP8500 (RGP) Instruction Read Cycle Timing

TL/F/9338-C5

18.0 Description of a DP8522A/DP8500 System Interface (Continued)

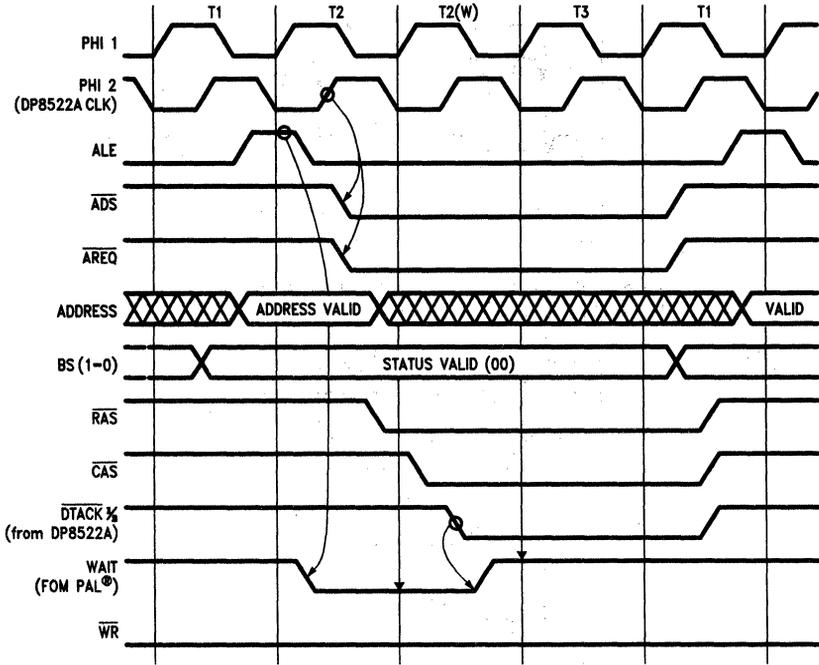


FIGURE 72. DP8500 (20 MHz)/DP8522A Write Operand Cycle Timing

TL/F/9338-C6

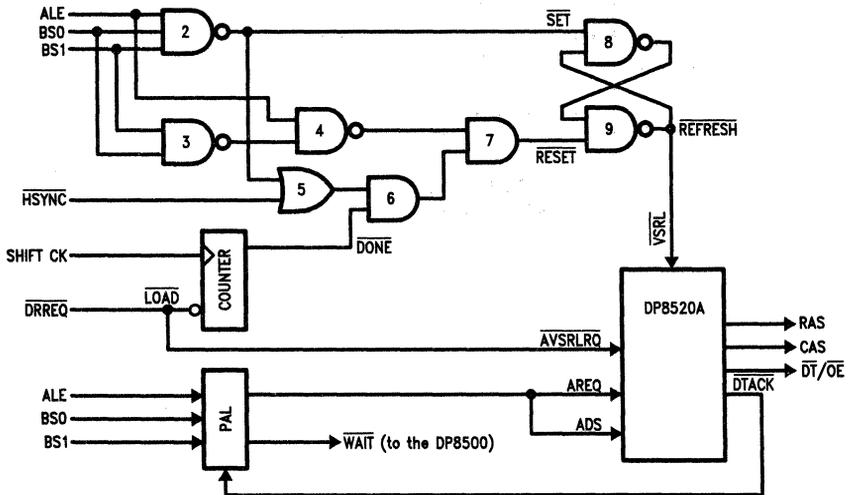


FIGURE 73. Mid-Scan Line Load Application Example

TL/F/9338-C7

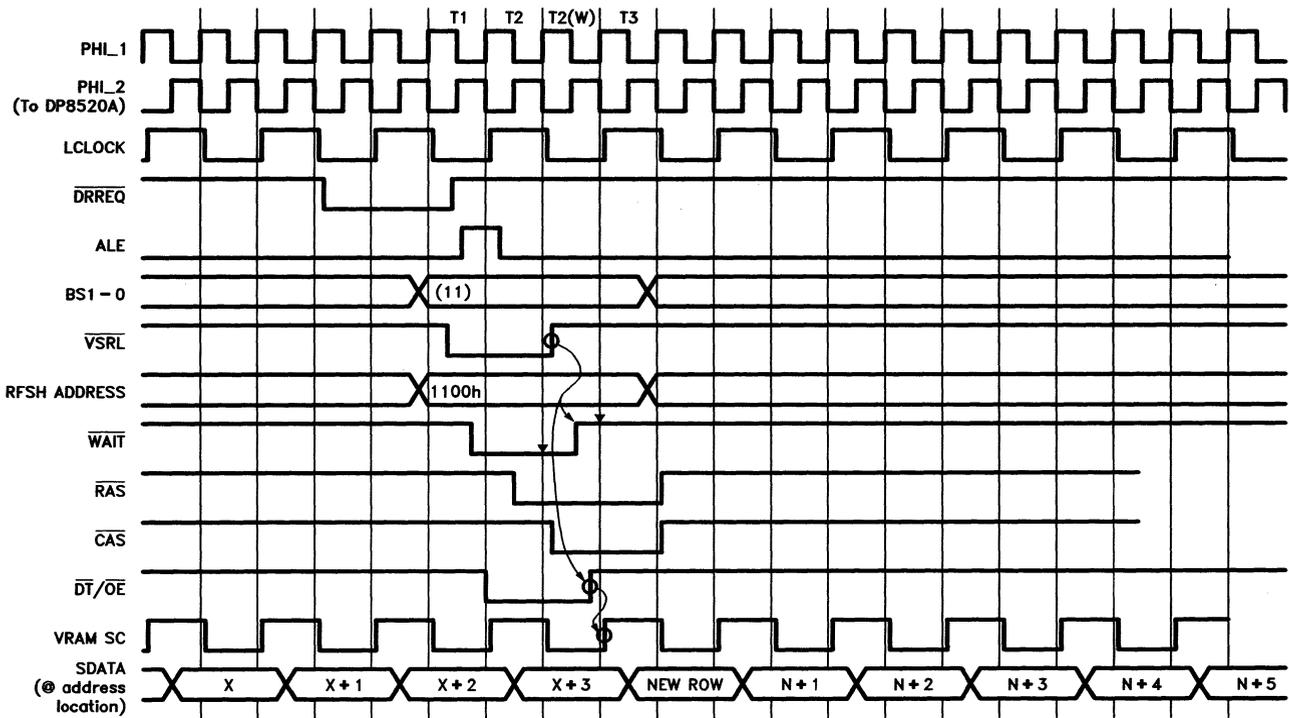


FIGURE 74. Mid-Scan Line Load Functional Timing

TL/F/9338-C6



DP8530 Clock Generator

General Description

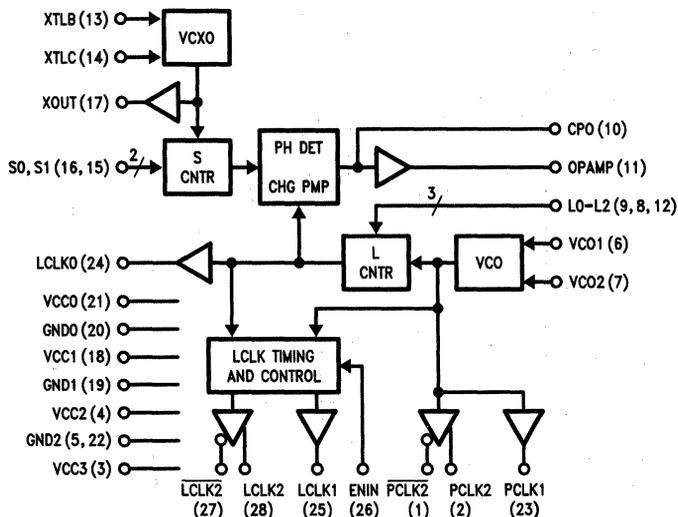
The DP8530 is a clock generator intended for use in medium-performance CRT graphics systems. The device generates both ECL and TTL pixel and load clocks from a single crystal resonator using digital phase locked loop (PLL) techniques. The L counter inputs allow the pixel clock to be divided by 4 to 32 in increments of 4 to drive the LCLK. The S counter inputs allow the system clock (XOUT) to run up to four times the LCLK. Both free-running and gated (by ENIN) LCLK outputs are available.

Features

- On-chip crystal oscillator and phase-locked-loop generate TTL and ECL PCLK and LCLK outputs
- 125 MHz ECL differential output pixel clock (PCLK)
- Gated TTL and ECL load clock (LCLK) outputs
- Pixel clock to load clock divide ratios from 4 to 32 in increments of 4

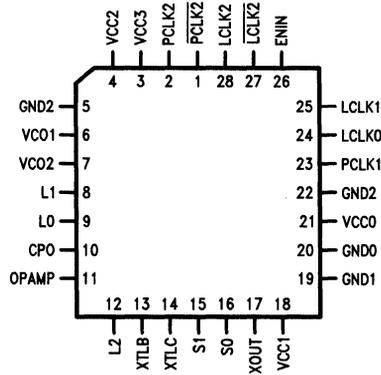
Block Diagram

Numbers in parentheses indicate pin numbers



TL/F/9328-1

Connection Diagram



Order Number DP8530V
See NS Package Number V28A

TL/F/9328-2

Pin Descriptions

1, 2, 23—PCLK2, PCLK2, PCLK1: Differential ECL and TTL compatible pixel clock outputs driven by the VCO (ECL outputs are 10k and 100k ECL compatible).

3—VCC3: ECL output buffer positive power supply.

4—VCC2: ECL internal logic positive power supply.

5, 22—GND2: ECL negative power supply.

6, 7—VCO1, VCO2: External tank circuit connections for the Pierce VCO.

8, 9, 12—L1, L0, L2: A three-bit word input used to select the L Counter modulus. Any modulus from 4 to 32 may be selected in increments of 4. L0 is the least significant bit.

10—CPO: Charge pump output. Used in conjunction with OPAMP to form the external loop filter.

11—OPAMP: Op amp output of the loop. This output is used to control the pixel clock frequency via the varactor diode in the LC tank circuit.

13, 14—XTLB, XTLC: External connections for the Pierce crystal oscillator.

15, 16—S1, S0: Two-bit word that determines the S Counter modulus. S0 is the least significant bit.

17—XOUT: MOS output generated from XTLC, XTLC.

18—VCC1: TTL output buffer supply. Specified for 5V \pm 10% operation.

19—GND1: TTL output buffer supply return.

20—GND0: TTL logic power supply return.

21—VCC0: TTL logic positive power supply. Specified for 5V \pm 10% operation.

24—LCLK0: TTL load clock output. Also is connected to an input of the phase comparator.

25—LCLK1: Gated load clock output.

26—ENIN: TTL compatible video enable input. A high on this input starts LCLK1 and LCLK2 on the next positive transition of LCLK0.

27, 28—LCLK2, LCLK2: ECL compatible load clock.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature -65°C to +150°C
 TTL Signals
 Inputs 7.0V
 Outputs 7.0V

ECL Signals
 Output Current -50 mA
 ESD rating 1000V

Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
VCC0, 1 to GND0, 1	TTL Power Supply	4.5		5.5	V
VCC2, 3 to GND2	ECL Power Supply	Mode 1 (Note 5)	4.5	5.5	V
		Mode 2 (Note 5)	4.2	5.7	
		V _{IH}	High Level Input Voltage, TTL Inputs (Note 2)	2	
V _{IL}	Low Level Input Voltage, TTL Inputs (Note 2)			0.8	V
I _{OH}	High Level Output Current, TTL Outputs (Note 3)			-0.4	mA
	High Level Output Current, MOS Outputs			-2	
I _{OL}	Low Level Output Current, TTL Outputs (Note 3)			8	mA
	Low Level Output Current, MOS Outputs			20	
F _{PCLK}	Pixel Clock Frequency	TTL		80	MHz
		ECL		125	
F _{XTL}	Crystal Frequency			20	MHz
T _{SU1}	Setup Time ENIN to LCLK0 (Note 1)	20	11		ns
T _{H1}	Hold Time LCLK0 to ENIN (Note 1)	0	-9		ns
T _{AMBIENT}	Operating Temp Range	0		70	°C

Note 1: See timing waveforms for relevant signal edges (positive or negative) from which all setup and hold time measurements are made.

Note 2: TTL inputs—ENIN, S0, S1.

Note 3: TTL outputs—XOUT, LCLK0, ENOUT1, 2.

Note 4: Inputs L0, L1, L2 designed to be tied to VCC(2, 3) for high level or tied to GND2 (or left open) for low level. See input schematics.

Note 5: Mode 1: GND2 = 0V;
 Mode 2: VCC2 = 0V.

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units	
V _{IC}	Input Clamp Voltage	VCC(0, 1) = 4.5V, I _{IN} = -18 mA			-1.5	V	
V _{OH}	Output High Voltage	VCC(0, 1) = 4.5V	TTL Outputs	I _{OH} = -400 μA	VCC(0, 1) - 2	V	
			MOS Outputs	I _{OH} = -100 μA	VCC(0, 1) - 0.4		
				I _{OH} = -400 μA	VCC(0, 1) - 2.3		
		VCC(2, 3) = 0V	ECL Outputs	50Ω Load to V _{CC} - 2V	VCC(2, 3) - 1135	VCC(2, 3) - 880	mV
			OPAMP Output	I _{OH} = -1.25 ICPO Sink	VCC(2, 3) - 1.2		V
V _{OL}	Output Low Voltage	VCC(0, 1) = 4.5V	TTL Outputs	I _{OL} = 8 mA		0.5	
			MOS Outputs	I _{OL} = 100 μA		0.3	
				I _{OL} = 20 mA		0.5	
		VCC(2, 3) = 0V	ECL Outputs	50Ω Load to V _{CC} - 2V	VCC(2, 3) - 1995	VCC(2, 3) - 1490	mV
			OPAMP Output	-1.25 ICPO Source		GND2 + 0.5	V
I _I	Max High Level Input Current	VCC(0, 1) = 5.5V GND(0, 1) = 0V	TTL Inputs	V _{IN} = 7V		100 μA	

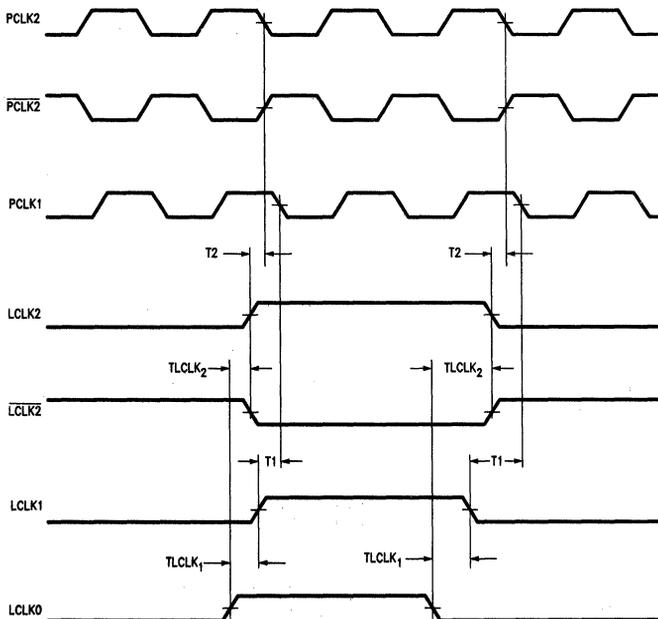
DC Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units	
I_{IH}	High Level Input Current	VCC(0, 1) = 5.5V TTL Inputs GND(0, 1) = 0V $V_{IN} = 2.7V$			20	μA	
I_{IL}	Low Level Input Current	VCC(0, 1) = 5.5V TTL Inputs GND(0, 1) = 0V $V_{IN} = 0.4V$			-200	μA	
I_O	Output Drive Current	VCC(0, 1) = 5.5V TTL Outputs GND(0, 1) = 0V $V_{OUT} = 2.25V$	-30		-110	mA	
I_{CPO}	Charge Pump Current	VCC2 = 0V GND(0, 1) = 0V	Source	-0.2	-0.5	-1.0	mA
			Sink	0.2	0.5	1.0	mA
			TRI-STATE®	-10	0	10	μA
I_{CC}	Supply Current	VCC(0, 1) = 5.5V, GND0, 1 = 0V			22	28	mA
		VCC(2, 3) = 5.5V, GND2 = 0V			95	140	
		VCC(2, 3) = 0V	GND2 = -5.7V 10k ECL range		95	140	
			GND2 = -4.8V 100k ECL Range		85	120	

AC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
F_{MAX}	Max PCLK Freq	ECL	125			MHz
		TTL	80			
F_{MAX}	Max XTL Freq		20			MHz
T1	LCLK1 to PCLK1	CL = 15 pF	4			ns
T2	LCLK2 to PCLK2	CL = 15 pF	2			ns
T_{LCLK1}	LCLK0 to LCLK1	CL = 15 pF	-5	0	5	ns
T_{LCLK2}	LCLK0 to LCLK2	CL = 15 pF	-15	-7	0	ns

Timing Diagram

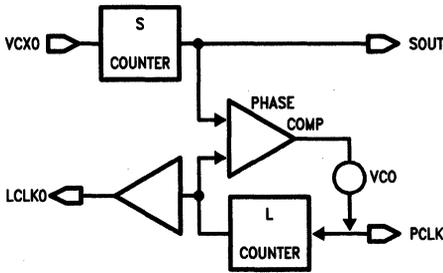


TL/F/9328-3

Circuit Operation

The heart of the DP8530 Video Clock Generator is a crystal oscillator which is used as a frequency reference to generate several clock signals required in a video display system. Among the clocks generated are the ECL load and pixel clocks (LCLK2 and PCLK2) for high speed video shift register parallel load and shift operations, and TTL load clocks (LCLK0 and LCLK1) for moving DRAM data to the video shift registers. The LCLK and PCLK outputs are all internally synchronized in order to simplify system timing.

The LCLK and PCLK outputs are generated using a digital phase locked loop as shown in Figure 1.



TL/F/9328-4

FIGURE 1. PLL Block Diagram

The loop consists of the S and L counters, a phase comparator, and a voltage controlled oscillator (VCO) with the relationship between these elements in the loop defined as:

$$PCLK = \frac{VCXO \times L}{S}$$

where PCLK is the pixel clock frequency, L is the L Counter modulus, and S is the S Counter modulus. When the frequency of the VCO (PCLK) in the phase locked loop is stable the inputs to the phase detector are in phase; thus the S Counter and L Counter outputs are identical in both phase and frequency. The crystal oscillator ensures that the phase and frequency of the S Counter output remain constant. Any drift, or change in frequency, of the VCO will be divided down and appear as a shift in phase at the L Counter out-

put. The phase detector will sense this error and generate a correction voltage for the VCO input which is proportional to the magnitude of the frequency error. This correction voltage will change the VCO frequency to eliminate the error thus keeping the loop locked.

The presence of the S Counter in the loop enables the graphics processor to operate at full speed independent of PCLK frequency. The video shift register's parallel data width determines the L Counter modulus. An 8-bit parallel shift register would use an L Counter modulus of 8 so that a parallel load pulse occurs once every 8 pixel clocks. The L Counter output is used to derive the three LCLK outputs.

The ECL LCLK2 output is used in conjunction with the PCLK2 output to load data into a high-speed video shift register. The PCLK2 provides the clock and LCLK2 provides the load signal for the shift register.

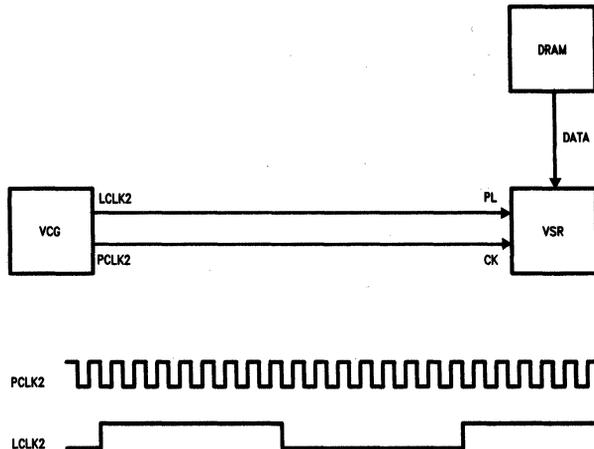
The other two LCLK outputs (LCLK0, LCLK1) are TTL outputs. They can be used to control a selection of different DRAMs/video shift register configurations as shown in the typical system architecture section which follows.

The S Counter can be programmed to divide by any integer up to 4 and the L Counter can be programmed for any word width from 4 bits to 32 bits in increments of 4. Table I shows some of the frequencies possible using various values for the S-modulus with an L-modulus of 8.

TABLE I. Partial Table of PCLK Frequencies

S MOD	8-Bit Word (L = 8)
	PCLK Frequency
	XTL = 10 MHz
1	80.0 MHz
2	40.0 MHz
3	26.7 MHz
4	20.0 MHz

Figure 2 demonstrates the configuration and the resultant waveforms with a DP8530 driving a video shift register and a DRAM.



TL/F/9328-5

FIGURE 2. System Configuration and LCLK Waveforms Using DRAM

Loop Filter Calculations

Several constraints need to be known in order to determine the loop filter components. They are the loop divide ratio (N), the phase detector gain (K_p), the VCO gain (K_O), the loop bandwidth (W₀), and the phase margin (φ).

The constant K_p is fixed at 80 μA/rad for the DP8530. N is the L counter modulus for the loop. A 60° phase margin is recommended, however the equations allow other values to be used if desired.

The oscillator gain constant K_O can be obtained from Table II or obtained experimentally. This is done by driving R2, the resistor which normally connects the varactor to the OPAMP OUTPUT, with an external power supply. Set the supply to GND2 + 3V and note the PCLK frequency. Next, set the supply to GND2 + 2V and note the frequency again. The difference in these two frequencies (times 2π to convert to radians) is K_O. For optimum performance the desired PCLK frequency should be somewhere between the two frequencies measured above. This may require adjustment of the coil.

Before choosing a value for W₀ one fact should be pointed out: R2 and C3 (the coupling capacitor between the coil and the varactor) form a low pass filter. Thus the loop bandwidth must be chosen to be less than the cutoff frequency of this filter. We recommend a value of 100 Hz to 3 KHz (times 2π) for W₀.

Having determined all these constants the following equations are used to find the component values:

- R1 = (1.08 N W₀)/(K_p K_O)
- C1 = (3.46 K_p K_O)/(N W₀²)
- C2 = (0.27 K_p K_O)/(N W₀²)

To use a phase margin of other than 60° use the following:

- R1 = (N W₀/2 K_p K_O) (cosec θ + 1)
- C1 = (2 K_p K_O/N W₀²) (tan θ)
- C2 = (K_p K_O/N W₀²) (sec θ - tan θ)

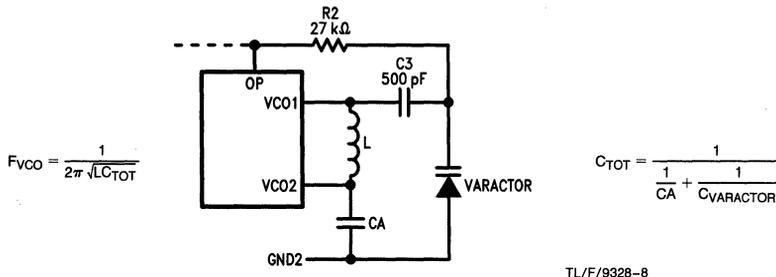
For example: design a system with the following characteristics:

- 1560 pixels per line (2000 pixels including retrace)
- 8 bit wide video data
- 10 MHz processor rate

The PCLK frequency will be 10 MHz × 8 bits = 80 MHz. The components in Table II will be used. Note that K_O is 19 Mrad/Volt. Because it is an 8-bit wide system the L counter modulus must be 8. By choosing W₀ to be 2800 Hz (times 2π) the equations give:

- R1 = 100Ω
- C1 = 2.0 μF
- C2 = 0.17 μF (0.2 μF may be used)

TABLE II
Recommended VCO Components



TL/F/9328-8

Frequency (MHz)	L μH	Toko Part #	CA pF	CVARACTOR pF	Motorola #	K _O Mrad/volt
60	0.258	E502HNS-6000026	56	30	MV209	16
80	0.17	E502HNS-4000024	39	30	MV209	19
100	0.12	E502HNS-3000023	30	30	MV209	21
120	0.07	E502HNS-2000022	39	30	MV209	31

Input Schematics

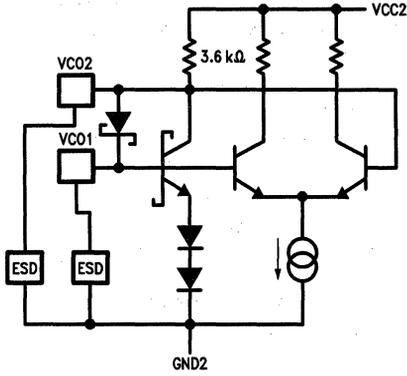


FIGURE 3. VCO Inputs

TL/F/9328-9

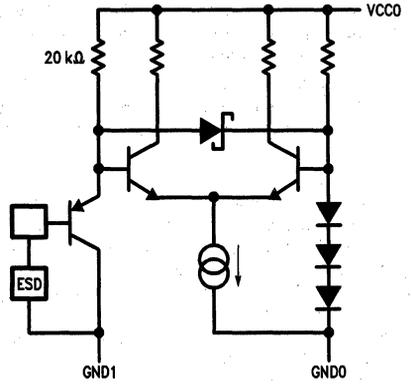


FIGURE 4. TTL Inputs

TL/F/9328-10

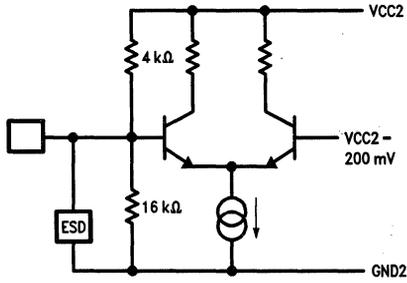


FIGURE 5. L Inputs

TL/F/9328-11

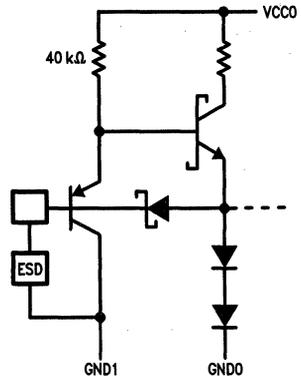


FIGURE 6. S Inputs

TL/F/9328-12

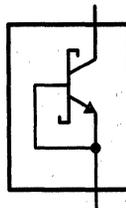


FIGURE 7. Typical ESD Circuit

TL/F/9328-13

Output Schematics

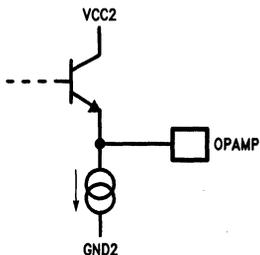


FIGURE 8. Op Amp Output

TL/F/9328-14

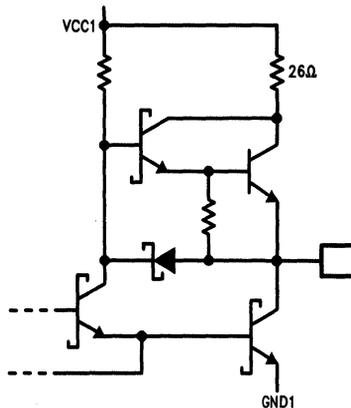


FIGURE 9. TTL Outputs

TL/F/9328-15

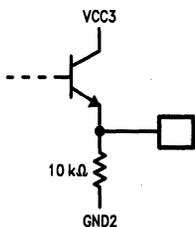


FIGURE 10. ECL Outputs

TL/F/9328-16

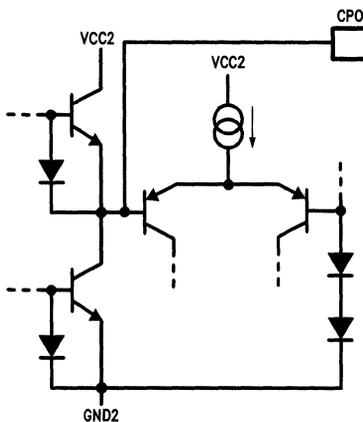
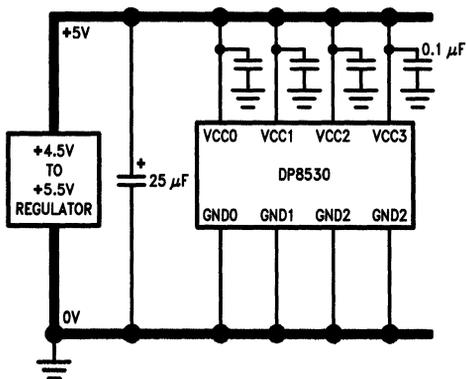


FIGURE 11. Charge Pump Output/Op Amp Input

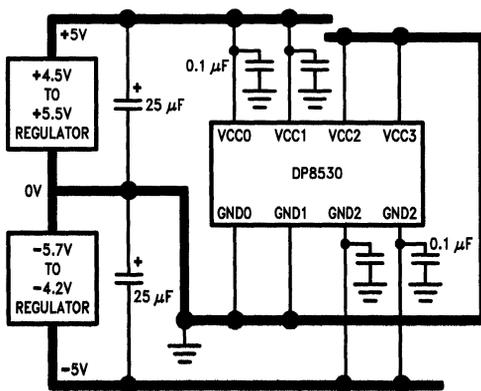
TL/F/9328-17

Single Power Supply Operation



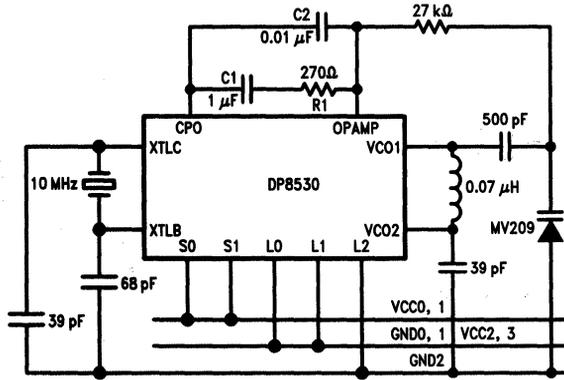
TL/F/9328-6

Dual Power Supply Operation



TL/F/9328-7

Typical Application



TL/F/9328-18

PCLK = 80 MHz
 L = +16
 Split Supply Mode

TABLE II. Counter Modulus Tables

S MOD	S Counter Inputs	
	S1	S0
1	L	L
2	L	H
3	H	L
4	H	H

L MOD	L Counter Inputs		
	L2	L1	L0
4	0	0	0
8	0	0	1
12	0	1	0
16	0	1	1
20	1	0	0
24	1	0	1
28	1	1	0
32	1	1	1

L = TTL Logic Zero (GND0, 1)
 H = TTL Logic One (VTTL0, 1)

0 = GND2
 1 = VCC(2, 3)



Section 2
DP8500 Development Tools



Section 2 Contents

DP850EB Raster Graphics Processor (RGP) 4-Plane Evaluation System	2-3
DP850DB8 Raster Graphics Processor (RGP) 8-Plane Development System	2-4
DP8500 Raster Graphics Processor Software Tools	2-5

DP850EB Raster Graphics Processor Four Plane Evaluation System



TL/F/10431-1

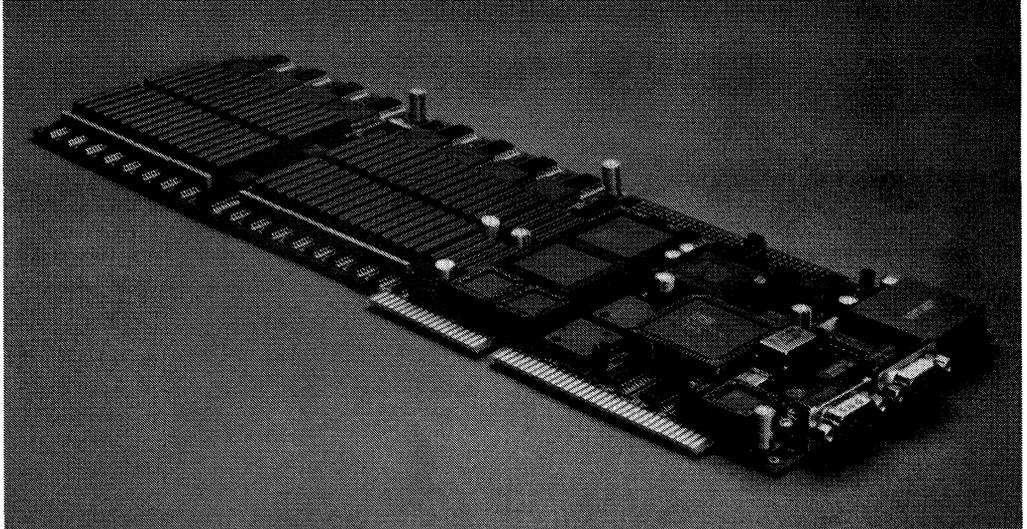
- 640 x 480 resolution stand-alone PC board for easy evaluation and a large breadboard area for development
- DP8500 Raster Graphics Processor featured in a basic 20 MHz four-plane architecture. On-board connectors for the power supply, multisync monitor, and 25-pin serial port cable
- Board design incorporates 16V8 GALs for reduced power consumption
- DOS-based software includes an absolute assembler, symbolic debugger, a software simulator, and sample source code programs
- Technical documentation includes schematics and programmable logic equations
- Host platform is a DOS AT/XT with an RS232 serial port and a multisync monitor

Product Overview

The DP850EB is designed to provide a simple, efficient means of evaluating the Advanced Graphics Chip Set architecture. Provided with complete technical documentation, its own power supply, and software tools, the board connects to the host PC com-

puter by means of an RS232 interface. This interface allows programs to be downloaded to the graphics board where they can also be debugged with the full-featured serial-interface debugger.

DP850DB8 Raster Graphics Processor Development System



TL/F/10432-1

- Utilizes National's 20 MHz AGCS architecture
- Supports video rates up to 125 MHz
- Supports 256 colors out of a palette of 16 million
- 256 kb of program and font memory
- 128 kb of socketed "OTP EPROM" memory for resident firmware
- 2 mb of frame buffer memory
- On-board RS232 serial port
- Complete technical documentation, including schematics and programmable logic equations
- Third party software support (ordered separately) for DGIS, GKS, CGI, X-Windows, and Microsoft Windows

General Description

The DP850DB8 is an 8-plane AT-bus add-in graphics card suitable for supporting extensive software development projects based on the AGCS architecture.

DP8500

Raster Graphics Processor Software Tools

A complete line of development and applications support software is provided for the DP8500 Raster Graphics Processor. The standard release for all of these products is on 5 1/4" floppy diskettes for hard-disk DOS systems containing 640 kBytes of memory. Certain UNIX® environments are also supported through special order, in particular GENIX™ V.3, DECT™ Ultrix™ V2.0-1, and Sun UNIX 4.2 Release 3.2. Special source code licenses are available for customers who desire to port the software to non-supported environments.

DP85SW1001M5: RGPT™ Software Development Package

(DP85SW1001SP: UNIX or source code special orders)

The Software Development Package Contains:

- Macro Assembler
- Software Simulator
- Linker
- Librarian
- Object-to-HEX file converter

DP85SW1002M5: RGP C Compiler

(DP85SW1002SP: UNIX or source code special orders)

(Note: The RGP C Compiler generates Macro Assembler source code as its output. The RGP Software Development Package is required in addition to the C compiler for a complete C development environment.)

DP85SW1003M5: RGP Symbolic Debugger

(DP85SW1003SP: UNIX or source code special orders)

For the DP850EB or DP850DB8 boards. Configurable to other hardware environments.

DP85SW2002M5: RGP Graphics Kernel

The RGP Graphics Kernel provides the complete source code for the implementation of a real-time, multi-tasking, device-independent graphics interface for the RGP.

(Note: The Graphics Kernel is written in RGP Macro Assembler source code. The RGP Software Development Package is required to assemble the code into executable format.)

DP85SW2003M5: RGP Software Utilities

The RGP Software Utilities provide source code examples of many common graphics-related functions such as:

- RGP initialization
- Mathematical operations
- Generation of curves
- Sub-dividing complex polygons into convex polygons.

(Note: The Software Utilities are written in Macro Assembler source code. The RGP Software Development Package is required to assemble the code into executable format.)

Third-Party Software Support:

Extensive third-party software support is available for the RGP, including DGIS, GCI, GKS, X-Windows, and Microsoft Windows. Contact your local sales representative for detailed information.



Section 3
Application Notes



Section 3 Contents

AN-451 An Architectural Solution for High Performance Graphics	3-3
AN-547 Interfacing the DP8500 Raster Graphics Processor	3-11
AN-391 The LM1823 A High Quality TV Video IF Amplifier and Synchronous Detector for Cable Receivers	3-26
AN-402 LM2889 R.F. Modulator	3-42
AN-580 A 16-Bit Video Shift Register with On-Board FIFO Operates at Rates Up to 350 Million Pixels per Second	3-53
AN-553 Mid-Scan-Line Load Techniques Using the DP8500 Raster Graphics Processor	3-60
AN-554 Accurate Timing for Multi-Board Graphics Systems	3-65
AN-604 DP8512, DP8513, DP8514 Video Clock Generator Evaluation Board	3-69
AN-609 A Graphics Acceleration Card for the AT Using the Advanced Graphics Chip Set	3-82

An Architectural Solution For High Performance Graphics

National Semiconductor
Application Note 451
Charles Carinalli



INTRODUCTION

Computer graphics is one of today's fastest growing market segments. The fundamental reason for this dramatic growth relates directly to the productivity improvement that a graphic representation can give to nearly all problems in the engineering, scientific, business, and consumer market segments. Nothing does more to ease the "man to computer" interface data interpretation and presentation problems than a graphics display and efficient application software.

Penetration of computer graphics into all market segments has grown especially fast in the past few years since it has become more affordable. This is due to the decreasing costs of high quality CRT displays and significant reductions in the cost of Dynamic RAM. In the near future, high performance graphics support hardware will be more available in the form of a number of VLSI integrated circuits, further decreasing the costs of graphics hardware.

Now that more people have a good understanding of the importance of computer graphics to the productivity of their job, the demand for higher resolution, faster graphics will increase significantly. This will be especially true in the engineering, scientific, and business communities. The current challenge for this generation of graphics hardware is clearly in the area of affordable, high resolution, fast computer graphics. Attaining these goals will not only do much to expand the useage of computer graphics but also expand application areas.

GRAPHICS ARCHITECTURE

It is not the purpose of this paper to discuss the varied aspects of software and hardware computer graphics architectures. What will be addressed are the problems of hardware architectures in the area of list driven display generation, manipulation, and screen refresh. Current monochrome graphics hardware architectures generally do not satisfy the current demands of high performance computer graphics which include display resolutions of 1000 pixels by 1000 pixels and above, very high speed screen manipulations, and color.

To reinforce the growth of computer graphics, hardware architectures must be developed which will support the rapid growth in display resolution which will occur in the next few

years. The architectures must allow a high degree of application independence yet not require significant changes in the overhead software interface. The hardware architecture must be expandable, again avoiding significant software changes. Finally, the architecture must be implemented in VLSI for reasons of cost and speed.

This paper will describe a graphics hardware architecture that meets these criteria.

KEY PERFORMANCE ISSUES

Figure 1 is a block representation of the logical hardware and software partitioning associated with a typical graphics system. The left block represents the application interface and image creation segments. Data must be translated from the database found on disk (or received via communications networks) to a format which will be meaningful in the CRT display. At the application level this may involve database interpretation and translation to and from graphics standards such as GKS and CORE. But it also involves the final translation from what is described as the "world space" representation to the "normalized" coordinate space associated with the hardware of the particular graphics system in use. This is done with the Main CPU in software or through hardware acceleration via usage of a number of VLSI integrated circuits on the market.

Once this translation is completed the transfer to the actual "display" coordinates is usually accomplished through an instruction list given to a graphics processor linked directly to the CRT frame buffer. It is the function of this graphics processor to offload the main CPU in the creation and manipulation of the massive amounts of data in the actual bit map in the frame buffer. Without this block, any large bit map manipulation would cripple the main CPU in its main application chores.

The final blocks represent the portion of the graphics system which handles the lower level image movements and manipulations as well as the screen refresh function. High speed hardware implementations of these blocks with an architecture that links closely with the graphics processor is the only way performance can be maintained as display resolution increases along with the increasing number of memory planes for color.

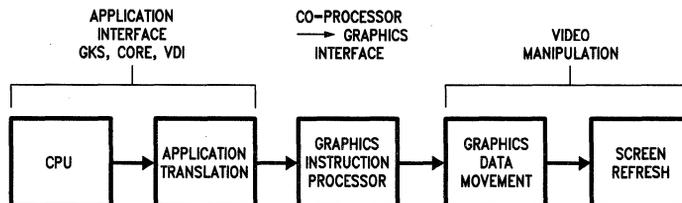


FIGURE 1

TL/F/8769-1

We will focus on the performance issues of the last two major blocks of *Figure 1* and factors that will influence.

- List Driven Display Generation
- Graphic Data Manipulation and Movement
- Screen Refresh

The technical problems of providing cost effective architectural solutions with present hardware (given the current and future performance requirements) make these functions ideal candidates for VLSI integrated circuits.

**THE GRAPHICS FRAME BUFFER—
SCREEN REFRESH ARCHITECTURE**

To focus on these performance issues requires a good understanding of the tradeoffs associated with the graphics frame buffer. *Figure 2* is a simplified diagram of a typical graphics workstation. The graphics controller is shown closely linked to the memory associated with the CRT screen display (called the CRT refresh memory or the frame buffer) and the main CPU with its associated graphics ge-

ometry processing hardware and software. This figure shows the popular configuration in which a special processor is allocated to handle graphics functionality thus off loading the main CPU with respect to fundamental graphics operations. Those operations are list driven based on a lower level graphics software language. We will talk later about advantages of this and some specific implementations.

The image that is stored in the frame buffer is actually stored conceptually in three dimensions as shown by the cube represented in *Figure 3*. Each pixel on the screen can be mapped to multiple pixels (in the case of color) contained within this cube. Access to the data stored in this cube is necessary for both screen refresh, update, and data manipulation. Ideally, the amount of time available for manipulation should greatly exceed the time required for screen refresh, thus enabling very quick display update and image movement.

The fundamental graphics tradeoffs involve the need to maximize CPU or graphics controller access to the frame buffer while maintaining regular (required) CRT refresh. As display resolution increases and color becomes more important these issues become significant factors in the design and cost of the graphics sub-system.

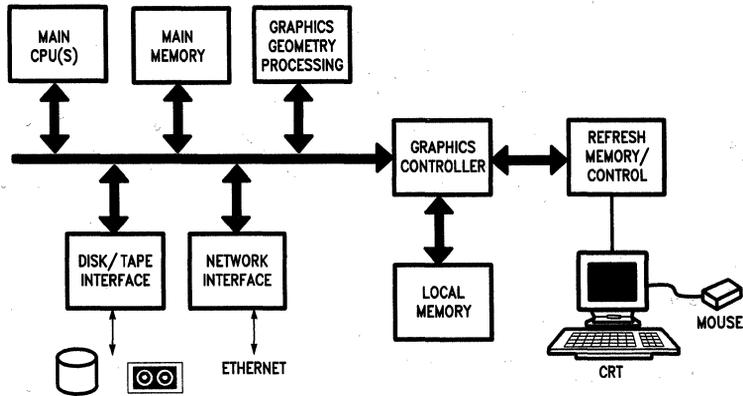


FIGURE 2. Typical Graphics Workstation

TL/F/8769-2

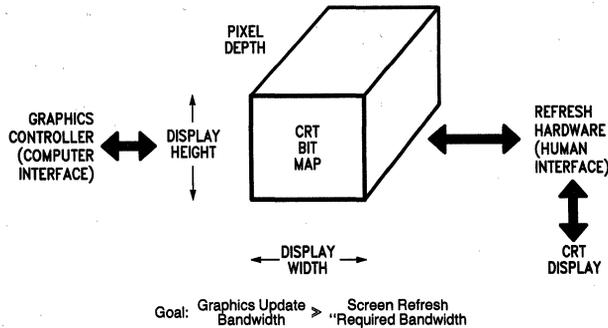


FIGURE 3. Graphics Frame Buffer

TL/F/8769-3

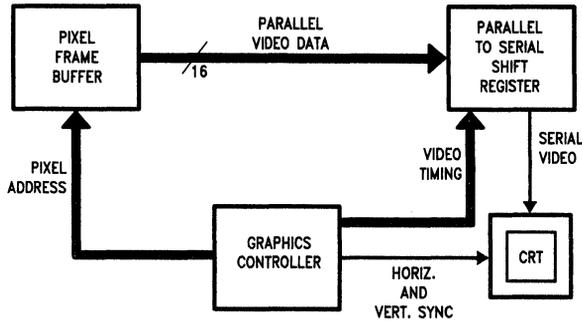


FIGURE 4. The Graphics Video Loop

TL/F/8769-4

Current video rates for medium resolution graphics systems easily exceed 50 MHz. High resolution systems attain video rates on the order of 100 to 125 MHz with clear trends to rates in excess of 125 MHz in the near future. Given these clock rates, screen refresh has become a significant problem in the design of the graphics sub-system.

The high video rates have forced a graphics memory organization for refresh as shown in Figure 4. Since the size of the frame buffer is large, the desired memory component because of cost and space is the DRAM. However, the DRAM does not have access speeds suitable for current high resolution displays. As a result, the screen refresh process is implemented through a parallel to serial conversion in the "video loop". To refresh the screen, the graphics controller presents a word address to the frame buffer, the resulting data word (usually 16 pixels) is then converted into a serial video stream via an external (usually TTL to ECL) shift register under the timing and control of the graphics refresh hardware. Figure 4 shows the implementation for a single plane system, multiple plane systems would require the same number of 16 Bit refresh pixel words and parallel to serial video shift registers as the number of planes.

If the video rate is 100 MHz, this parallel to serial buffering reduces the need for a parallel word from the frame buffer to a clock rate of 6.25 MHz if the word is 16 bits wide. This is still an access time of 160 ns. Given this, it is clear that with conventional time multiplexed DRAM design, the only time left for data manipulation and screen update will be during horizontal and vertical blanking when screen refresh is disabled. With the demands of current graphics end applications, this is clearly not enough!

A number of frame buffer architectures have been developed to get around this problem. These include double frame buffering, dual porting, and making maximum use of page mode access. The most popular solution to this problem is the one provided via a new type of DRAM called the Video DRAM shown in Figure 5. The video DRAM is a con-

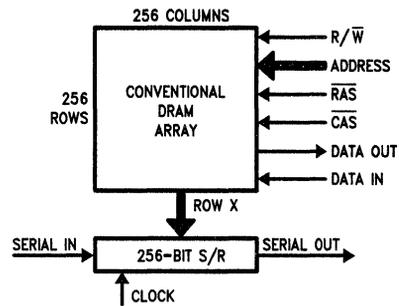


FIGURE 5. The Video DRAM

TL/F/8769-5

ventional DRAM with a 256 bit shift register on board. This shift register is connected such that a complete row of memory cells can be loaded into the shift register at one time. The shift register has its own output which can be controlled with separate input lines to the video DRAM. If you configure the Video DRAM as shown in Figure 6, you can see we have implemented a dual ported memory. The left side is configured as a typical DRAM interface. The right side is configured for screen refresh with the parallel word being generated from the shift registers of the Video DRAM (this parallel configuration is still needed since the Video DRAM shift registers are not able to clock at the full video rate in high performance systems).

The advantage that this configuration provides is nearly full time access for screen update and manipulation since only one load to the shift register is generally needed per scan line. While the pre-loaded video data is clocked out of the shift register, random access can occur in the frame buffer by either the graphics controller or the main CPU.

Many configurations of the Video DRAM are becoming available with multiple vendors committing to the architecture already in place. It will clearly become the standard component for building graphics frame buffers. Any new graphics hardware architecture must be designed to optimize the use of this type of memory component.

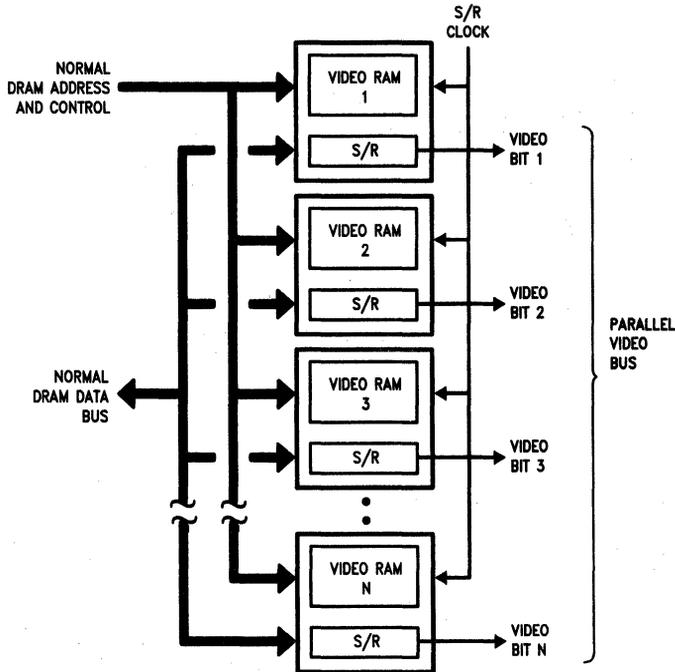


FIGURE 6. Multiple Video DRAMs

TL/F/8769-6

**THE GRAPHICS FRAME BUFFER—
UPDATE/MANIPULATION ARCHITECTURE**

We have discussed the preferred frame buffer architecture with respect to screen refresh. Now we will evaluate the tradeoffs associated with the update and manipulation side. The performance issues associated with the graphics processor or main CPU update or manipulation of the image in the frame buffer have a major influence on the data and address configuration of the frame buffer. There are generally three such configurations—pixel, plane, and mixed.

The pixel architecture is shown in Figure 7. This architecture is best described as one in which the frame buffer data is manipulated one pixel at a time. For multiple planes, the address to the frame buffer generates a data word which is composed of pixels at the same location across multiple planes—thus the term "pixel depth". This is the architecture

often found in image processing and solids modeling applications where the value of each pixel is very computation intensive due to color value or shading variations. These applications typically require 16 to 32 memory planes.

In the plane architecture, Figure 8, the frame buffer data is manipulated one word (usually 16 bits) at a time within each plane. To change one bit, 15 other bits must be carried along. Also, since the word boundary of the 16 bits exists, a barrel shifter is needed if image placement and movement accuracy is needed down to the actual pixel level. Despite these disadvantages, in the engineering and business application area, the plane architecture is the most popular since these applications are less pixel computational intensive but more data creation and image movement intensive. This architecture is lower cost and brings with it higher performance when large bit maps must be manipulated.

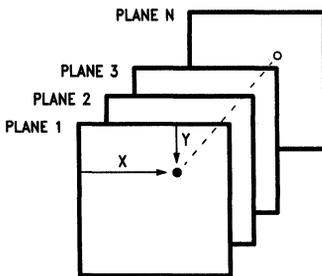


FIGURE 7. Pixel Architecture

TL/F/8769-7

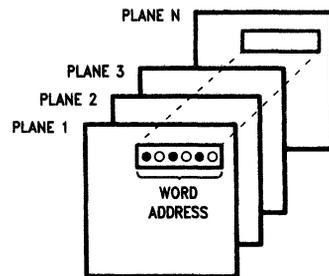


FIGURE 8. Plane Architecture

TL/F/8769-8

Where the application has need for both types of architectures, the mixed architecture shown in *Figure 9* is implemented. Here, access to the frame buffer can be either at word width or pixel depth, thus providing the best of both worlds. In the past, this architecture has been implemented only in the more expensive of workstations due to the overhead hardware costs. But, the applications of workstations that need high speed update along with computation intensive displays is increasing. The time has come for a graphics hardware architecture that can efficiently merge these two architectures.

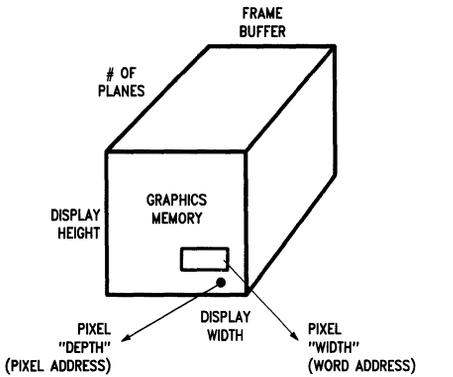


FIGURE 9. Plane and Pixel Architecture

WORD BOUNDARIES AND THE BARREL SHIFTER

In the engineering and business community, the most popular frame buffer architecture is the plane architecture. But as the quality of these displays increase, restrictions associated with the plane architecture word boundary constraints have limited the flexibility and performance of these systems. To solve these problems, some of the designs have switched to the pixel architecture. With the "correct" architectural solution this switch would have not been necessary.

Figure 10 demonstrates the word boundary problem via a 4 bit word example. This figure demonstrates a "word boundary aligned" translation. The relative pixel locations of the 4 bit words are the same at the destination image as they are in the source image. Thus the pixels maintain the same alignment to the word boundary. This is a simple transfer where manipulation of the pixel positions in the source word need not be modified to transfer it to the destination.

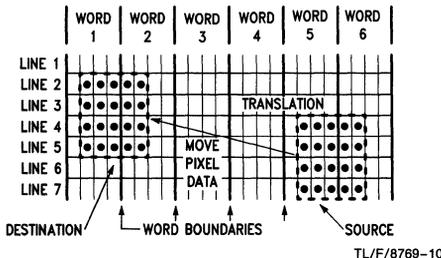


FIGURE 10. Plane Oriented Translation (Word Boundary Aligned)

In *Figure 11*, the source word pixel alignment does not agree with the desired pixel destination alignment. In this transfer, the source pixel map is shifted one pixel to the right in the transfer. This type of control is not available if the frame buffer is addressed by words. If the frame buffer is addressed by pixel, then the price paid for overhead access to do this simple manipulation is significant. A better solution to this problem is to employ an additional hardware function called a barrel shifter and maintain word boundary addressing.

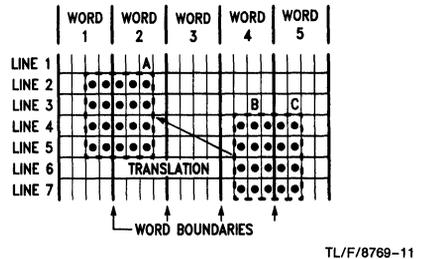


FIGURE 11. Plane Oriented Translation (Non-Word Boundary Aligned)

A simplified diagram of a barrel shifter is shown in *Figure 12*. In order to maintain consistency across word boundaries on a shift like the example in *Figure 11*, two source words must be read to create a new shifted destination word. The barrel shifter solves the fundamental restrictions of the plane architecture with respect to exact pixel manipulations. But, if these manipulations occur within the main CPU or for that matter in the main graphics processor a new performance bottle neck may result in multiple plane color systems. The ultimate performance of such a system is associated with the speed of the barrel shifter and its location. Ideally, for maximum performance, you would want a barrel shifter for each plane.

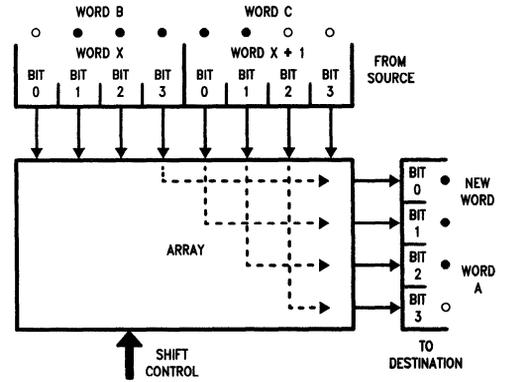


FIGURE 12. The Barrel Shifter

BITBLT

Due to the growth of display resolution and the resulting demand for movement of massive amounts of data, the need for a lower level graphics operator that well describes these data movements and manipulations has increased. The most popular operator or function is BITBLT (for BIT Boundary BLock Transfer) originally developed at Xerox PARC (Palo Alto Research Center). This function is also called Raster Op (short for Raster Operator).

Fundamentally, BITBLT is a logical operator which simply describes the manipulation of rectangular bit maps of any size. For this reason it should be viewed as a lower level language for graphics bit maps. Figure 13 is a conceptual drawing of how BITBLT works.

The block on the right is a source bit map area which can be outside or within the view area of the frame buffer. On the left is the destination area generally, though not necessarily, within the view area of the frame buffer. If the graphics hardware is implemented with BITBLT in mind, a single simple setup from the main CPU (via an instruction list) can cause the graphics controller to move massive amounts of data. The only information needed is the absolute pixel address of the source and its width and height in pixels, the destination pixel address, and the clipping rectangle pixel address with its associated pixel width and height.

Resulting from this single setup, any size bit map can be moved without intervention from the main CPU. Additionally, if a barrel shifter is integrated within this operation, the BITBLT is not bounded by word boundary constraints. Finally, if multiple BITBLT processors and barrel shifters are used, for example one set per plane, transfers within all planes can be done in parallel.

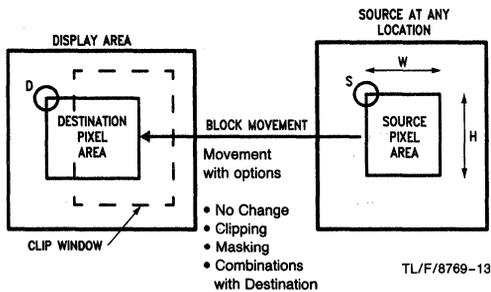


FIGURE 13

If interaction is desired from the source image with the destination area, logical operations may be implemented within the BITBLT operation, such as AND, OR, and XOR. To do this, the source data is first read and barrel shifted to align with the destination word. The destination word is then read and the desired operation performed, creating a new destination word. This word is then transferred to the destination. This is described by the following equation—

$$\text{SOURCE OP DESTINATION} \rightarrow \text{DESTINATION}$$

Where OP can be a logical operator such as AND, OR, XOR, etc.

BITBLT is particularly well suited for applications where there is a high mixture of graphics, text, and windowing. Pop-up menu's, icons, accelerated filling, fat line drawing, and high speed text transfer are all functions where BITBLT provides accelerated solutions.

Clearly, for any architecture to be a good match to high resolution displays and fast screen update, the main CPU should not be involved in the simplest of image movement chores. The main advantage of BITBLT functionality comes in off loading the main CPU with the detail of the bit map operations and at the same time adding a powerful functional operator.

PREVIOUS SOLUTIONS

Historically, graphics VLSI controllers have not well addressed the high resolution color applications of modern bit mapped graphics terminals. As a consequence, much of the graphics hardware was implemented with bipolar bit slice processors and random logic, much of which was Schottky TTL or ECL. The result was a costly high power solution, difficult to map into future high resolution multiple plane applications.

A new generation of graphics VLSI integrated circuits will be introduced during the next year, from a variety of semiconductor vendors. Many of these I.C.'s have been developed to address the performance problems associated with high performance bit mapped displays. In nearly all cases, to make these controllers effective, a graphics hardware architecture had to be selected. In most cases this has restricted the performance and applicability of these devices across the spectrum of graphics applications.

Typically, most of these architectures suffer from inflexibility in the processor to memory plane interface. Figure 14 is an example of this problem. Some of these architectures are designed to directly support 1, 4, or 8 planes of memory. Transition to more planes requires additional processors. The result is an associated cost increase in hardware usually coupled with a degradation of performance due to synchronization problems between processors.

Although performance may be good within the fundamental bounds of a single processor, the rules significantly change when the transition is made to multiple processors. Nearly without exception, this new generation of VLSI components has the main controller or processor intimately involved with both frame buffer addressing and data manipulation. This is where the main limitation exists; when memory plane expansion occurs, the graphics processor again becomes the bottle neck to graphics system performance. It matters little whether this is a BITBLT based architecture with internal

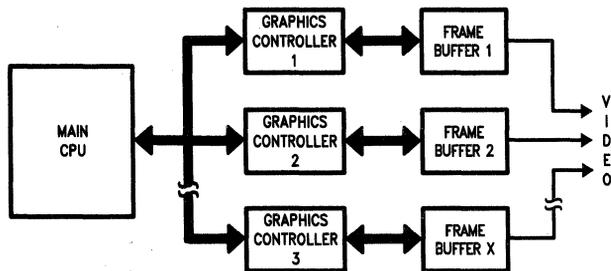


FIGURE 14. Current Graphics Chips Color Application

barrel shifters, the common data and address graphics function is the limitation to expansion and performance.

Many of these new chips begin to address the notion of parallelism. Operations can occur in different memory planes at the same time. However, due to pin and architecture limitations, expansion to more planes than supported by a single processor significantly degrades performance.

OPTIMUM ARCHITECTURE

The only practical way to solve these problems and still adequately address all the performance issues we have discussed here is to implement an architecture as shown in *Figure 15*.

In this diagram a single graphics processor assumes the responsibility of address and timing associated with the graphics frame buffer while maintaining the classical address and data interface with the main CPU and processing functionality with its local program storage memory.

With such a processor, based on a BITBLT architecture, frame buffer address operations are its responsibility while actual data manipulation is the responsibility of the slave BITBLT data manipulation functions.

The Slave Controller (data manipulator) is a data handling chip which receives all control from the single main graphics processor. It is responsible for masking, barrel shifting, and BITBLT operations associated with its own memory plane.

A separate control bus from the graphics processor passes all control and setup information to the slave manipulators in parallel with other control information via the data bus. Once this initial information is set-up, the main graphics processor is no longer involved in graphics data manipulation while the graphics function is implemented. The slave manipulators can be configured via the control and data bus for the exact destination left and right masking, BITBLT operation, and amount of barrel shift.

Additionally, operations can occur in parallel with all slave manipulators working within their own plane. When plane to plane transfers are required, one slave manipulator acts as the source and any number or combination of slave manipulators act as the destination. Here again, any BITBLT operation, either within a plane or plane to plane is fully set-up via a single graphics processor independent of the number of planes used.

Most importantly, performance is independent of the number of planes. Operations for 32 planes of BITBLT take the

same short time as if only one plane were used since the architecture is the same. Plane to plane transfers have the same speed whether you are transferring the image to one plane or 16 planes!

In summary, in this architecture, the main graphics processor is concerned with graphics setup and addressing to the frame buffer during the graphics operation. With a BITBLT architecture, large data movements, character/text transfer, and line drawing and filling all exhibit high performance. Since the processor function occurs only once in the system, the higher cost associated with such processors is a constant and independent of the number of planes.

The more cost effective Slave manipulators are concerned with the actual data manipulation via control from the main graphics processor. They are local to each memory plane frame buffer and fully synchronized by the main processor for operations within the plane or for plane to plane transfers.

This architecture provides one consistent hardware interface independent of the number of memory planes utilized. The hardware interface is the same for one plane or 32. This lends itself to a level of software consistency not found in any other graphics architecture.

Finally, the level of parallelism in this architecture creates a consistent growth path for performance. As the number of frame buffer memory planes is increased, the performance does not degrade but remains equal to a single plane architecture. In fact, with proper implementation in VLSI components, the true speed limitation of such an architecture is limited only by the speed of the DRAM used for the frame buffer.

VLSI IMPLEMENTATION

National Semiconductor Corporation is developing a graphics chip set to match this graphics architecture. The first four chips in this chip set will be introduced during 1986 and are shown in *Figure 16* for an example of 3 memory planes. The chip set is fully expandable to any number of memory planes.

The RGP (Raster Graphics Processor) is the main graphics processor, the BPU (BITBLT Processing Unit) the slave data manipulator. The VCG (Video Clock Generator) is a timing and control generator for the graphics system with

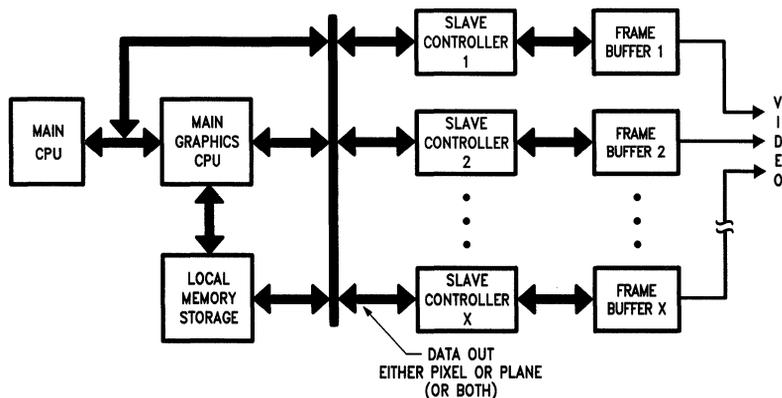


FIGURE 15. Optimum Solution BITBLT Based

TL/F/8769-15

capability of generating clocks in excess of 150 MHz. The VSR (Video Shift Register) acts as a parallel to serial converter capable of clock rates in excess of 150 MHz. The chip set has been designed to support all types of RAM components that may be used in the frame buffer, including Video DRAM.

The National Semiconductor architecture is very well suited for plane oriented architectures, but can work equally well in pixel oriented architectures. Additionally, it is the first chip set which will provide a complete solution for mixed mode applications.

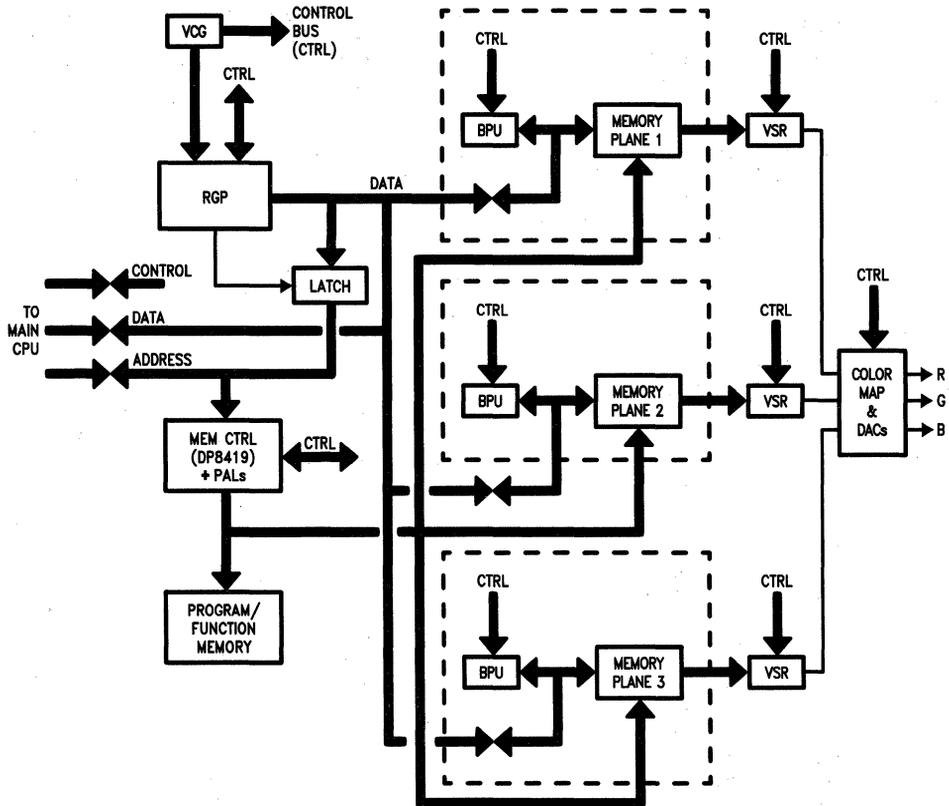


FIGURE 16

TL/F/8769-16

Interfacing the DP8500 Raster Graphics Processor

National Semiconductor
Application Note 547
Rob Wilson



1.0 INTRODUCTION

A graphics subsystem using National Semiconductor's Advanced Graphics Chip Set (AGCS) requires some interface circuitry between the DP8500 Raster Graphics Processor (RGP) and other devices on the bus. Specifically these devices may include:

- the Frame Buffers (memory) in each plane,
- the Bitblt Processing Unit (BPU) in each plane,
- a data transceiver in each plane,
- other types of memory used in the system, such as EPROM for basic functions and startup,
- memory-mapped I/O devices, including latches for control of color.

In general the interface circuitry required between the RGP and other devices will consist of two main blocks:

- the Bus State Machine (BSM), which generates memory strobes as required by the particular type of memory being accessed, and transceiver control signals, and
- the Video Plane Controller (VPC), which enables only selected planes of memory, and ensures correct data flow between memory planes.

Additionally an Address Decoder block produces various enable signals according to the address being generated by the RGP.

The RGP's bus interface has been deliberately kept general-purpose to allow flexibility of memory types; this means that the system designer must tailor the interface to suit his requirements. The purpose of this application note is to describe some of the issues and to suggest some solutions.

Memory Architecture and Graphics Operations

The AGCS has an open architecture which permits both planar and pixel operations. The memory is organized in planes (see *Figure 1*), in which the "color value" of a particular pixel is the parallel combination of the corresponding bits in separate words of memory. This requires that the graphics controller be able to access each plane separately; but by dividing the graphics rendering functions between the RGP (address generation) and a BPU on each plane (data manipulation), the AGCS allows parallel graphics processing. Thus the time required to render a particular graphics object is independent of the number of bits per pixel and the

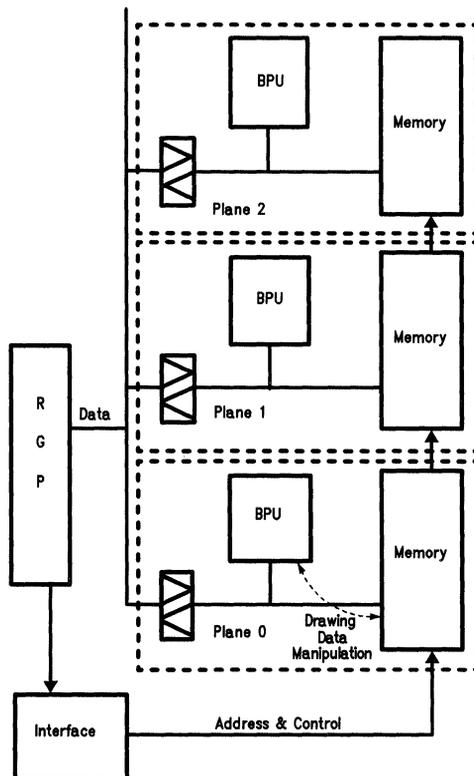


FIGURE 1

TL/F/9762-1

data processing bandwidth is matched to the memory bandwidth with uniform expandability.

Contrast this with the more common, but less powerful, packed-pixel architecture. This requires only a single graphics controller, with access to just one block of memory. Within the memory each word holds the "color value" of a number of pixels (for a 4 plane system, i.e., 4 bits per pixel, a 16-bit memory word would hold the color values of 4 pixels). With this architecture therefore, the time required to render a graphics object does depend on the number of bits per pixel—the system gets slower as the number of bits per pixel increases. Thus the performance is limited by the data processing bandwidth.

In addition to being able to perform planar operations (such as Character Drawing and Bitblt) at high speed, the AGCS can also execute pixel operations (such as Read and Draw Point) efficiently. This is achieved by means of the Pixel Port in each BPU, which allows the RGP, or some other processor, to access data representing a single pixel.

It is important to understand that the RGP is a processor—it fetches instructions from memory and executes them. Specific graphics instructions are "hardwired" rather than microcoded, thus providing the highest performance for rendering operations. During these operations the RGP does not process the data in each plane—this function is handled by the BPU. Thus the RGP does not need to have any "knowledge" of the number of planes attached to it.

This leads to the concept of addressing duality: during instruction fetches and operand reads and writes, the RGP is accessing just one word in memory, and data passes to or from the RGP—the "linear" type of access. During "drawing" operations, however, the RGP is providing only address and control information to memory. Drawing operations include Line Drawing, Bitblt, and higher level operations which use these, such as Text rendering, and also Screen Refresh.

Thus the system designer needs to build an interface which can recognize these two types of access, and act on memory accordingly. Furthermore, for drawing operations there must exist some mechanism in the interface circuitry to allow the RGP's programmer to select just one or more planes which will take part in the operation. This allows attributes such as color and intensity of the graphics objects to be controlled.

Figure 2 shows the block diagram of the interface logic.

The Pixel Port

The BPU has a multiplexer on-chip, which, in point-wise drawing mode (Line Drawing) where one pixel is modified at a time, selects one of the 16 bits of the destination word and latches it into a single-bit bidirectional port. Conversely a single bit can be latched into this port prior to drawing: the bit is replicated 16 times and presented to the BPU's Logic Unit along with the Destination word. Thus the selected bit of the Destination word can be modified by the latched bit in conjunction with the programmed logic function.

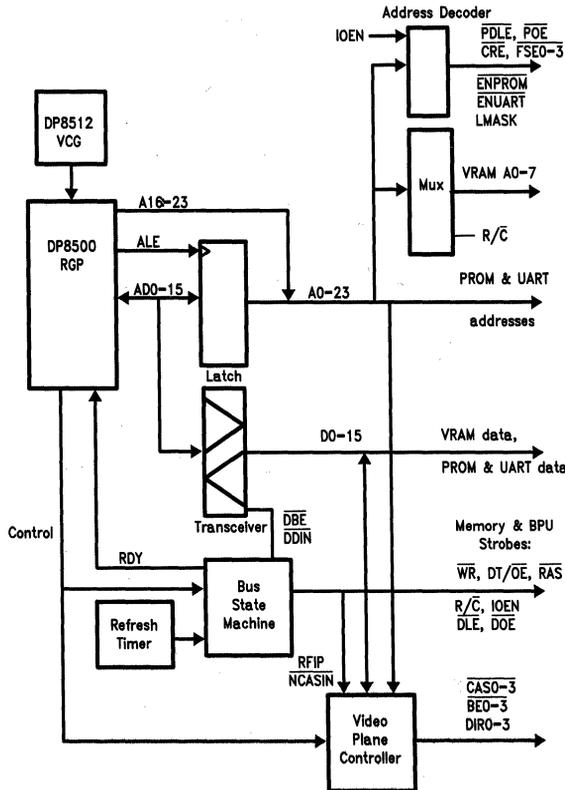


FIGURE 2

TL/F/9762-2

This mechanism is used in the RGP's Read Point and Draw Point instructions, which allow the RGP to read or write the "color value" of a single pixel. In addition to uses in color arithmetic this mechanism is useful in allowing pixel architecture data bases to be easily and rapidly translated to the AGCS's planar architecture.

Local and Interplane Bitblt

For Bitblt and derivative operations it is common for the Source and Destination rectangles to lie in the same plane (Local Bitblt). But it may be a requirement to have a source rectangle in one plane with the Destination lying in multiple planes (Interplane Bitblt). This would commonly be used for text rendering where the font bitmap is stored only once in one plane—perhaps in low-cost DRAM rather than Video DRAM, but will be required to be rendered in different colors by transfer to various Destination planes.

2.0 EXAMPLE SYSTEM

To illustrate the design of an RGP interface a typical small system will be used. The actual example is National's DP850EB, an evaluation board for the RGP. This is a stand-alone 4-plane system with 1M pixels per plane, which has basic system firmware in PROM, and a UART for communications with a Host computer. The main components of the system are:

- One DP8500 RGP
- Four planes, each containing one DP8511 BPU, four 64k x 4 Video DRAMs (VRAMs), a transceiver, and a shift register for parallel to serial conversion
- One NS16450 UART
- Two 2k x 8 87SR193 PROMs for resident firmware.

Memory Map

The RGP will start to execute instructions from address 0 after reset, thus the PROM space must begin at 0. The four planes are allocated separate, but congruent, memory spaces. This simplifies the system design (but systems which need very large amounts of memory for each plane might need to map each plane to the same space). These considerations, together with the desire for simple memory space decoding, result in the following memory map:

Space (Hex)	Address Range in Space	Total Words	Comments
PROM	000000–3FFFFFFF	4M	Resident Firmware
PLANE 0	400000–4FFFFFFF	64k	Bit-Maps for Each of the Four Planes
PLANE 1	410000–41FFFFFFF	64k	
PLANE 2	420000–42FFFFFFF	64k	
PLANE 3	430000–43FFFFFFF	64k	
BPU \overline{CRE}	800001	1	For BPU Control Reg
BPU \overline{PDLE}	800002	1	For BPU Pixel Port Load
BPU \overline{POE}	800004	1	For BPU Pixel Port Output
BPU0 \overline{FSE}	A00001	1	For BPU0 Function Reg
BPU1 \overline{FSE}	A00002	1	For BPU1 Function Reg
BPU2 \overline{FSE}	A00004	1	For BPU2 Function Reg
BPU3 \overline{FSE}	A00008	1	For BPU3 Function Reg
(The RGP can access all BPU \overline{FSE} s simultaneously at address A0000F)			
UART	C00000–DFFFFFFF	2M	
VPC	E00000–FFFFFFF	2M	For Video Plane Control

BPU Registers

The BPU Control registers must be memory mapped so that, when the RGP automatically loads them (in one access) prior to drawing, the interface logic can generate the required BPU control signal (\overline{CRE}). The BPU Base Address (800001 hex) needs to be loaded into the RGP upon initialization.

Similarly the BPU Function Select registers need to be memory mapped so that the user can load them individually. During the RGP's linear access to each register the interface logic needs to generate the required BPU control signal (\overline{FSE}).

Other I/O Devices

The Bus State Machine and Address Decoder produce an Enable signal (\overline{ENUART}), which, together with Read (\overline{R}) and Write (\overline{WR}) strobes, permit RGP access to the UART registers.

Memory

The system uses VRAM in page mode for drawing accesses only. There is negligible increase in performance for page mode during other VRAM accesses. The RGP's Page Break (\overline{PB}) output is useful in controlling page mode accesses. When High at the beginning of an access (\overline{PB} becomes valid during the second half of the RGP's T1 cycle) \overline{PB} indicates that the current access is within the same 256-word page as the previous access. A Low on \overline{PB} indicates that the access is to a new "page", or row, of memory.

3.0 BASIC TIMING

The example system uses Video DRAM with the following parameters:

RAS Access Time:	120 ns Max
RAS Pulse Width	120 ns Min
RAS Precharge Time:	90 ns Min
CAS Access Time:	60 ns Max
CAS Pulse Width:	60 ns Min
CAS Precharge Time:	50 ns Min (Page Mode)

At an RGP clock frequency of 20 MHz these parameters can be met by using the following:

RAS Low for 3 Clock Periods

RAS High for 2 Clock Periods

CAS Low for 2 Clock Periods

CAS High for 1 Clock Period (Page Mode)

4.0 ADDRESS DECODER

This uses the BSM's IOEN signal to generate the following "chip selects":

$\overline{\text{CRE}}$ BPU Control Reg Load

$\overline{\text{PDLE}}$ BPU Pixel Port Latch Enable

$\overline{\text{POE}}$ BPU Pixel Port Output Enable

$\overline{\text{ENPROM}}$ PROM Output Enable

$\overline{\text{ENUART}}$ UART I/O Enable

$\overline{\text{LMASK}}$ Latch VPC Control Data from RGP

$\overline{\text{FSE0}}$ BPU Function Select Reg Load, Plane 0

$\overline{\text{FSE1}}$ BPU Function Select Reg Load, Plane 1

$\overline{\text{FSE2}}$ BPU Function Select Reg Load, Plane 2

$\overline{\text{FSE3}}$ BPU Function Select Reg Load, Plane 3

These signals are derived using straightforward combinational logic. Their timing follows the BSM's IOEN (see next section).

5.0 BUS STATE MACHINE

This generates the following signals:

$\overline{\text{RDY}}$ RGP Ready Signal

$\overline{\text{RAS}}$ RAS signal to VRAM

$\overline{\text{R/C}}$ Row/Column Address Selector to Address Multiplexer

$\overline{\text{NCASIN}}$ Master $\overline{\text{CAS}}$ Signal to VPC

$\overline{\text{WR}}$ Write Signal to VRAM and UART

$\overline{\text{DT/OE}}$ Data Transfer/Output Enable to VRAM, Read Signal to UART

$\overline{\text{DDIN}}$ Data Direction Signal to Transceiver on RGP's Bus

$\overline{\text{DBE}}$ Data Buffer Enable to Transceiver on RGP's Bus

$\overline{\text{DLE}}$ BPU Data Latch Enable

$\overline{\text{DOE}}$ BPU Data Output Enable

$\overline{\text{RFIP}}$ Refresh in Progress to VPC

$\overline{\text{IOEN}}$ Linear Access Timing Strobe

5.1 BSM Basics

To derive a scheme for controlling the bus accesses consider that data flow during Drawing accesses can overlap with the start of the next RGP cycle, as shown in *Figure 3*. Indeed, the RGP's BPU FIFO control signals (FRD and FWR) are timed to use this overlap.

This causes no problem with Local Bitbit; for Interplane Bitbit however, since the Source data appears on the common bus during Source Reads, it is necessary to "cut" the RGP's data path for the first part of the overlapped access as this could be a linear access.

BSM TIMING

During each cycle of the RGP the BSM will need to change both its state and its outputs based upon its present state and inputs from the RGP.

To avoid having to use very fast logic the example system uses a pipelined technique for the BSM:

- Input changes occurring during a particular RGP cycle cause the BSM to change state at the beginning of the following cycle. The state is clocked by the rising edge of PH1.
- Output changes which occur as a result of this state change happen at the beginning of the cycle after this, and are clocked by the rising edge of PH1.

Consequences of this are that an input change occurring during cycle "N" will result in an output change in cycle "N + 2"—something that must be remembered when looking at the BSM timing diagrams which follow.

The one exception to this scheme is the signal $\overline{\text{NCASIN}}$, which is used by the VPC to selectively generate $\overline{\text{CAS0-3}}$ for the memory planes. $\overline{\text{NCASIN}}$ is a combinational BSM output, and is clocked by the rising edge of PH1 in the VPC after being processed by more combinational logic (see section 6.0). Thus $\overline{\text{CAS0-3}}$ become valid at the beginning of RGP cycles, just like BSM outputs. In the timing diagrams (*Figures 7 to 14*) which follow, $\overline{\text{CASn}}$ is shown rather than $\overline{\text{NCASIN}}$, the "n" representing any one or more of 0 to 3.

Figure 4 shows the structure of the BSM and the timing relationships between inputs, states and outputs.

BSM Bus Requests

There are two sources of bus access requests: the RGP and the Refresh Timer. The BSM contains two flip flops to denote a bus request from these sources:

$\overline{\text{RGPRQ}}$: Set by the RGP's ALE, Reset by the BSM

$\overline{\text{RFRQ}}$: Set by the Refresh Timer, Reset by the BSM.

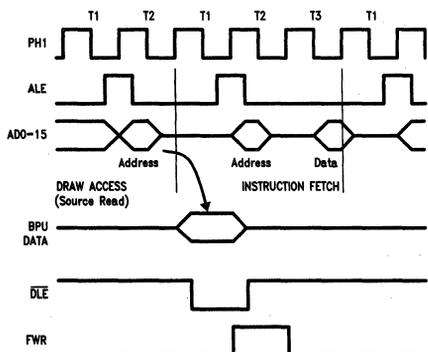


FIGURE 3

TL/F/9762-3

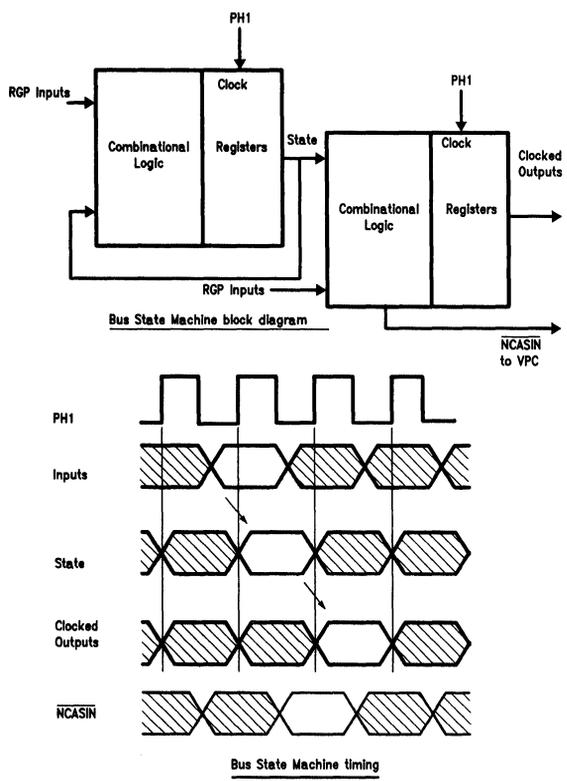


FIGURE 4. Bus State Machine Timing

TL/F/9762-4

The principle of the BSM described here is that, on getting a bus request from one of these sources (with RGPRQ having the higher priority), the BSM counts through a sequence of states, ultimately ending in state 0, the idle state, where it remains until another bus request occurs. The particular sequence used depends on the type of access required, and can be determined by decoding the RGP status lines and the Bus access requests:

Access Type	RGPRQ	RFRQ	BS1	BS0	\bar{R}	W
Linear Read	H	L	L	X	L	H
Linear Write	H	L	L	X	H	L
Bitbit Source Read	H	L	H	L	L	H
Bitbit Dest Write	H	L	H	L	H	L
Draw Read/Mod/Write	H	L	H	L	L	L
VRAM Transfer	H	X	H	H	L	H
VRAM Refresh	H	H	L	X	X	X
or	H	H	X	L	X	X

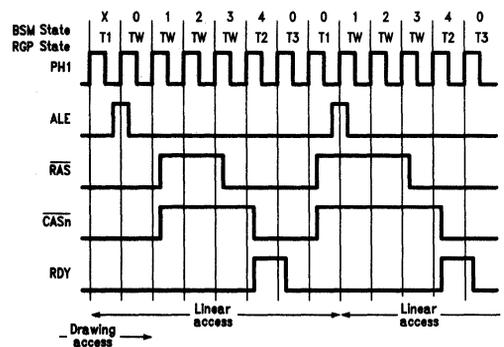


FIGURE 5

TL/F/9762-5

5.2 BSM Linear Accesses

The linear access is used for Instruction Fetches, Operand Reads and Writes, and general Input/Output. The previous access could have been either a Drawing access (\overline{RAS} ending in the cycle after T1) or another linear access (\overline{RAS} ends in previous T3). Both possibilities are shown in Figure 5.

To meet the VRAM timing requirements for the system we need to allow for the worst case (in which the previous access is a Drawing access). Thus \overline{RAS} for the Linear access can start two cycles later as in Figure 5.

The above analysis is for a relatively simple Bus State Machine. A more complex design could reduce the impact (1 extra clock cycle) of allowing for a previous Drawing access by recognizing a sequence of Linear accesses. In this case, because only two cycles of RAS precharge are needed, the access can be started one cycle earlier. Additionally, for the example BSM design, the same timing is produced for all linear accesses regardless of the type of memory being accessed—again, a more complex design could vary the timing according to the type and speed of memory being accessed.

The complete BSM behaviour for a Linear Read access followed by a Linear Write access is shown in Figure 6. Once

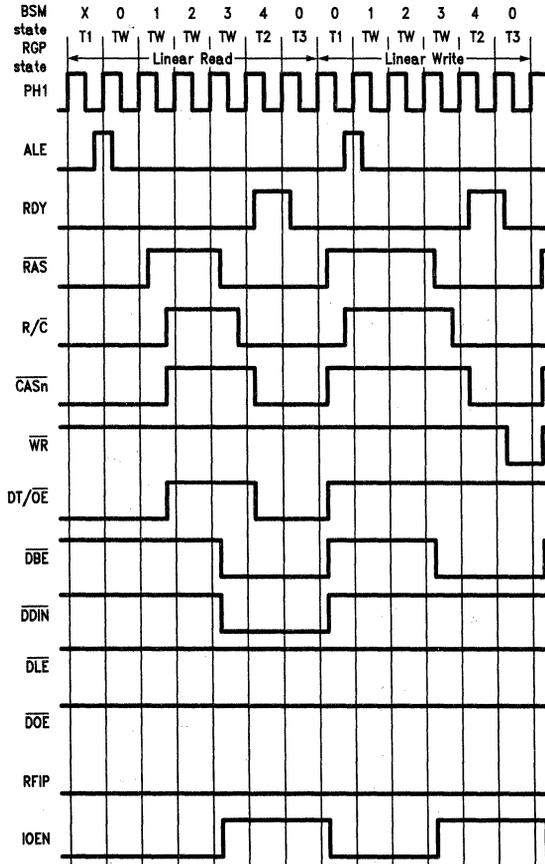
the previous access has completed (i.e., the BSM has reached state 0) the BSM determines that an RGP bus request is pending (RGPRQ has been set by ALE going high), and starts the Linear access by branching to state 1.

In subsequent clock cycles the BSM enters states 2, 3 and 4, and finally state 0. The RGP's RDY input is asserted during state 4 so that the RGP will complete the access, with data becoming valid at the RGP's inputs at the end of T3.

Note that R/\overline{C} , which switches the VRAM address lines between Row addresses (when High) and Column addresses (when Low) is generated by delaying the \overline{RAS} signal by half a clock period.

Also, since the RGP will latch the read data from the accessed memory or device at the end of T3 during the Read access and send data out during the write access, the RGP's Data bus transceiver is enabled by \overline{DBE} and its direction controlled by \overline{DDIN} . The timing strobe IOEN is produced, with the same timing as \overline{CASn} , so that the Address Decoder block can produce the appropriate strobes to the accessed device.

DT/ \overline{OE} pulses low to enable data from the accessed memory during the Read access, and \overline{WR} pulses low to write data to memory.



TL/F/9782-6

Conditions for Linear Access: BS1 = L, RGPRQ = H, RFRQ = L

FIGURE 6. Linear Read and Write, Preceded by Draw Access

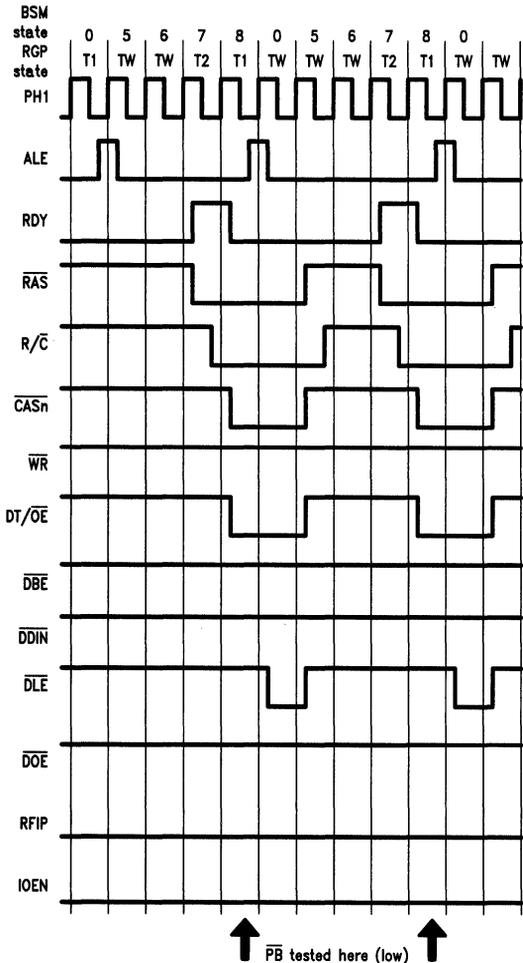
5.3 BITBLT Source Read and Destination Write Accesses

The first type of Drawing access to be considered is the Bitblt Source Read: this occurs when the BPU FIFOs are being filled prior to the Destination Write or Read/Modify/Write accesses during a Bitblt.

Figure 7 shows the BSM behaviour when a Page Break occurs, i.e., page-mode memory cycles are not possible because subsequent accesses are to different rows (or "pages") of memory. During state 8 the RGP's PB signal is determined to be Low, so the BSM branches to state 0 to terminate the access (by bringing RAS High).

DT/OE is produced to enable the data from memory, which is latched into the BPU's with DLE. Note that for interplane Bitblts only one memory plane is read, but the data may be latched in several BPU's: the VPC logic controls the data routing.

Figure 8 shows the BSM behaviour for page-mode memory accesses.



TL/F/9762-7

Conditions for Bitblt Source Read: BS1 = H, BS0 = L, RGPRQ = H, RFRQ = L, R \bar{R} = L, W \bar{W} = H

FIGURE 7. Bitblt Source Read, Non-Page Mode

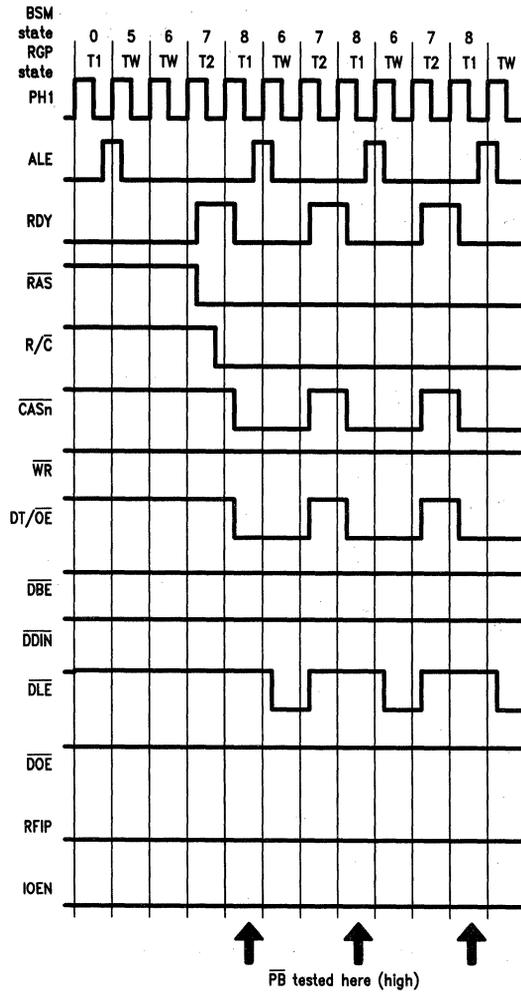
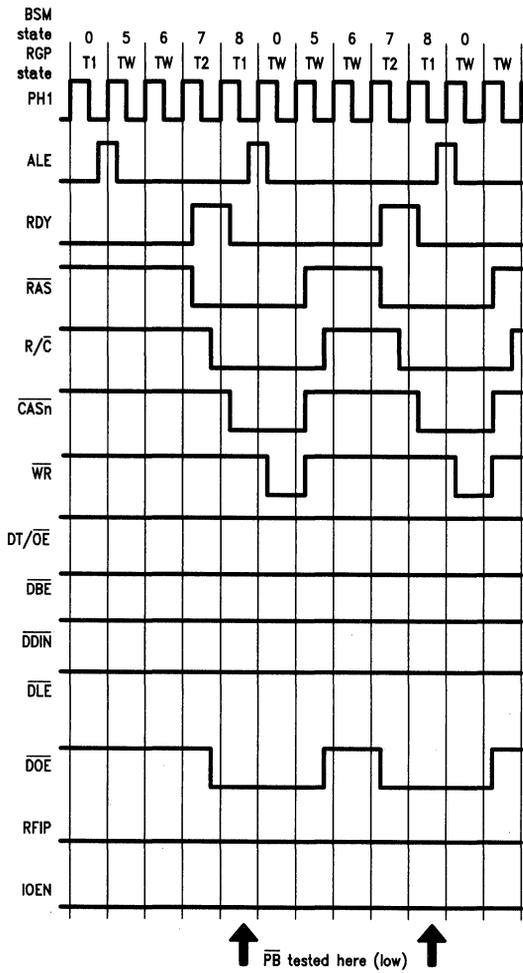


FIGURE 8. Bitbit Source Read, Page Mode

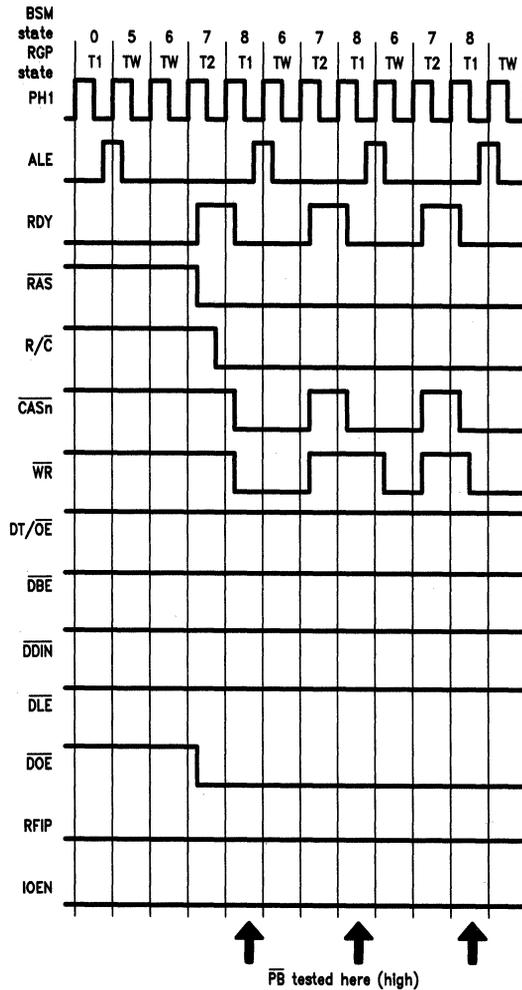
TL/F/9762-8



Conditions for Bitblt Destination Write: BS1 = H, BS0 = L, RGPRQ = H, RFRQ = L, R̄ = H, W = L

FIGURE 9. Bitblt Destination Write, Non-Page Mode

TL/F/9762-9



TL/F/9762-10

FIGURE 10. Bitbit Destination Write, Page Mode

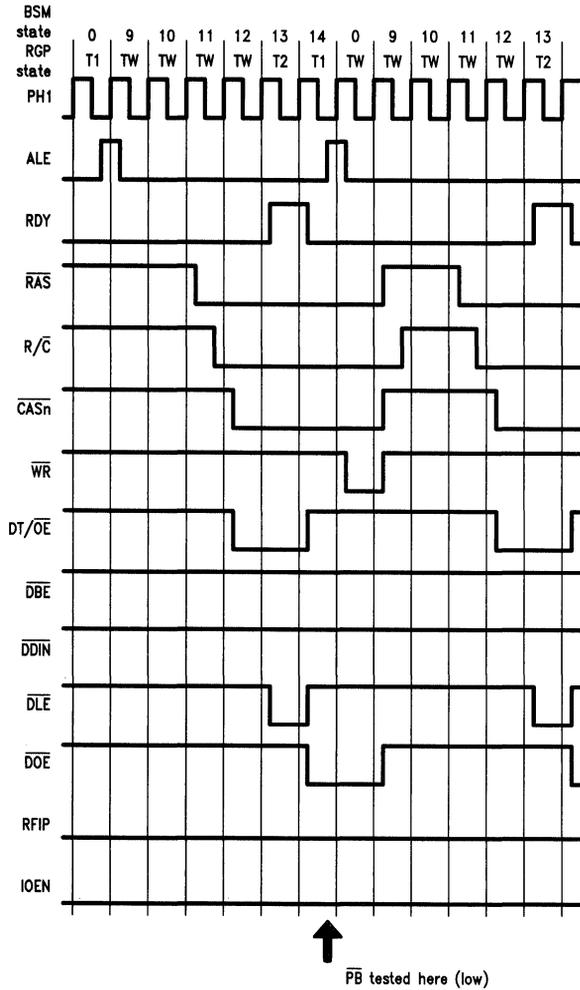
If, during state 8, the BSM finds that the next access is the other basic type of Drawing operation (Read/Modify/Write) and there is no Page Break then a branch will occur to the corresponding point in its state sequence (state 10).

Figures 9 and 10 show the state sequences for Destination Write accesses; these are identical to the states for Source Reads but the outputs differ. In particular the \overline{DOE} output is asserted during the \overline{RAS} pulse to enable data out of the BPU, and the \overline{WR} output is pulsed at the end of \overline{CASn} to write the data to the VRAM.

5.4 Draw Read/Modify/Write Accesses

This type of access occurs during both Line Drawing (always) and Bitblt (if the Bitblt Combine option is chosen, and for both this and the OverWrite option during modification of boundary Destination words).

Figure 11 shows non-Page mode accesses. The BSM uses states 9 to 14. DT/OE is pulsed during the first half of the \overline{CASn} signal to read the memory data into the BPU, where it is latched with DLE . (For Bitblt accesses the RGP's FRD signal simultaneously fetches the aligned Source data from the BPU FIFO).



Conditions for Draw R/M/W: BS1 = H, BS0 = L, RGPRQ = H, RFRQ = L, \bar{R} = L, \bar{W} = L

FIGURE 11. Draw Read/Modify/Write, Non-Page Mode

TL/F/9762-11

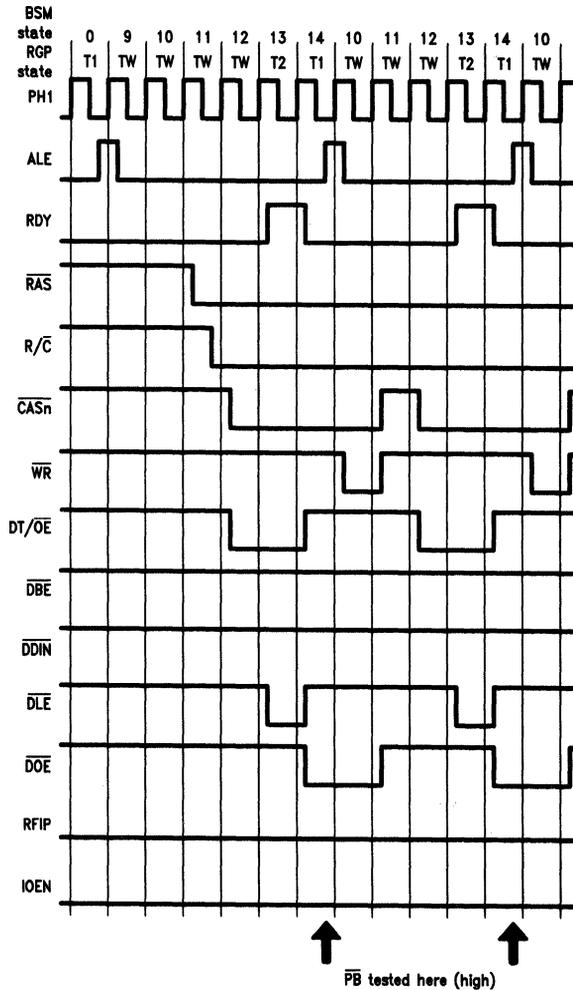
During the second half of $\overline{\text{CASn}}$ the BPU's logic unit output data is enabled out to memory by $\overline{\text{DOE}}$, and written to memory by $\overline{\text{WR}}$.

If, during state 14, the BSM finds that the next access is the other basic type of Drawing operation (Source Read or Destination Write) and there is no Page Break then a branch will occur to the corresponding point in its state sequence (state 6).

Figure 12 shows the same accesses in Page mode.

5.5 VRAM Transfer Access (Screen Refresh)

The RGP, if appropriately programmed, will request a bus access for VRAM data transfer at the beginning of each display scan line, or when the display row address is zero. During this access up to 256 words of data are transferred from the VRAM memory array to the VRAM shift registers, and from there are transferred to the display device.



Conditions for Draw R/M/W: BS1 = H, BS0 = L, RGPRQ = H, RFRQ = L, $\overline{\text{R}}$ = L, $\overline{\text{W}}$ = L

FIGURE 12. Draw Read/Modify/Write, Page Mode

TL/F/9762-12

The example system uses a display "warp" of 64 words, which avoids the requirement for VRAM transfers in the middle of scan lines. So VRAM transfer accesses occur only at the beginning of scan lines, during blanking time, and well before active video time.

The VRAM performs a transfer when DT/\overline{OE} is Low before \overline{RAS} is asserted. The BSM asserts DT/\overline{OE} in state 16, and then asserts \overline{RAS} in state 17, as shown in Figure 13. The remainder of the access is conventional except that the output RFIP is asserted. This tells the VPC that \overline{CAS} outputs to all planes must be asserted.

5.6 VRAM Refresh Access

When the Refresh timer causes a bus request, by setting RFRQ, and the current bus access has completed by reach-

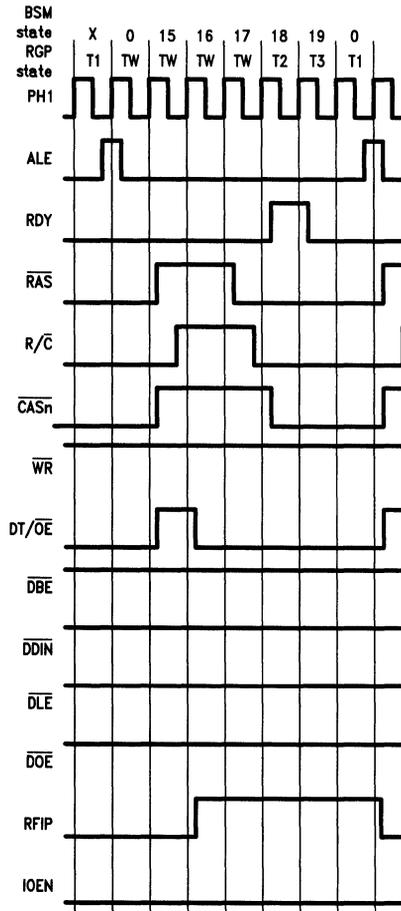
ing state 0, the BSM will initiate a VRAM Refresh access. This uses the VRAM's internal refresh row counter by asserting \overline{CAS} Low before \overline{RAS} .

The BSM inserts RGP wait states by holding RDY Low until the refresh access is completed—the pending RGP bus access can then continue. RFIP is asserted during the refresh so that the VPC will assert \overline{CAS} to all planes.

Figure 14 shows the BSM's behavior.

6.0 THE VIDEO PLANE CONTROLLER

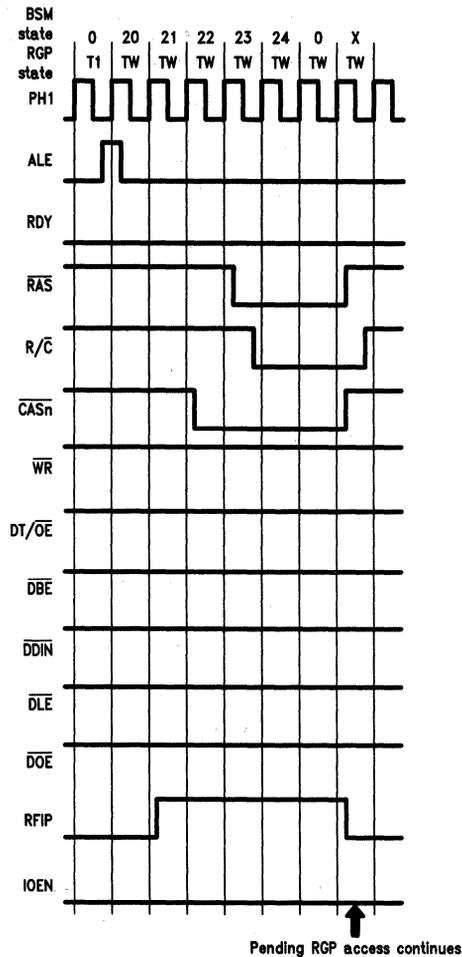
The function of the Video Plane Controller (VPC) is to enable, under program control, some or all of the planes for Drawing operations, and to control the data transceivers on each plane.



TL/F/9762-13

Conditions for VRAM Transfer: BS1 = H, BS0 = H, RGPRQ = H, RFRQ = X, \overline{R} = L, \overline{W} = H

FIGURE 13. VRAM Transfer (Preceded by Draw Access)



Conditions for VRAM Refresh: RGP_{RQ} = H, RFRQ = H, BS1 = L or BS0 = L

TL/F/9762-14

FIGURE 14. VRAM Refresh

Plane Memory Control

Several control methods are possible:

- Disabling inactive planes by programming their BPUs to the "Destination = Destination" function. These planes therefore write back their images without any changes. This method does not require additional hardware, but does not allow the use of interplane Bitblt (there will be bus contention during Source reads).
- Steering WR to selected planes, so that only active planes can change their data. This method also does not allow the use of interplane Bitblt.
- Steering RAS to selected planes. This method could incur extra delay in the RAS pulse, which in many designs can not be tolerated. Also this method is not guaranteed to work for all VRAMs.
- Steering CAS to selected planes. This method does work satisfactorily, and any extra delay in asserting CAS can usually be tolerated.

CAS steering is the method used in the example system. By using the CAS timing strobe, NCASIN, from the BSM prior to clocking on the rising edge of PH1, no additional delay in CASn is incurred.

The VPC must be programmed (by an Operand Write from the RGP to address E00007 in the example system) for which plane or planes are to be active—the "Write mask", and if an interplane Bitblt is required. Four RGP data bits, D0-3, correspond to the four plane's Write Mask, and D15 is used for the interplane bit. The signal (LMASK) which latches these bits in the VPC is generated by the Address Decoder logic.

During drawing operations, except for interplane Bitblt Source Reads, plane n is enabled (NCASIN from the BSM is routed to CASn and clocked on the rising edge of PH1) if its corresponding Write Mask bit is a 1.

During linear accesses (BS1 is Low) and the interplane Bitblt Source Reads (BSE and BS1 are High, BS0 is Low, and the interplane bit is 1) a plane is enabled if the RGP's address lies within the plane's address bounds.

During VRAM Transfer accesses and VRAM Refresh operations (RFIP is High) all $\overline{\text{CAS}}$ outputs need to be asserted for the duration of RFIP.

Thus in the example system the VPC needs to decode address lines A17, 18, 22 and 23 to detect the plane being addressed, and decode RFIP, BS1, BS0, BSE and the interplane bit.

It should be noted that during interplane Bitblt Source Reads, however, BS1, BS0 and BSE are not valid during state 8; at this time they will change for the next RGP access. Therefore the VPC must effectively latch their value during these accesses.

Plane Transceiver Control

The VPC also generates the Enable ($\overline{\text{BEn}}$) and Direction (DIRn) control signals for the transceiver on each plane. The sense of DIR is that when it is High data is moved from the VRAM to the common bus (i.e., for memory reads).

During linear accesses to VRAM (BS1 and A23 are Low and A22 is High) $\overline{\text{BEn}}$ is asserted according to which plane is being addressed (a decode of A17 and A18). DIRn is the inverse of RGP's $\overline{\text{R}}$ output.

During linear accesses to the BPU Control registers ($\overline{\text{CRE}}$ is Low), all $\overline{\text{BEn}}$ are asserted and DIRn is Low.

During linear accesses to the BPU Function Select registers ($\overline{\text{FSEn}}$ is Low), $\overline{\text{BEn}}$ is asserted according to which one or more registers is being accessed and DIRn is Low.

During Interplane Bitblt Source Reads all $\overline{\text{BEn}}$ are asserted. DIRn for the Source plane (the plane being addressed by the RGP) is High, and for the other planes (which will latch the Source plane's data in their BPU's) DIRn is Low.

During all other Drawing accesses the transceivers are disabled ($\overline{\text{BEn}}$ are High).

Thus for transceiver control the VPC needs to decode address lines A17, 18, 22 and 23, and BS1, BS0, BSE and the interplane bit. The earlier comment about the validity of the RGP status signals during state 8 of the Bitblt Source Reads applies to the transceiver control logic as well.

7.0 INTERFACING TO AN EXTERNAL PROCESSOR

The open architecture of the Advanced Graphics Chip Set means that an interface to an external or co-processor can be achieved in a number of different ways, depending on the designer's requirements.

The technique offering the highest performance is that of using shared memory, where the external processor can take control of the graphics subsystem's bus and access memory directly.

The interface design described above requires three basic sets of signals from the bus controller.

- Address Signals (Inputs)
- Data Signals (Bidirectional)
- Status Signals (Inputs, Plus RDY Output)

The BSM design could easily be modified to cater for an external processor by adding an input which indicates a bus request by the additional processor. Once the BSM has granted bus access (putting the RGP in HOLD or inserting Wait states), this processor can present its Address, Data and Status signals to the BSM, VPC and Address Decoder, and access the memory. The BPU's can also be controlled, allowing both planar and pixel accesses by the external processor.

8.0 SUMMARY

The implementation of a memory interface for the National Semiconductor DP8500 Raster Graphics Processor is straightforward. The RGP provides sufficient status and timing signals to interface flexibly and efficiently to any variety and mixture of memory and peripheral devices.

This application note has described the implementation of an interface between the RGP and various types of memory, by reference to the DP850EB system.

The LM1823: A High Quality TV Video I.F. Amplifier and Synchronous Detector for Cable Receivers

National Semiconductor
Application Note 391
Martin Giles



INTRODUCTION

The LM1823 is a video I.F. amplifier designed to operate at intermediate carrier frequencies up to 70 MHz, and employ phase locked loops for synchronous detection of amplitude modulation on these carrier frequencies. The high gain, wide AGC range and low noise of the LM1823 make it ideal for use in television receivers, video cassette recorders and in cable TV set-top converters requiring high quality detected base-band video and an audio intercarrier. Typical performance characteristics and features of this I/C are summarized in Table I below.

TABLE I

Maximum system operating frequency	70 MHz
Typical I.F. amplifier Gain (45.75 MHz)	>60 dB
I.F. amplifier gain control range	55 dB
True synchronous detector with a PLL	
Detector conversion gain	34 dB
Detector output bandwidth	9 MHz
Detector differential gain	2%
Detector differential phase	1 degree
Noise averaged AGC system	
Internal AGC gated comparator	
Reverse tuner AGC output	
DC controlled video detection phase	
AFC detector	

THE R.F. SIGNAL FORMAT

Despite the wide variety of signal sources available to the home television receiver—broadcast, cable, satellite, video games etc.—on channel carrier frequencies from 55.25 MHz to 885.25 MHz, the spectral content of each R.F. channel has been established for many years. In the United States the channel bandwidth is fixed at 6 MHz with the picture carrier located 1.25 MHz from the lower end of the band, and an aural carrier placed 4.5 MHz above the picture (pix) carrier. Introduction of color television in the early fifties added another carrier, the chroma sub-carrier, positioned 3.58 MHz above the pix carrier frequency. The pix carrier is amplitude modulated* by the baseband video signal (which

* A more appropriate term is "negative downward modulation" since any modulating signal causes a decrease in the peak carrier amplitude (compared to conventional a.m., where the modulating signal alternately increases and decreases the peak carrier level with the mean carrier level remaining constant). For television carriers, syncs correspond to peak carrier and increasing brightness causes decreasing carrier amplitudes.

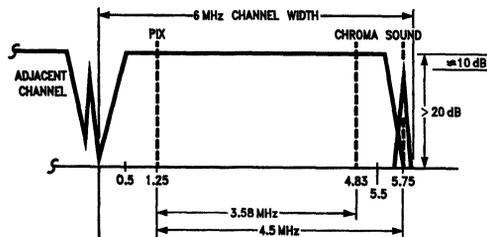
includes the synchronization information and the phase and amplitude modulated chroma subcarrier) while the aural carrier is frequency modulated. Television channels in Europe use similar carriers with the refinement of a fluctuating chroma subcarrier phase (P.A.L.).

The signal coming into the receiver has this general format and the receiver R.F. and I.F. circuits are designed to handle such a signal and reduce it back to the baseband composite video and audio intercarrier. Even where signal scrambling is used to protect the video modulation from unauthorized detection, the R.F. spectrum must remain within this format. For satellite broadcasts with frequency modulation of the video signal, the signal is demodulated and then remodulated onto a low VHF channel for reception by standard television receivers. In connection with this, the LM1823 PLL detector is not suitable for wide-band FM detection—even though the I.F. carrier (70 MHz) is well within the LM1823 I.F. amplifier frequency capability.

Notice again that the pix carrier is located at one end of the occupied bandwidth and only the upper sidebands are being fully transmitted. The lower sidebands are truncated with only frequencies close to the pix carrier frequency modulating the carrier. This method of conserving the frequency spectrum is referred to as vestigial sideband transmission.

THE CABLE CONNECTION

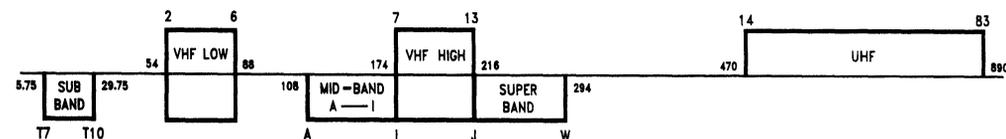
Originally introduced many years ago as a means for providing broadcast TV to isolated areas or where the terrain made direct reception difficult, cable TV had modest growth in the U.S. and was a stagnating industry until the mid-seventies. Lower cost satellite earth stations were the turning point, allowing cable operators access to many varied program sources from any part of the country.



TL/H/8421-1

FIGURE 1. U.S. Broadcast Channel Spectrum

Broadcast Channel Frequency Allocations



Cable Channel Frequency Allocations

FIGURE 2. Broadcast and Cable Bands in the U.S.

TL/H/8421-2

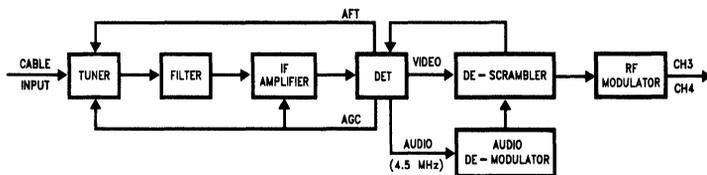


FIGURE 3. Cable Set-Top Converter Block Diagram

TL/H/8421-3

Standard television receivers in the U.S. tune to VHF channels 2 through 13 and UHF channels 14 through 83, and initially cable operators used the 12 VHF channels for their program material. With increased sources soon all channels were occupied on some systems creating significant demands on television tuner and I.F. amplifier strips. More space yet was needed and rather than using UHF channel allocations starting at 470 MHz because of cable signal attenuation (typically 0.8 dB/100 ft. at 300 MHz), operators turned to the unused spectrum space between VHF channel 13 and UHF channel 14. Naturally, since standard TV receivers could not tune to these channels, the set-top converter came into being. Each of the new channels could be converted to a low VHF channel to be received on the standard TV. Television manufacturers responded, and with the common introduction of varactor tuners were soon able to offer "cable ready" televisions capable of tuning to all the new cable frequencies. This meant that customer conveniences such as remote control of channel selection also became available. Unfortunately it aggravated a problem already confronting the cable operator. Since standard television receivers couldn't tune to the cable channels, operators had been able to offer premium services on some of these frequencies, paid for by subscribers who rented the appropriate set-top converter box. This didn't prove very secure since one operator's "free" channel was another operator's "pay" channel, and the introduction of cable-ready televisions ensured the eventual demise of such systems.

Scrambling the signal, a technique already being used by over-the-air subscription television, has become common in the cable service. The degree of scrambling* is limited since the scrambled signal spectrum must remain within the channel allocation and anything done to the signal must be subsequently undone without noticeable degradation of the signal.

Generally for television, scrambling means a pulse or sine wave suppression of the signal horizontal blanking pulse interval so that the sync-tips occur between the black and white levels instead of always below black level. The standard television sync separator does not function well with this signal and the I.F./tuner AGC circuits will not work properly, effectively scrambling the displayed picture. Other techniques include random inversion of the video information to provide an even greater degree of security.

The means used to encode such a scrambled signal gives rise to the terms "in band scrambling" and "out of band scrambling". With cable ready television receivers capable of tuning to the scrambled channel, the decoder can be a simple broad-band gain switch (to change the signal R.F. amplitude during horizontal blanking) with a separate receiver tuned to the decoding data carrier frequency, which is

*Other security techniques such as jamming or trapping are used but since jamming is easy to defeat and trapping requires removal or replacement of filters in the cable drop to individual subscribers, scrambling the signal is receiving a lot more attention.

located outside the signal channel. This permits use of the television receiver in a normal way but does require simultaneous switching of the decoder receiver with channel changes.

Also, spectrum space must be reserved for each scrambled channel's data carrier.

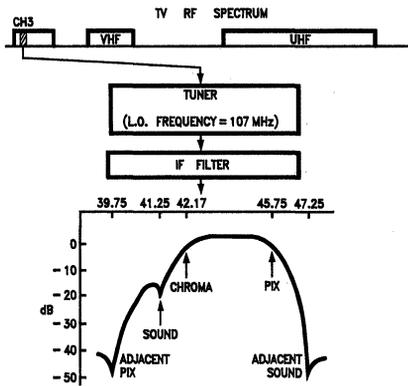
A more popular method of scrambling is "in band scrambling" where the data carrier to decode the signal is included inside the transmitted signal channel, usually within the aural carrier. Any number of channels can be scrambled and now different levels of service can easily be added or deleted without the need to rewire the decoder box. This is achieved by including time multiplexed binary "tags" along with the sync information so that special programs can be identified. Individual subscriber boxes can be similarly addressed and turned on or off by the cable operator. In these types of systems, the LM1823 and LM2889 have obvious applications. The LM1823 is able to provide an excellent baseband signal inside the decoder box, which signal is then remodulated on a low VHF channel carrier by the LM2889 for retransmission to the standard television receiver. Clearly, the highest possible performance is desirable to prevent any noticeable difference between a converted channel, whether scrambled or not, and a regular off-air broadcast channel. (For a complete description of the LM2889 modulator I/C see AN402).

THE RECEIVER FRONT-END

The typical receiver front-end consists of a tuner, I.F. amplifier, I.F. filters and a video/sound intercarrier detector stage. These circuits are designed to provide a number of functions:

- 1) Select (tune) a specific R.F. channel in a band of frequencies.
- 2) Provide rejection to adjacent and other channels in the band.
- 3) Amplify low level R.F./I.F. signals prior to detection of the modulation.
- 4) Avoid overload on high level R.F. signals.
- 5) Trap or attenuate specific frequencies within the channel bandwidth to ensure a proper detected frequency response is obtained.
- 6) Linearly demodulate all desired modulating frequencies on the carrier.
- 7) Produce a noise-free video signal at the detector.
- 8) Provide automatic gain control (AGC) to compensate for changing signal strength at the receiver input.
- 9) Provide automatic frequency control (AFC) to the tuner local oscillator (L.O.) to maintain the carrier intermodulation frequency (I.F.).

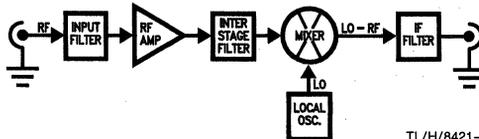
N.B. Items 8) and 9) have previously been provided in part by circuits external to the conventional I.F. amplifier. However, these functions are completely included with the LM1823 leading to overall performance improvements and reduction in external parts count and cost.



TL/H/8421-4

FIGURE 4. R.F. Tuning and I.F. Conversion (Note High Side L.O. Reverses the Relative Position of the Picture Chroma & Sound Carriers. c.f. Figure 1).

Although we are not directly concerned with the tuner design in this application note, it is useful to understand the design goals and constraints on the tuner for at least two reasons. First, since the tuner and I.F. amplifier interact very closely to obtain and maintain a noise-free picture, we need to know something about the tuner in order to provide the correct gain distribution and AGC action. Second, when the two functions are finally placed together, we need to know where to look to solve visible problems that may have become apparent. In some instances, either the tuner or the I.F. amplifier may be at fault, and a good understanding of the system interaction is needed to ensure that the appropriate action is taken.



TL/H/8421-5

FIGURE 5. Typical Single Conversion Tuner

Both single conversion and double conversion techniques are used in cable converter tuners. The single conversion type is similar to the conventional TV receiver tuner and consists essentially of an R.F. stage, mixer stage, and local oscillator. Usually some input filtering is done to help match the cable to the input device and provide some rejection to unwanted signals outside the operating channel. Further rejection to unwanted signals, such as the I.F. frequency radiated back from the I.F. amplifier, is accomplished with inter-stage filtering between the R.F. amplifier and the mixer, and finally an output filter matches the mixer output to the cable feeding the I.F. amplifier. For convenience, we are assuming

the desired output impedance is 75Ω and that the major I.F. amplifier frequency selectivity is determined by a block filter placed between the tuner output and I.F. amplifier input. This is consistent with modern practice using surface acoustic wave filters (SAWF's) and high gain stabilized I/C amplifiers (LM1823). Even so, as noted in more detail later, the LM1823 does provide opportunities for more filtering at the I.F. amplifier output prior to the detector stage.

Dual conversion tuners have been popular for a number of years and use first L.O. frequencies that are above the input R.F. bandwidth, avoiding problems with L.O. leakage back onto the feed cable. The second L.O. and mixer convert the high first I.F. to a Ch 3 or Ch 4 carrier for reception by the TV receiver. The addition of PLL's to control the first L.O. and descrambling networks on the R.F. output have added sufficient complexity to such converters that they are now called "set-top terminals". Also, since the scrambling techniques have become more sophisticated the signal is now frequently converted down to baseband before decoding and re-modulation on Ch 3 or Ch 4 carriers. The high first I.F. has the advantage that image signal rejection is achieved without the switchable filters necessary at the input to the single conversion tuner. However, the absence of these filters does mean that care must be exercised to avoid generation of intermodulation products that "talk back" onto the cable (up conversion of the R.F. signal has been proposed as a way to minimize intermodulation components). Another disadvantage of the dual conversion tuner shown in Figure 6 is that it typically has a very high Noise Figure, often between 14 dB to 16 dB. This is because the signal is applied directly to the first mixer which is a passive, double balanced diode mixer. As discussed in more detail later when we look at SAWF's between the tuner and the I.F. amplifier, a pre-amp in front of the mixer can improve the N.F. to 6 dB to 8 dB, especially in a baseband converter where an AGC voltage is available to help the tuner handle the input signal strength range.

Returning to the single conversion tuner, the major parameters to be considered are as follows:

- 1) Power gain
- 2) Noise Figure
- 3) Good Cross-Modulation rejection
- 4) VSWR
- 5) AGC Range
- 6) Impedance changes with AGC
- 7) Overload capability
- 8) Channel 6 beat rejection
- 9) Curve tilt (tracking)
- 10) L.O. drift and radiation

For an I.F. amplifier design, items 1), 2), 7), and 8) are the most significant, but if the tuner designer has overlooked the others we may see some problems when the tuner and I.F. amplifier are hooked together.

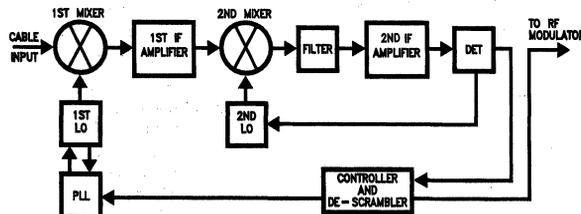


FIGURE 6. Dual Conversion Tuner

TL/H/8421-6

Crossmodulation describes the condition wherein the modulation information on an adjacent channel (usually) is transferred on to the desired carrier. A typical specification is the undesired carrier level with 30% modulation needed to cause 1% modulation of the desired carrier level.

Crossmodulation is particularly likely to occur in cable systems and is usually observed as sync bars drifting through the picture. In particularly severe cases the interfering picture can actually be seen. High signal levels at the input of the mixer are a frequent cause of crossmodulation, particularly when high gain R.F. stages are used to obtain a low tuner noise figure (N.F.). But when AGC is applied the crossmodulation source often shifts to the R.F. device.

When overload occurs, (measured as the total harmonic distortion of a specified modulation frequency), the peaks of the R.F. carrier waveform become compressed and this will show up at the video detector as a smaller sync pulse amplitude (sync tip to black level). Since the AGC system operates on the sync tip level the effective result is that the black level appears to go blacker than black—i.e. some near black information will be lost and the picture will appear to have too much contrast. Alternatively if the subsequent receiver circuits have black level restoration the screen brightness increases and picture tube blooming on peak whites may occur. As overload increases there is a strong chance that vertical sync will be lost. Generally the tuner mixer device is the first stage to overload, followed by the R.F. stage. While overload is caused by very strong signal strengths and therefore may appear to be of limited concern it can also occur at weak to intermediate signal strengths because of incorrect AGC threshold settings and this will be discussed in detail later.

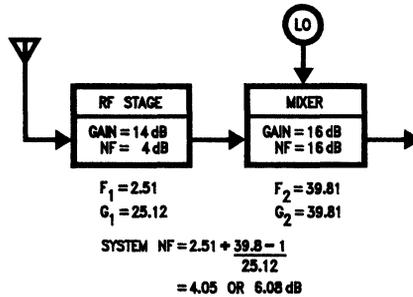
Channel 6 beat is a phenomenon related to mixer overload and occurs because of the choice in the U.S. of 45.75 MHz as the intermediate frequency. On channel 6, mixing of the sound and pix carriers produces a signal at 171 MHz which is then mixed with the channel 6 L.O. frequency to give 42 MHz. The I.F. sound and pix carriers can also mix with the channel 6 L.O. to produce 42 MHz. Since 42 MHz is only 170.455 kHz from the I.F. chroma subcarrier of 42.17 MHz, after detection wavy lines will appear in colored areas of the picture. Turning down the receiver color level (saturation) control will eliminate the 170 kHz pattern and identify the problem as Channel 6 beat.

Curve tilt or tracking refers to the ability of the tuner filters to track the L.O. frequency as the channel selection is changed. Problems in this area are easily identified at the video detector output (sometimes referred to as the 2nd detector) since the effect is to cause changes in the relative amplitudes of the pix, sound and chroma carriers compared to that expected from the I.F. filter response. When the detector VCO and AFT circuits of the LM1823 are aligned to 45.75 MHz, the chroma burst located on the back porch (or breezeway) portion of the horizontal blanking period in the video signal will normally be -6 dB compared to the sync pulse amplitude. If mistracking is causing a loss of high frequencies on certain channels, the burst amplitude will be lower on these channels and the picture (in severe cases) will have watery and noisy colored areas with smeared off picture detail. When the loss occurs down at the pix carrier

frequency, the burst amplitude is increased and the picture will become harsh with excessive overshoots.

Similar problems can occur on any specific channel simply due to mis-tuning or L.O. drift. In particular, as the L.O. frequency drifts high and the chroma subcarrier amplitude increases, the sound carrier also increases and chroma/sound beats will appear in the picture. In the U.S. the chroma/sound carrier beat is at 920 kHz (4.5 MHz—3.58 MHz) and appears as a herringbone pattern while the audio modulates the sound carrier. This 920 kHz beat can also be caused by detector non-linearities, and after the video detector by the detected 4.5 MHz sound intercarrier mixing with the chroma subcarrier in subsequent receiver stages. If turning down the color level control removes the 920 kHz beat then a better 4.5 MHz trap is needed at the video detector output.

These preceding comments are not meant to imply that the tuner is the root cause of all the nasty phenomena that can be observed in the picture display. Overload, Channel 6 beat and video noise are very dependent of the tuner/I.F./AGC interaction. To understand why this is the case, we need to look at the demands that the input signal field strength puts on the system.



TL/H/8421-7

FIGURE 7. Typical Tuner Gain and Noise Figure

INPUT SIGNAL LEVELS

The smallest input signal is, of course, no signal or simply the noise level generated at the cable drop. To this noise level will be added the input noise of the tuner itself, giving rise to an equivalent noise input defined by the tuner noise figure (N.F.) While a specific design will have to take into account the actual operating parameters of the tuners available, we will assume a typical tuner configuration with an R.F. stage providing 14 dB gain and having a 4 dB N.F., followed by a mixer stage with 16 dB conversion gain and a 16 dB N.F. The N.F. of this combination is 6 dB, a fairly typical number, which will have the effect of increasing the actual input noise by a factor of 2. If our noise source is the cable impedance with a real part of 75Ω, at an ambient temperature of 290k, then the equivalent input noise is 2.2 uVrms (the noise contribution of any matching network or cable termination is ignored as this is included in the tuner N.F.).

Cable signal levels run from -6 dBmV to +15 dBmV with a typical system goal of maintaining a C/N ratio of at least

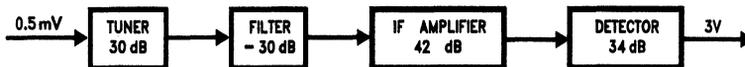


FIGURE 8. System Gain Distribution

TL/H/8421-8

43 dB at the cable drop to the subscriber. If a 0.5 mVrms signal is to produce the rated detector output of 3V (o-p)* for the LM1823 then we need a total system gain of at least 75 dB. Usually the SAWF connected between the tuner output and the I.F. input will have an insertion loss of 20 dB to 30 dB so that with the 30 dB tuner gain, the I.F. amplifier/detector is required to provide the remaining 76 dB. If the tuner is simply a diode mixer with a 6–8 dB insertion loss, the gain requirement increases to 114 dB.

* (o-p) means the detected zero carrier voltage level to the detected sync tip voltage level. The actual peak white signal to sync tip excursion at the detector will be 87.5% of this—2.63V (p-p). In the absence of a carrier, thermal noise will be present with amplitude peaks on both sides of the detected zero carrier voltage.

Fortunately the LM1823 has a high conversion gain detector (34 dB) and the I.F. amplifier gain can be set to well over 75 dB at 45.75 MHz (but we will see that some gain prior to the I.F. amplifier filter will be necessary if a good system N.F. is desired). Substantially more gain than necessary should be avoided however, even though there is plenty of AGC range in the I.F. amplifier (from 48 dB to 60 dB depending on external components). While at least 22 dB AGC capability is needed to accommodate the expected input signal strength range, if excessive system gain is used, forcing the I.F. amplifier into early gain reduction, the I.F. amplifier N.F. will begin to increase. With a diode mixer front end, the I.F. amplifier N.F. may contribute directly to the system N.F. and prevent noise-free pictures from being obtained. If a pre-amp or tuner is part of the AGC loop, gain reduction should be limited to the I.F. amplifier as much as possible, transferring gain reduction to the tuner only when the signal strength is high enough to cause distortion or cross modulation problems. The tuner gain will prevent the prior increase in I.F. amplifier N.F. from impacting the system noise performance, but excess system gain causing premature tuner gain reduction will increase the tuner N.F. and hence the system N.F.

Of great interest to us is the R.F.C/N ratio required for the detected output to be considered noise free. Actual television video S/N ratios are a little complicated by the fact that the displayed video signal does not occupy the full R.F. carrier envelope. 25% of the carrier is reserved for the synchronizing pulses and 12 1/2% is retained even under conditions of peak white modulation, for the benefit of intercarrier sound detectors. A common definition of the video S/N ratio is the ratio measured in decibels of the peak video signal amplitude to the r.m.s. noise voltage amplitude. In this context peak video refers to the voltage excursion between black and white levels (from 75% peak carrier to 12 1/2% peak carrier). With this definition in mind, it is generally accepted that the subjective effect of imperceptible noise occurs at an S/N ratio of 43 dB. Noise will become perceptible (for most viewers) at an S/N ratio around 38 dB; is clearly visible but not necessarily disturbing at 34 dB and becomes objectionable at 28 dB to 30 dB. Alternatively if we measure the signal amplitude as an r.m.s. sine wave with the same peak to peak amplitude as the R.F. carrier during the sync

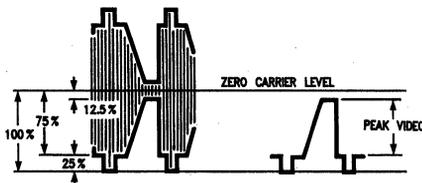


FIGURE 9. Television R.F. Modulation Envelope

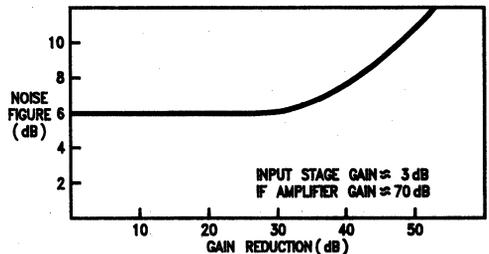
TL/H/8421-9

pulse period, our signal is free of noise for a 47 dB C/N ratio.

If the input signal were completely noise-free (i.e. no excess noise from head-end amplifiers etc.) then the detected C/N ratio is determined by the equivalent input noise level of the tuner—2.2 uVrms for a 6 dB N.F. With a minimum signal level of 0.5 mVrms the detected C/N ratio will be 47 dB for the converter alone. When the actual signal has noise, for a cable C/N ratio of 43 dB the noise detected at the converter output is now

$$e_n = 10^{-6} \sqrt{(2.2)^2 + (3.5)^2} = 4.13 \mu\text{V}$$

This gives a detected C/N ratio of 41.6 dB, a loss of 1.3 dB compared to the original signal. For most viewers this is the just perceptible level for video noise. On the other hand, if a 14 dB N.F. converter is used, the detected C/N is 32.7 dB which is considered objectionable. A 0 dBmV signal would produce 38.7 dB C/N ratio which would be acceptable. Obviously a low N.F. is important, and any increases in N.F. should be carefully controlled to get the best picture quality possible. Figure 10 shows the change in N.F. for the LM1823 I.F. amplifier. For over 30 dB gain reduction, the N.F. is unchanged and increases by only 4 dB for the next 20 dB of gain reduction.

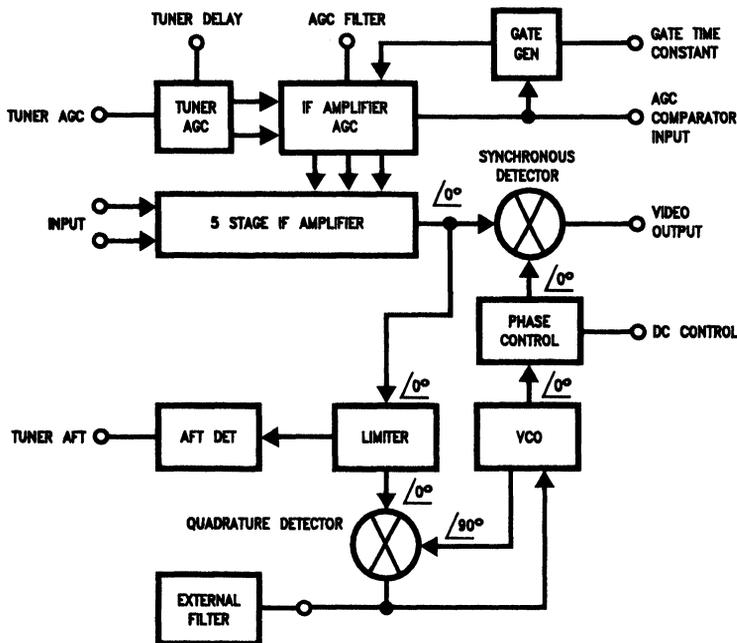


TL/H/8421-10

FIGURE 10. Increase in I.F. Amplifier N.F. with Gain Reduction

LM1823-GENERAL CIRCUIT DESCRIPTION

The basic arrangement of the LM1823 is shown in Figure 11. A five stage I.F. amplifier provides gain with a low impedance input stage to ensure adequate suppression of triple transit echo in SAW filters, and AGC on the three interstages. The output stage buffers the I.F. signal which is split off into two paths. A linear path takes the modulated signal to a true synchronous detector while a high gain limiter amplifier passes the I.F. carrier waveform to a second phase detector which is part of the PLL for the VCO. The PLL has an externally adjustable filter and locks the oscillator in quadrature with the incoming I.F. carrier. An in-phase component of the oscillator also drives the linear path detector to recover the signal amplitude modulation. An external DC control allows fine adjustment of the detection phase in order to optimize the detector linearity. The output from the detector is coupled back into the AGC comparator input, and is internally gated during the sync pulse period for good noise immunity and a fast response. Two AGC voltages are available; an early AGC for the I.F. amplifiers and a late, or delayed AGC for the tuner. The take-over point between the I.F. AGC and the tuner AGC is set by an external potentiometer. Also included is an AFT output for fine control of the tuner L.O. All these functions are contained in a 28-pin DIP with a pin-out designed to facilitate stable p.c.b. layouts—even with the high system gain of the LM1823 at frequencies up to 70 MHz.



TL/H/8421-11

FIGURE 11. Block Diagram of the LM1823

I.F. Amplifier Stages:

The LM1823 I.F. amplifier is composed of five separate stages designed to provide high gain primarily in the frequency range of 35 MHz to 60 MHz, and gain control over a 60 dB range without overload of any stage and without introducing excess noise into the signal.

To achieve this, AGC is applied to the second through fourth stages by a control voltage that is either internally generated from the video detector output or from an externally applied bias voltage at Pin 13. AGC action starts when the voltage at Pin 13 reaches approximately 4 VDC and over 50 dB of gain reduction is obtained by the time Pin 13 voltage reaches 6.5 VDC. For a typical application, the I.F. noise figure is around 6 dB for the first 30 dB of gain reduction, and then begins to increase to above 10 dB by the time the amplifier is gain reduced over 50 dB (see Figure 10).

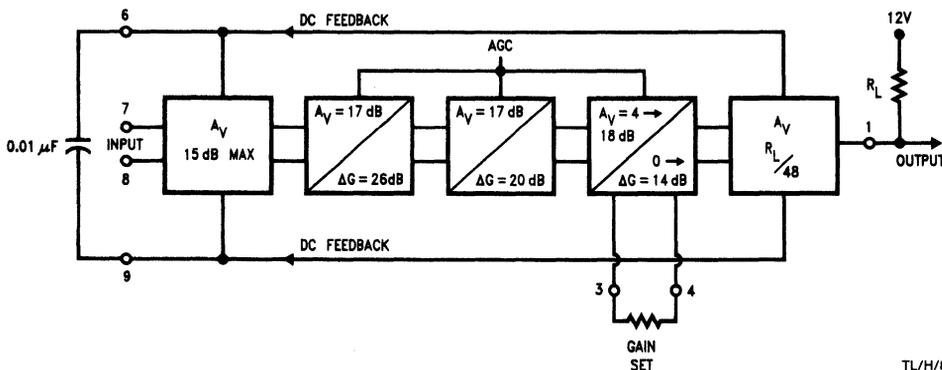
As mentioned earlier, the total system gain desired from the I.F. amplifier input to the video detector output needs to be

selected for a specific set of tuner parameters and I.F. filter losses. Excess gain simply means premature AGC action with possible loss of optimum video S/N ratios. To see how and where the LM1823 gain can be adjusted, we will look at each gain stage in turn.

Input Stage:

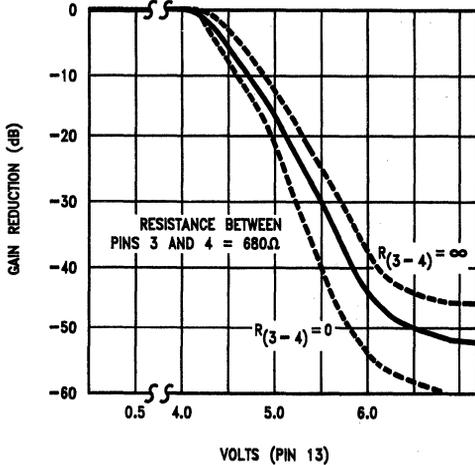
The input stage is a common-base differential amplifier designed to give good rejection of unwanted I.F. output and detector VCO signals that may be radiated back to the input. The low input impedance of 60Ω ensures that SAW filters are terminated sufficiently to keep the TTE better than 40 dB below the signal level, even with low impedance SAWF's. Because it is a common base stage, the input stage gain is determined by the source impedance presented to the input. An approximate expression for the gain is given by Equation (1)

$$A_v = 531 / (Z_s + 60) \tag{1}$$



TL/H/8421-12

FIGURE 12. Gain Distribution in the I.F. Amplifier



TL/H/8421-13

FIGURE 13. I.F. Amplifier Gain Reduction Characteristic

As an example, if we use a high impedance SAW filter such as the Murata SAF45 MC series with an output impedance that can be modelled as a 2.8 k Ω resistor in parallel with 8 pF capacitance, our input Zs is 345 Ω (including 2 pF input stray capacitance) at 45.75 MHz. From (1), the input stage gain is 2.4 dB. If a filter is used that matches to the input stage with 60 Ω , then the gain can be as much as 13 dB.

A balanced input is extremely important since the input leads Pins 6-9 are the most sensitive parts in the system to unwanted I.F. coupling. For example, if the I.F. output couples into these pins it can cause changes in the frequency response and can easily promote oscillation. A spectrum analyzer is invaluable for helping determine the system susceptibility to this phenomenon. With the input terminated by the I.F. filter (or an equivalent resistor), the I.F. amplifier output noise spectrum will show if oscillation is likely to occur.

Another signal that can appear at the input is the detector local oscillator waveform. Unlike quasi-synchronous detectors, the LM1823 has a constant (and relatively high) oscillator signal for good linear detection, even with low input signal levels. It is the balance between the input pins to the VCO radiation pick-up that will determine whether the p.c.b. layout is good enough. VCO pick-up can cause AFC skewing and asymmetrical oscillator pull-in, but probably the most serious effect is failure of the oscillator to acquire lock at weak signal levels. This is caused by the fact that the PLL

phase detector sees two input frequencies—the desired I.F. and the undesired L.O. frequency. As a result the L.O. "chases itself" and is driven outside the loop acquisition range.

Again the spectrum analyzer is a useful tool for measuring the level of VCO pick-up and the degree of improvement that any circuit modification or component relocation makes. A good layout will have symmetrical input leads placed as close together as possible, shielded input coils (where used) and external components mounted as close to the I/C as possible. The DC feedback decoupling capacitor connected between Pins 6 & 9 should be right against the pins. The pcb layout shown later, even though it uses an I/C socket, is able to keep the equivalent VCO input level to under 2 μ Vrms. To put this number in perspective, it is -97 dB compared to the original VCO level. For the measurements, the spectrum analyzer should be connected through a FET probe at the I.F. output, which is disconnected from the detector stage. The VCO control pin is grounded, the detector input is de-coupled with a 0.01 μ F capacitor to ground, and a reference signal CW of the order of 100 μ Vrms is applied at the filter input.

Second and Third Stages

These are easy to handle since they are completely self contained within the LM1823. The maximum gain is fixed at 17 dB each with 26 dB and 20 dB of gain reduction capability respectively.

Fourth Stage

Unlike the preceding stages, the emitters of the fourth differential amplifier are available at Pins 3 & 4. An internal resistance of 1360 Ω between these pins sets the minimum stage gain at 4 dB, and under these conditions (Pins 3 & 4 open) the stage does not provide significant gain reduction with AGC action. However, when an external resistor is connected between the emitters, the gain increases. For Pins 3 & 4 shorted together the gain is as much as 18 dB and the stage can provide up to 14 dB gain reduction with AGC action. Because of the way in which the total I.F. amplifier gain reduction is shared between the stages, the effective gain increase obtained by a resistor between Pins 3 & 4 occurs only for signals below the AGC threshold. After 20 dB of system gain reduction the fourth stage is fixed at 4 dB.

Fifth Stage and I.F. Amplifier Output

The fifth and final I.F. amplifier stage has a single-ended output. There is no internal connection to the detector stage, permitting convenient isolation of the IF amplifier and detector functions. Pin 1 is also a point at which any additional signal filtering may be applied. A resistive load con-

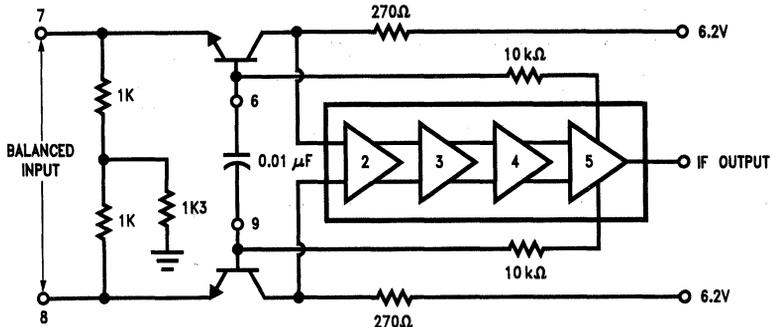


FIGURE 14. Low Impedance Common Base Input Stage

TL/H/8421-14

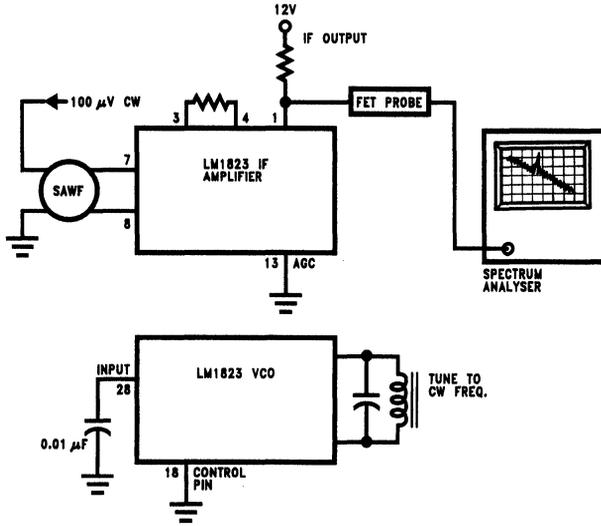
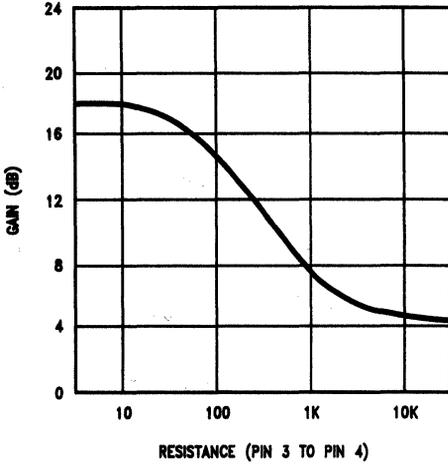


FIGURE 15. Checking the pcb for Excess VCO Pick-up

TL/H/8421-16



RESISTANCE (PIN 3 TO PIN 4)

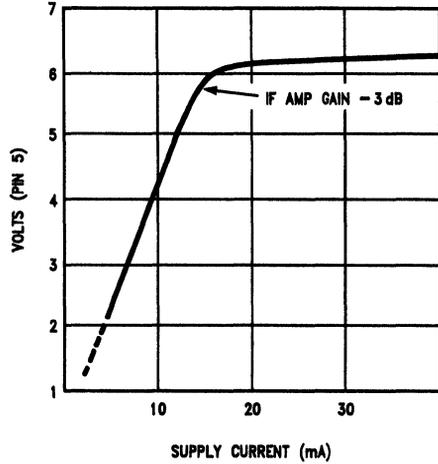
TL/H/8421-17

FIGURE 16. Fourth I.F. Amplifier Stage Gain with External Resistor

connected to the 12V power supply can be used, but the maximum value is limited in practice to less than 500Ω at intermediate frequencies because of stray p.c.b. capacitance and the loading of the detector stage input impedance of 3 kΩ. The stage gain for a total load impedance of Z is given by Equation (2)

$$AV = 1Z/48 \quad (2)$$

The last part of the I.F. amplifier concerns the power supply input at Pin 5. This is a shunt regulated input with a nominal value of 6.3V and the I.F. amplifier current is delivered through a dropping resistor from the 12V rail supplying the remainder of the I/C. The 0.01 μF ceramic r.f. decoupling capacitor at Pin 5 should be grounded through very short



SUPPLY CURRENT (mA)

TL/H/8421-18

FIGURE 17. I.F. Amplifier Voltage Regulator Current Requirement

leads—preferably on the copper side of the p.c.b. A nominal current level into Pin 5 is 32 mA, set by a 180Ω resistor. This current should not exceed 60 mA and the minimum current is about 20 mA, below which the I.F. amplifier will start to lose gain as Pin 5 voltage drops below the regulated level.

SELECTING THE I.F. GAIN

Clearly the LM1823, with all the gain provided by five I.F. amplifier stages and with 34 dB detector conversion gain, has a more than adequate gain margin to provide signal sensitivity and compensate for interstage filter losses. To show how this gain may be distributed we can look at a first cut design example.

If we continue with the 30 dB gain tuner with a 6 dB N.F., using the tuner 75 Ω output to mismaternate the SAWF input will produce a very high insertion loss for the filter. This can easily be over 30 dB but before using the LM1823 gain capability to compensate for this loss, we must look at another aspect of filter insertion loss—the N.F. goes up. Previously we assumed that the tuner N.F. will dominate the system N.F.—and with a tuner amplifier N.F. of 6 dB and 30 dB gain this is indeed true. But when the I.F. amplifier and SAWF are combined the N.F. for the combination exceeds 30 dB. This degrades the system N.F. to 7 dB* and after 50 dB of I.F. amplifier gain reduction the N.F. will be over 8 dB. Frequently this will be alright but it is instructive to consider improving the SAWF N.F. by matching the tuner output impedance to the filter or using an impedance matching pre-amp. For example, the 10 dB gain pre-amp shown in *Figure 18* has a 4 dB N.F. and reduces the filter loss to less than 20 dB. After 50 dB I.F. amplifier gain reduction, the combined N.F. is only 27 dB—for a worst case system N.F. of 6.6 dB. In a dual conversion system with a diode mixer (and already high N.F.), some gain *must* be provided prior to the SAWF.

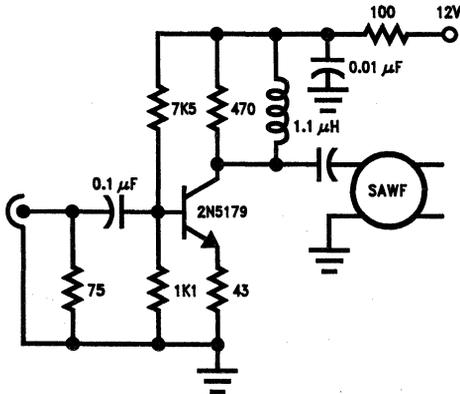
$$*NF_{\text{system}} = NF_{\text{tuner}} + \frac{NF_{\text{IF}}}{(\text{Tuner Gain})}$$

Leaving a 10 dB gain margin over that required to raise a -6 dBmV signal to the rated detector output, the total gain requirement of the I.F. amplifier is

$$75.6 \text{ dB} - 30 \text{ dB} + 30 \text{ dB} - 34 \text{ dB} + 10 \text{ dB} = 51.6 \text{ dB}$$

(0.5 mV \rightarrow 3V) (tuner) (SAWF) (detector) (gain margin)

(With a 10 dB gain impedance matching amplifier between the tuner and the SAWF, the gain requirement falls by 20 dB to 31.6 dB.) To avoid overload in the high gain tuner, we probably have to start gain reducing the tuner when the input signal reaches +10 dBmV (but certainly not before 0 dBmV in order to preserve the tuner NF) so that the I.F. AGC range requirement is approximately 26 dB. This amount of AGC range can be obtained without a resistor connected between Pins 4 & 5 putting the fourth stage gain



TL/H/8421-19

FIGURE 18. Impedance Matching Pre-amplifier

at 4 dB. The SAWF impedance sets the input stage gain at 3 dB for a total of 41 dB to the input of the final stage. A 180 Ω resistor at Pin 1 gives the desired last stage gain of 11 dB, or this resistor is reduced to 50 Ω and a 10 dB pad is

inserted between the I.F. amplifier output and the detector input when a pre-amp is used.

LM1823 VIDEO DETECTOR

The second major function of the LM1823 is the video detector stage, including the AFT/AFC detector and AGC detector/amplifier.

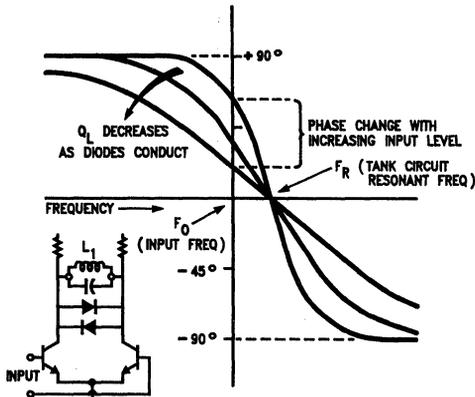
The video detector stage of the LM1823 has a fixed conversion gain of 34 dB—giving a 60 Vrms input level for a 3V (o-p) detected output. This input level is required for AGC action to commence and is well below the input level that can cause intermodulation or catastrophic overload.

Synchronous detection of an amplitude modulated carrier involves a source of constant amplitude CW with the same frequency as the signal carrier, and two phase detectors. One detector is operated in quadrature—i.e. the CW phase and the signal carrier phase have a 90 degree difference at the inputs to the phase detector. This detector operates solely to keep the CW source phase-locked to the signal carrier. The second phase detector has synchronous or in-phase inputs so that the detector output responds to the amplitude difference between the inputs and therefore tracks the signal amplitude modulation.

The benefits of synchronous detection over envelope detection are well known, and most modern receivers incorporate a type of detector known as a quasi-synchronous detector, which is a signal amplitude detector. The I.F. signal is amplified and stripped of modulation in order to be used as the detector CW. The disadvantages of this type of detector are the loss of linearity at very low signal inputs (corresponding to peak video modulation) and a fundamental compromise in the bandwidth of the limiter stage used to strip the modulation. To maintain ease of tuning and a relative immunity from center frequency drift caused by temperature changes and aging, the limiter bandwidth is sufficiently wide that the resulting CW is phase modulated by the information on the original I.F. carrier. Since this can generate intermodulation products, a high Q is desirable and a trade-off in ease of alignment occurs.

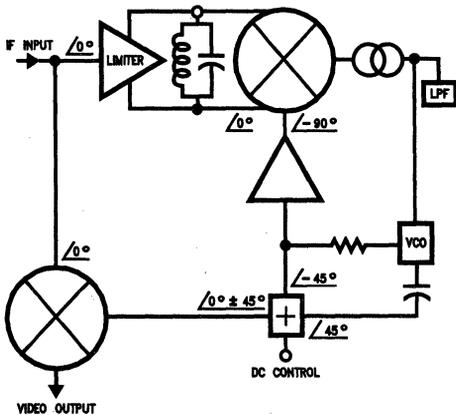
A less obvious problem with this type of detector is the actual static detection phase that is being regenerated. Internal I/C related phase shifts cause the limited carrier waveform applied to the detector to be more or less than 0 degrees phase-shifted with respect to the signal carrier phase. A loss in detector efficiency results, but if the limiter tuning is adjusted to compensate for this, the CW phase from the limiter will depend on the drive to the limiter. The detection phase then changes with amplitude modulation of the original I.F. carrier. The effect of this is observed primarily as differential phase in the chroma subcarrier signal and increased levels of sound buzz. Although, as discussed later, the desired phase difference between the detector CW and signal carrier is not necessarily 0 degrees, the limiter tuning cannot be used to correct the amplitude modulation detector phase—the limiter must be center tuned to avoid carrier phase shifts with modulation level.

The LM1823 overcomes these problems by providing a true synchronous detector system, which, as the block diagram shows, comprises of an internal VCO and in-phase and quadrature phase detectors. The incoming signal from the



TL/H/8421-20

FIGURE 19. Limited I.F. Carrier Phase Shifts with Input Amplitude when the Limiter Tank is Mismatched

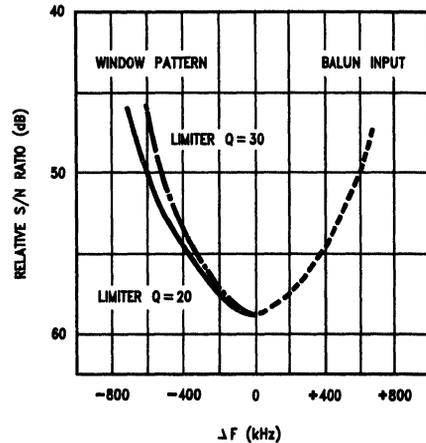


TL/H/8421-21

FIGURE 20. LM1823 Synchronous Detector and DC Controlled Detection Phase

I.F. amplifier is split into two paths. One path is through a high gain limiter stage which strips the amplitude modulation from the CW and applies it to one input of the quadrature phase detector. The other detector input is from the VCO and, once synchronized to the intermediate frequency, if the VCO phase deviates from a 90 degree relationship with the limiter CW phase, a control current is generated by the phase detector and is filtered at Pin 18 to correct the VCO. Even though the limiter stage tuned circuit faces the same compromises of desired narrow bandwidth versus ease and stability of tuning, the filter at Pin 18 can be made to have a very narrow bandwidth. Therefore the VCO can provide a reference signal to the phase detectors with a high degree of spectral purity. The second path for the I.F. signal is directly to the in-phase detector. The VCO output passes through a DC voltage controlled phase shifter before being applied to this detector. The DC phase shifter allows precise adjustment of the synchronized VCO phase for maximum amplitude modulation detection efficiency, and compensates for any internal I/C phase shift variations. At the same time, proper center-tuning of the limiter coil is possible.

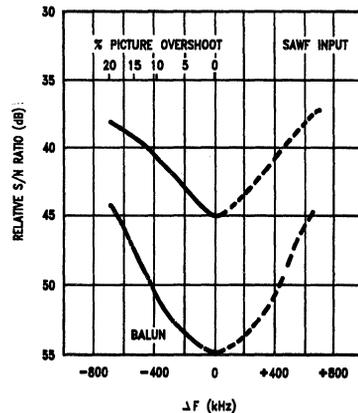
The benefits of center-tuning the limiter are clearly shown by comparing the differential chroma phase of the LM1823



TL/H/8421-22

FIGURE 21. Relative S/N Ratio with Limiter Tuning (No SAWF)

with a conventional quasi-synchronous detector. The LM1823 can consistently produce DP'S of under 1 degree compared with up to 10 degrees for a quasi-synchronous detector. There is also a substantial improvement in the sound carrier S/N ratio. When the limiter is detuned to compensate for internal I/C phase shifts or for detection phase-lags to produce video overshoots (for a subjectively crisper picture), the S/N ratio degrades by 5 dB to 7 dB, depending on the video modulating signal.



TL/H/8421-23

FIGURE 22. Effect of Limiter Retuning with SAWF

THE LIMITER

The limiter tuned circuit at Pins 24 and 25 is driven by a differential stage with a 6.6 kΩ internal load impedance. A small signal gain of 50 (with a tuned circuit dynamic resistance of 8 KΩ) ensures that full quadrature detector efficiency is obtained with input levels above 10 mVrms, and internal Schottky diodes limit the maximum amplitude at Pins 24 and 25 to about 500 mV (p-p). Tuning is achieved either for a peak amplitude signal measured with an F.E.T. probe (low

capacitance) at Pin 24 or Pin 25 with a 10 mVrms CW input, or by monitoring the video detector and adjusting for minimum differential chroma subcarrier phase. The latter adjustment will require a signal source modulated with a chroma/video ramp or stair-step pattern including a 20 IRE level chroma subcarrier, but does have the advantage that the adjustment can be made at strong signal levels, and does not require dis-connection of the tuner.

AFT/AFC CIRCUIT

The AFT phase detector is a doubly-balanced phase detector with the switching signal provided internally from the limiter stage described previously. The quadrature signal input is obtained by light external capacitive coupling from the limiter tuned circuit to the AFT tuned circuit at Pins 23 and 26. Parallel p.c.b. tracks to the limiter and AFT coils will usually provide sufficient coupling and the 1 pF capacitors on the LM1823 test circuit (see LM1823 data sheet) are shown only to illustrate the level of coupling involved. Since the AFT tuned circuit is driving an amplifier with a differential input resistance of 20 kΩ, it is able to operate close to the unloaded Q of the inductor.

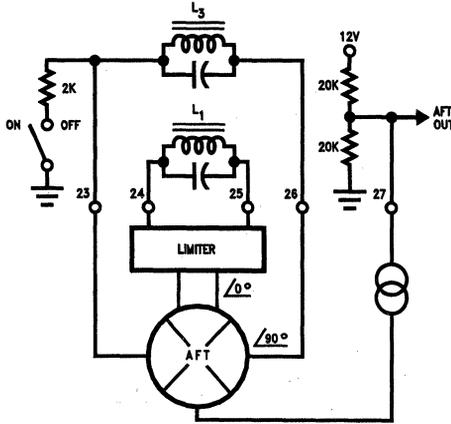


FIGURE 23. AFT Circuit with pcb Coupling Between the Limiter & AFT Tuned Circuits

The AFT output Pin 27 is driven from a current source so that the output voltage at the proper center frequency is set by an external resistive divider network. The parallel resistance of this divider will determine the voltage swing obtained for a given frequency deviation and in combination with the AFT tuned circuit Q, provides a means to adjust the AFT output slope.

Once outside the desired tuning range the AFT output voltage should stay either close to ground (I.F. frequency high) or close to the positive supply voltage (I.F. frequency low). If the voltage moves back towards the center voltage as the signal moves further away from the desired tuning range, then more coupling from the limiter tank may be needed. Grounding Pin 26 through a 2 kΩ resistor will defeat the AFT circuit for receiver fine-tuning purposes. The 2 kΩ provides isolation of the AFT switch & associated cable from the tuned circuit which has a relatively low dynamic resistance of 1.8 kΩ. Resistor values larger than 2 kΩ may prevent the circuit from being defeated, but either Pin 23 or Pin 26 can be grounded directly without damaging the I/C.

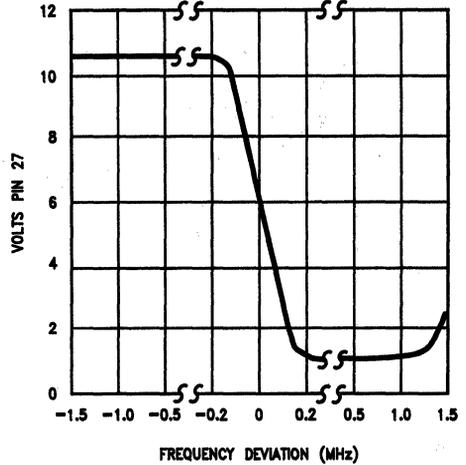


FIGURE 24. AFT Circuit Output Voltage Characteristic (RLOAD at Pin 27 = 10 kΩ)

THE PHASE LOCKED LOOP (PLL)

For true synchronous operation the LM1823 has an internal VCO operating at the video intermediate frequency of 45.75 MHz.

A parallel tuned circuit between Pins 19 and 20 will set the oscillator free-running center frequency and the tuned circuit dynamic resistance is loaded by an internal 1.5 kΩ resistor. Since the oscillator frequency must be controlled, a basic tradeoff exists between oscillator stability, control sensitivity and control range. To obtain a control range of over 2 MHz, the working Q of the tuned circuit should be around 15. Increasing the Q by raising the capacitive arm of the tuned circuit will improve the oscillator stability. This reduces the change in free-running frequency as a result of temperature effects etc. The control sensitivity will decrease correspondingly and there will be a reduction in the control range. The control range in the application circuit has been chosen to cover the expected deviations in the I.F. carrier that are allowed by AFT circuits. With a coil unloaded Qu of

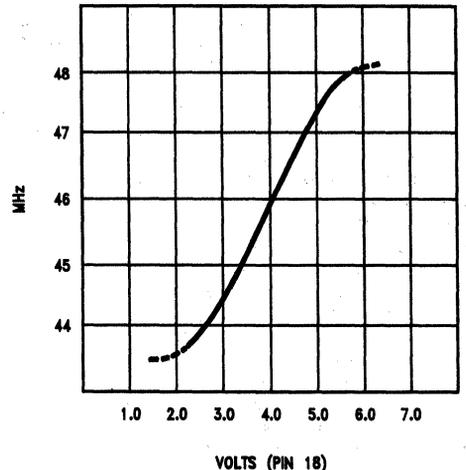


FIGURE 25. VCO Control Sensitivity Characteristic

55, and a working Q of 15, the inductance should be 0.24 uH, which tunes with 51 pF at 45.75 MHz.

The V.C.O. frequency is adjusted by injecting a 60 mVrms CW at Pin 28. If the VCO tuning (L₃) is a long way from being correct, the detector output Pin 16 will show an AC signal of about 4V (p-p) centered around 7.5 VDC. As the oscillator is tuned toward the correct frequency the AC beat note will decrease and abruptly disappear as the oscillator locks to the carrier frequency. Final adjustment of the VCO is done by tuning L₃ until the voltage at the phase detector filter Pin 18 is 4 VDC.

Oscillator control is accomplished by internally phase shifting the currents in a direct cross-coupled differential stage in response to the control voltage developed at Pin 18. Direct cross-coupling of the bases and collectors of this differential stage means that the transistors are operating in a soft-saturated mode, enabling a constant output amplitude to be obtained of about 500 mV (p-p). This output amplitude does not change with coil tuning or over the frequency control range of the oscillator. With the specified tuning components at Pins 19 and 20, the VCO sensitivity is 1.5 MHz/volt. Other general characteristics of the VCO are a negative temperature coefficient of 150 ppm/degree C, and a tendency for the oscillator control sensitivity to decrease with decreasing frequency of operation (below 10 MHz).

The VCO tuning components are mounted across the I/C package from the I.F. amplifier input. This minimizes inductive coupling and yields approximately 105 dB isolation for the I/C alone. Leads and components connected to the I.F. amplifier input will reduce the VCO isolation (as will higher operating frequencies).

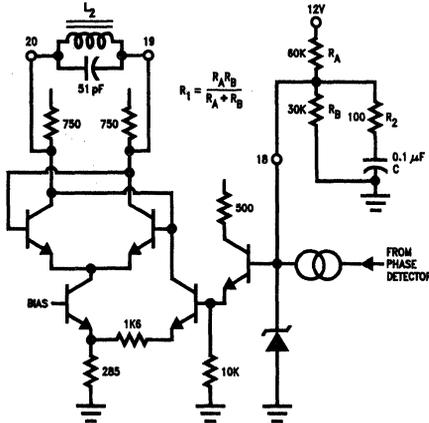


FIGURE 26. LM1823 VCO Circuit

The quadrature phase detector output is a push-pull current source so that the control voltage at Pin 18 is determined by the parallel resistance of the external divider network, which also sets the quiescent control voltage in the absence of an I.F. signal. This divider voltage should be centered at 4 VDC since the lower voltage swing for controlling the oscillator frequency is 2 VDC, and an internal clamp prevents Pin 18 increasing above 5.6 VDC. By using a 20 kΩ parallel resistance at Pin 18, the phase detector current of 7.5 uA/degree gives a phase detector sensitivity (μ) of 0.15 volts/degree. This parallel resistance is equivalent to R₁ in the conventional filter for a 2nd order PLL. The oscillator and phase detector sensitivities given above yield a DC loop gain of 12.9 MHz/radian. For the data sheet value of 100Ω for R₂,

and a filter capacitor of 0.1 uF, the loop damping factor (K) is 1.01 and the natural resonant frequency (ω_n) is 32 kHz. From this we can calculate that the loop -3 dB bandwidth is 73 kHz which is substantially less than would be practicable with a quasi-synchronous detection system, and this brings the desired benefits of low luma/sound/chroma crosstalk and freedom from quadrature distortion produced by the I.F. filter slope characteristic in the vicinity of the picture carrier frequency. Nevertheless, some signal conditions may cause wider PLL bandwidths to be used. A probable problem is incidental carrier phase modulation (ICPM).

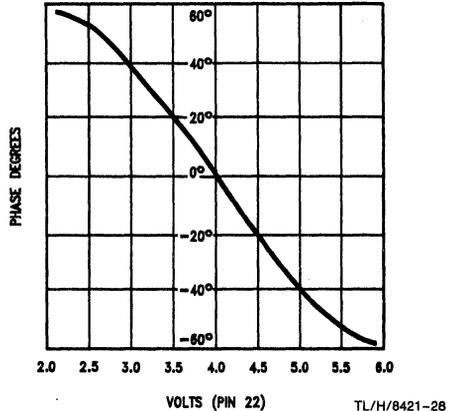


FIGURE 27. DC Controlled Phase Shifter Characteristic

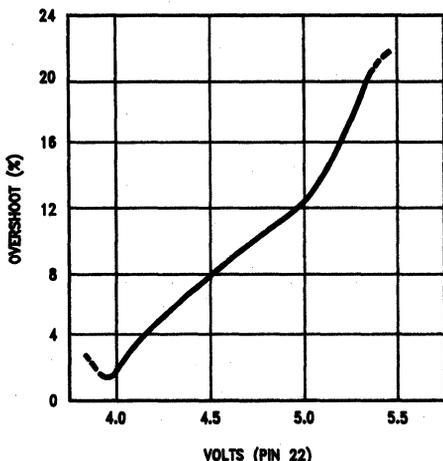
This describes the shift in carrier phase as the modulation depth changes, and is particularly likely to happen where prior processing of the original carrier waveform has occurred—in distribution or conversion amplifiers employed in MATV and cable systems for example. It is also present to an extent in broadcast transmitters and if the PLL loop bandwidth is too narrow for the VCO to track this phase shift, then the ICPM is transferred to the signal modulation. This can be observed as a tint shift in color bars or a smear

PHASE LOCKED LOOP PARAMETERS		
<p>$R_1 = R_A \parallel R_B$</p>	<p>VCO SENSITIVITY (β) = 1.5 MHz/VOLT</p> <p>PHASE DETECTOR SENSITIVITY (μ) = 0.15V/DEGREE</p>	
<p>DC LOOP GAIN ($\mu\beta$) = $\frac{W_C}{2\pi}$</p> <p>LOOP NATURAL FREQUENCY (F_n) = $\frac{W_n}{2\pi}$</p>		
$W_n = \sqrt{\frac{W_C}{C(R_1 + R_2)}}$		
<p>LOOP DAMPING FACTOR (K) = $\frac{R_2}{2} \sqrt{\frac{C W_C}{R_1}}$</p>		
<p>LOOP -3dB FREQUENCY = $\frac{W_n}{2\pi} [2K^2 + 1 + (2K^2 + 1)^2 + 1]^{1/2}$ or $\frac{W_n 2K}{\pi}$ (K > 2)</p>		
<p>$W_C = 81 \times 10^6$ RADS/SEC</p> <p>$W_n = 2 \times 10^5$ RADS/SEC</p>	<p>$R_2 = 100\Omega$</p> <p>K = 1.01</p> <p>$F_{3dB} = 36$ kHz</p>	<p>$R_2 = 680\Omega$</p> <p>K = 69</p> <p>$F_{3dB} = 440$ kHz</p>

TL/H/8421-29

in the leading edge of a color bar as the VCO belatedly attempts to track the phase change. For these types of signals it is desirable to increase the loop bandwidth to about 500 kHz—changing R2 to 680 Ω is an easy fix. The loop damping factor is kept greater than 1 to avoid ringing on the phase transients. Larger loop bandwidths will increase the possibility of luma/sound/chroma crosstalk.

Once the VCO is locked in phase to the I.F. signal, the DC phase shifter Pin 22 is normally around 4 VDC for peak detector efficiency. Usually some extra phase lag will be introduced since a subjectively crisper picture is obtained if picture transients have an overshoot. Between 12% and 20% overshoot without ringing is desirable, corresponding to a 400 mV to 800 mV shift in Pin 22 voltage.



TL/H/8421-30

FIGURE 28. Signal Overshoot Produced by Carrier Detection Phase Shift

VIDEO DETECTOR POST AMPLIFIER

The response of the video amplifier is rolled off above 9 MHz to minimize the amount of the VCO waveform and its harmonics appearing in the output at Pin 16. Typical oscillator products are 40 dB below the desired signal level.

Zener diodes are used in the video amplifiers for level shifting so that the use of PNP transistors is avoided and the detector linearity is preserved. Excellent differential gain characteristics are obtained—typically less than 3%. Pin 16 is a Darlington NPN emitter follower output. With no detector CW input signal, Pin 16 is at 7.6 VDC, representing zero carrier level which is slightly higher than peak white (by 12½%). As the CW input increases, Pin 16 voltage decreases towards black level with the sync pulses producing the most negative detector level.

The level reached by the sync tips is determined by the AGC loop threshold and if the internal AGC comparator is used (Pin 16 is directly connected to Pin 17), the sync tips will be clamped at 4 VDC. This produces a nominal detector output of 3.2V (p-p) but this is subject to variations in the Pin 16 detected zero carrier level. The resistive network shown connected between Pin 16 and Pin 17 in Figure 30 can be used to change the zero carrier level at Pin 17 for an adjustable recovered video level. For best performance the recovered video level should never be less than 1V (p-p) or greater than 4V (p-p). In suppressed sync systems, the recovered video at Pin 16 is routed to the descrambler for restoration of the sync amplitude before it is applied to Pin 17. Obviously the signal DC content must be preserved through the descrambler if proper AGC action is to be maintained.

AGC Self Gating Comparator (LM1823)

The AGC comparator input has a low pass filter to protect the AGC loop from noise interference. Conventional detector systems often use noise gates to prevent the AGC system "backing off" on noise peaks that occur below the sync tip level. It is difficult to set the noise gate threshold close enough to the sync tip level for it to provide any benefit without risking AGC lock-out. For the LM1823 however, syn-

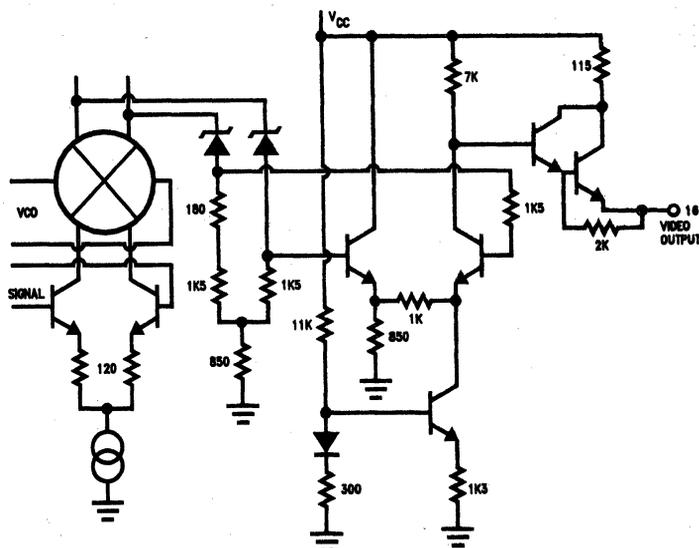


FIGURE 29. LM1823 Detector and Video Amplifier

TL/H/8421-31

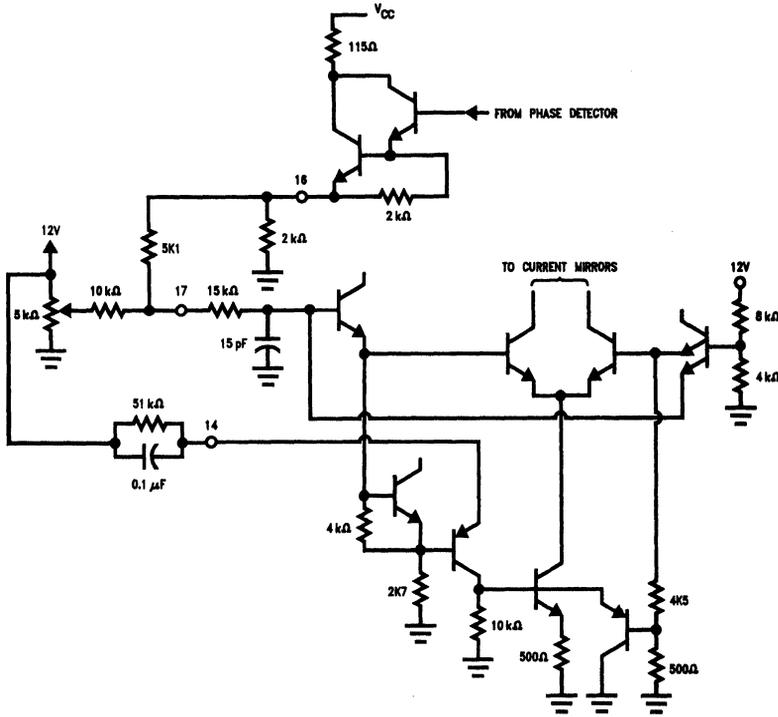


FIGURE 30. LM1823 Self Gating AGC Comparator

TL/H/8421-32

chronous detection allows the noise gate to be eliminated. Since the noise is random phase, the synchronous detector will not rectify the noise voltage and the low pass filter can average out the noise input to the comparator.

Further protection of the AGC comparator is provided by gating the comparator on only during the sync pulse period. The gate pulse is obtained from the input video waveform sync pulses at Pin 16. Essentially an emitter coupled sync stripper circuit, the slice level is set by an external time con-

stant at Pin 14. During the sync pulse period the capacitor at Pin 14 is being charged toward ground potential and the comparator is gated on. Between sync pulses the capacitor discharges towards the positive supply voltage through the resistor and the comparator is off. The sync slice level is determined by the Pin 14 RC time constant and is given in Equation (3) as the number of millivolts the slice level is above the sync tip voltage.

$$V_{SLICE} = 1/2 RC (mV) \tag{3}$$

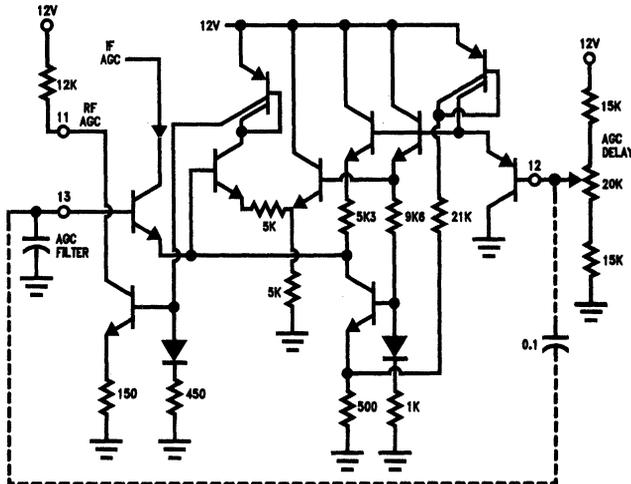


FIGURE 31. RF AGC Amplifier

TL/H/8421-33

A typical slice level for a 3V (o-p) video signal is between 100 mV and 250 mV. Different slice levels can be obtained with other capacitor values (the resistor should be left unchanged). Small capacitors will allow a faster response to a fluctuating sync tip level but also may cause the consequently deeper slice to include video overshoots.

RE AGC DELAY AND OUTPUT AMPLIFIER

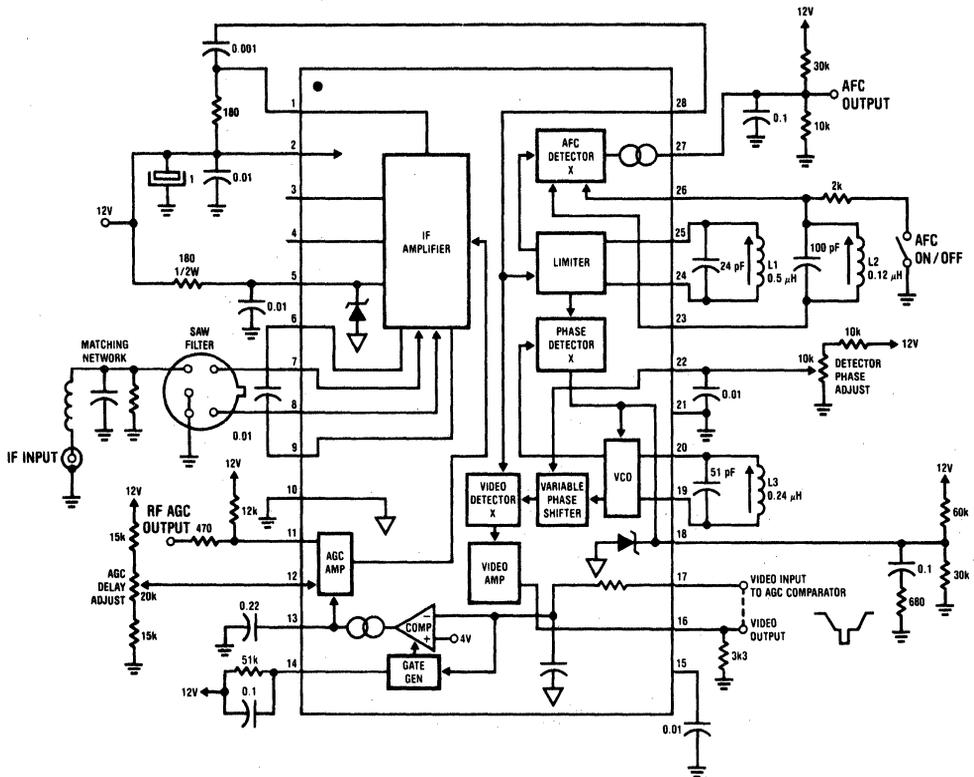
The I.F. amplifier is at full gain below 4 VDC on Pin 13. At anywhere from 5.5 VDC to 6.5 VDC we will want to shift gain control into the R.F. stages, and this is accomplished by a delayed AGC threshold control at Pin 12. When the filter voltage on Pin 13 is 0.7V above the pre-set level on Pin 12, the R.F. AGC amplifier at Pin 11 will start to sink current.

The capacitor shown connected between Pins 12 and 13 is optional and intended to provide an increase in AGC action

for signal amplitude transients at high R.F. signal levels (tuner in gain reduction). AC changes on Pin 13 are coupled to the threshold level control allowing the I.F. amplifier to gain reduce (or increase) during the signal transient. This happens only during the signal change, so that the detected video returns more rapidly to the proper output levels. Once signal equilibrium is restored, the appropriate gain balance between the R.F. and I.F. amplifiers returns.

CONCLUSION

This note has described a high quality video I.F. amplifier/detector combination that can provide excellent baseband video signals. A complete schematic of the external components required in such an application is shown in *Figure 32*, with a suitable p.c.b. layout in *Figure 33*.



SAW Filter-MuRata SAF45MC/MA

L1-9 $\frac{1}{2}$ T } #22 wire

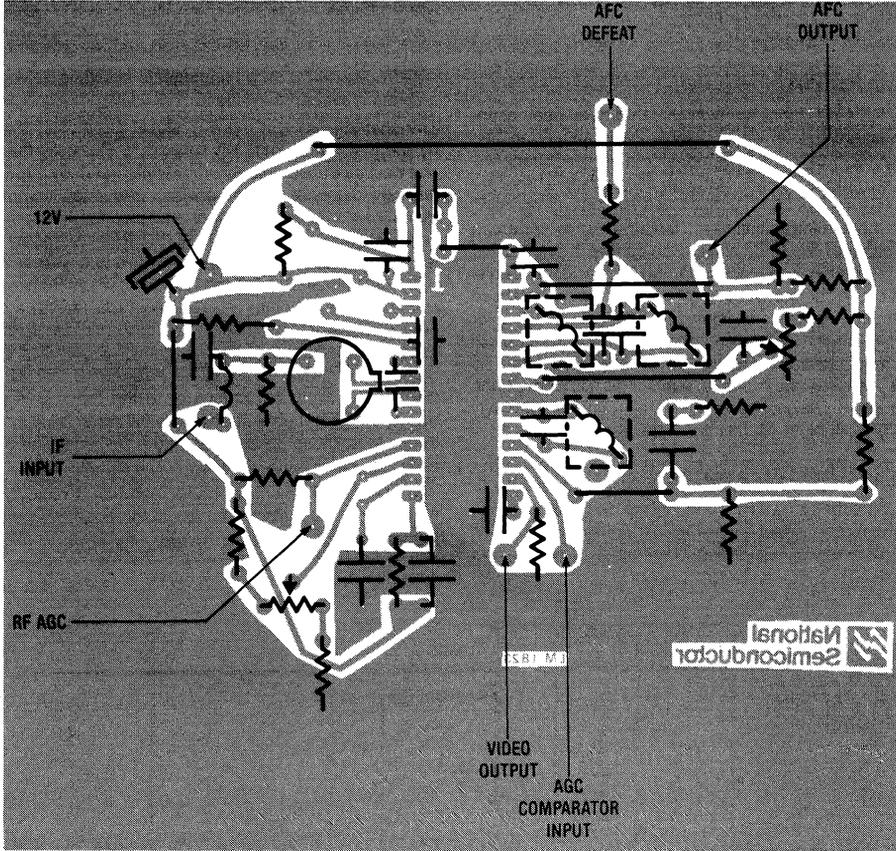
L2-4 $\frac{1}{2}$ T } on 3.16" form with

L3-6 $\frac{1}{2}$ T } HF core, shielded

All caps in μ F unless noted

FIGURE 32. LM1823 External Circuit Components

TL/H/8421-34



TL/H/8421-35

FIGURE 33. LM1823 Printed Circuit Board Layout (Component Side)

LM2889 R.F. Modulator

National Semiconductor
Application Note 402
Martin Giles



Introduction

Two I/C RF modulators are available that have been especially designed to convert a suitable baseband video and audio signal up to a low VHF modulated carrier (Channel 2 through 6 in the U.S., and 1 through 3 in Japan). These are the LM1889 and LM2889. Both I/C's are identical regarding the R.F. modulation function—including pin-outs—and can provide either of two R.F. carriers with dc switch selection of the desired carrier frequency. The LM1889 includes a crystal controlled chroma subcarrier oscillator and balanced modulators for encoding (R-Y) and (B-Y) or (U) and (V) color difference signals. A sound intercarrier frequency L-C oscillator is modulated using an external varactor diode. The LM2889 replaces the chroma subcarrier function of the LM1889 with a video dc restoration clamp and an internally frequency modulated sound intercarrier oscillator.

Modulation Parameters

In the U.S., either of two R.F. channels is made available so that the user can select a vacant channel allocation in his geographic area, thus avoiding co-channel problems with

older receivers that have inadequate shielding between the antenna input and the tuner.

The characteristics of the R.F. signal are loosely regulated by the FCC under part 15, subpart H. Basically the signal can occupy the standard T.V. channel bandwidth of 6 MHz, and any spurious (or otherwise) frequency components more than 3 MHz away from the channel limits must be suppressed by more than -30 dB from the peak carrier level. The peak carrier power is limited to 3 mVrms in 75Ω or 6 mVrms in 300Ω, and the R.F. signal must be hard-wired to the receiver through a cable. Most receivers are able to provide noise-free pictures when the antenna signal level exceeds 1 mVrms and so our goal will be to have an R.F. output level above 1 mVrms but less than 3 mVrms. Since the distance from the converter to the receiver is usually only a few feet, cable attenuation will rarely be a problem, but mis-termination can change both the amplitude and relative frequency characteristics of the signal.

The standard T.V. channel spectrum has a picture carrier located 1.25 MHz from the lower band edge. This carrier is amplitude modulated by the video and sync signal. In the

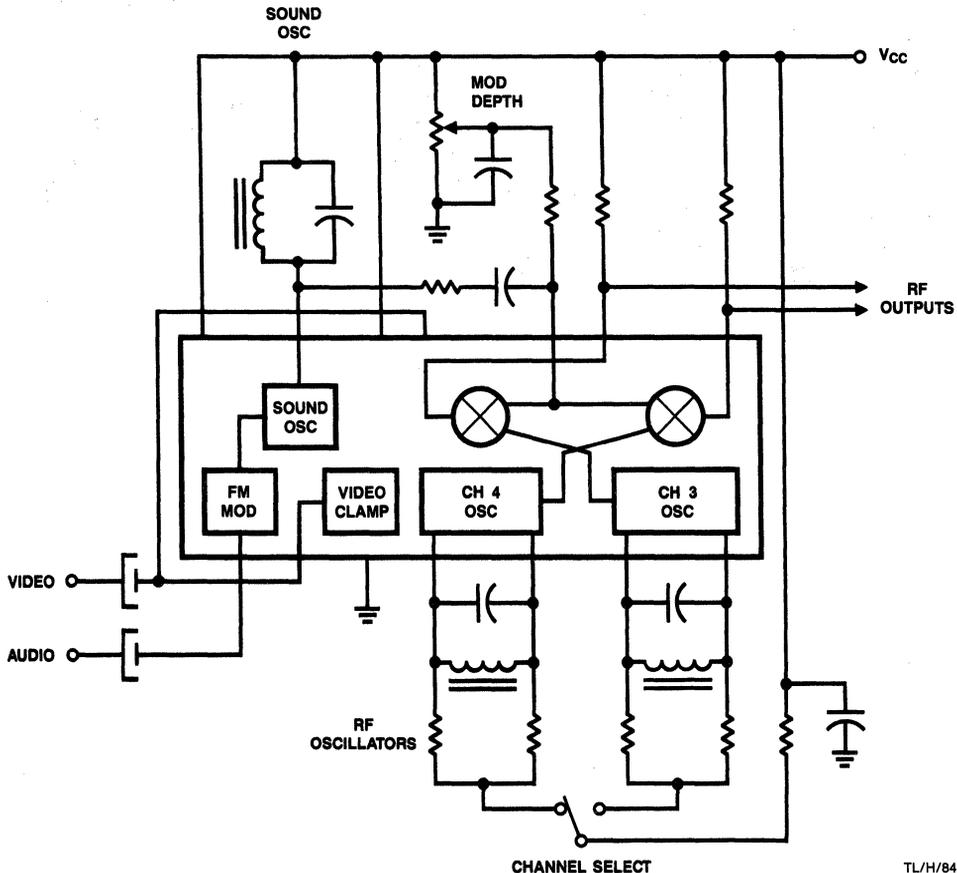
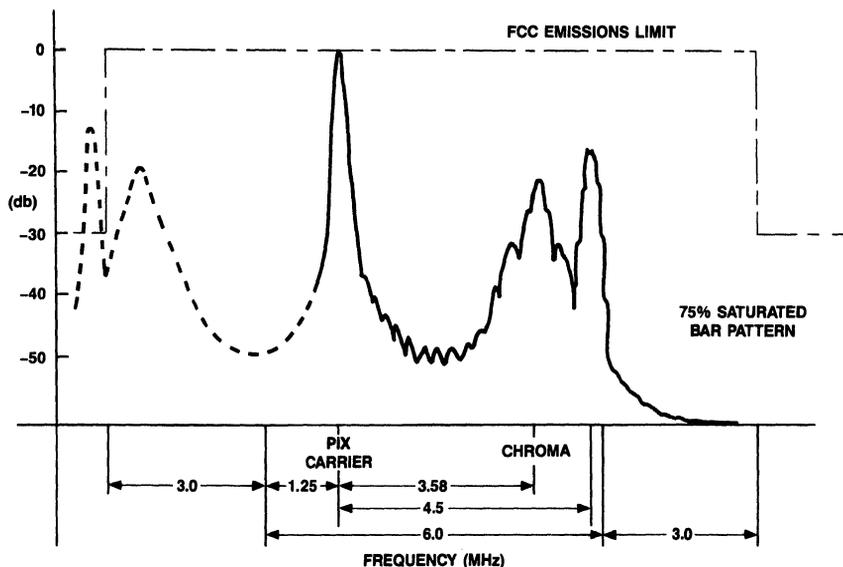


FIGURE 1. The LM2889 Block Diagram With External Components

TL/H/8452-1

Modulation Parameters (Continued)



TL/H/8452-2

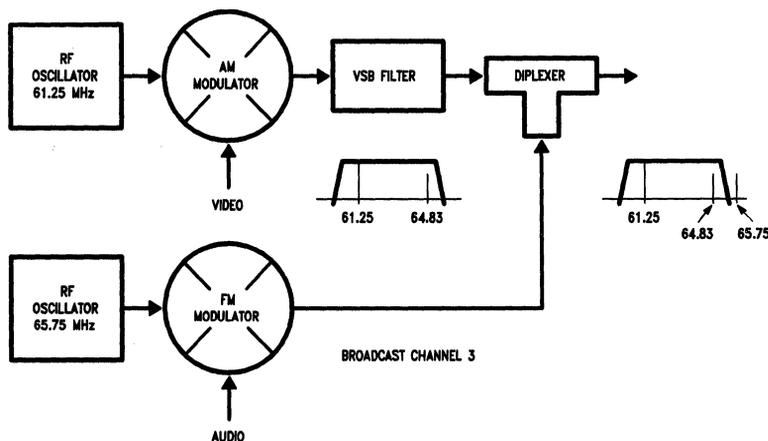
FIGURE 2. Television Channel R.F. Spectrum

case of a color signal, a second subcarrier is added 3.58 MHz above the picture carrier. The sound or aural carrier is 4.5 MHz above the picture carrier and is frequency modulated with the audio signal to a peak deviation of 25 kHz. This audio signal has pre-emphasis above 2.1 kHz (a 75 μ s time constant). Similar modulation methods and standards are used in Japan and Europe.

With the picture carrier located near one end of the channel bandwidth, most of the available spectrum is used by the upper sideband modulation components. Only modulating frequencies within 0.75 MHz of the carrier frequency are transmitted double sideband and the lower sideband is truncated by at least -20 dB compared to the peak carrier level by the time the lower channel edge is reached. This is referred to as Vestigial Sideband (VSB) modulation and since most R.F. modulators are double sideband, a VSB filter is used at the transmitter output. A filter is needed for each

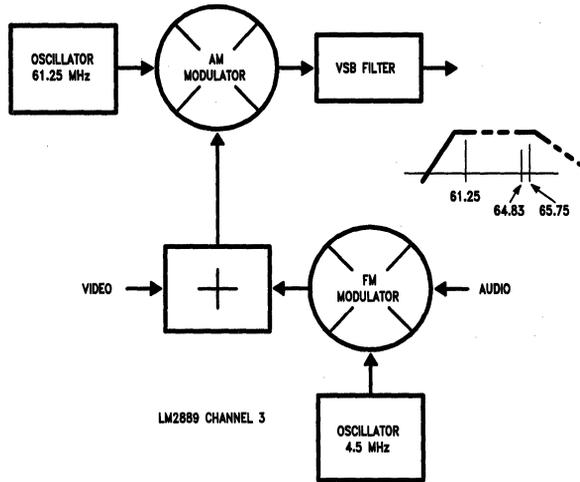
channel and consists of bandpass and harmonic filter sections. A broadcast transmitter uses a separate modulator for the sound carrier and this is added to the picture carrier via a diplexer before reaching the transmitting antenna. Close control is maintained on the picture and sound carrier frequencies to keep a 4.5 MHz spacing between them. This tight frequency control is used to advantage by the majority of television receivers which employ intercarrier sound circuits. The I.F. amplifier processes both the picture and sound I.F. carriers and detects the 4.5 MHz difference frequency at the video detector stage. This frequency modulated sound intercarrier is then stripped of amplitude modulation by a high gain limiter circuit and a quadrature demodulator recovers the audio.

The LM1889 and LM2889 use a slightly different modulation scheme to that described above for several reasons. For circuit economy L-C oscillators are used to generate the picture



TL/H/8452-3

FIGURE 3. Broadcast Transmitter Block Diagram



TL/H/8452-4

FIGURE 4. LM1889/2889 Sound and Video Modulation

carrier frequencies. The stability of such oscillators is good enough for the AFT circuits in modern receivers to maintain picture quality, but if a separate L-C sound carrier oscillator were used, the relative drift of the two carrier frequencies would be much too great for intercarrier sound receivers. For example, a typical television sound circuit tuned to 4.5 MHz will generate as much as 3% distortion if the difference between the R.F. carriers changes by 15 kHz. Apart from the difficulty of setting the initial frequency with sufficient accuracy, it is unlikely that two L-C oscillators could be kept within 15 kHz of each other at 60 MHz to 100 MHz operating frequencies. However, when the audio signal is modulated onto a 4.5 MHz intercarrier oscillator frequency and this carrier is used to modulate the picture carrier, we have only the 4.5 MHz oscillator drift to worry about.

A less obvious problem, but nevertheless significant if good audio quality is to be obtained, is incidental carrier phase modulation (ICPM). Even broadcast transmitters cannot maintain an invariant carrier phase as the modulation depth changes. Without feedback loops to control ICPM, a broadcast transmitter can produce from 3 degrees to as much as 30 degrees phase change as the carrier modulation decreases from sync tips to peak white. While the separate sound carrier is unaffected by this ICPM of the pix carrier, on reception in the intercarrier sound receiver the phase shift with picture information is transferred onto the 4.5 MHz sound intercarrier. This results in a phenomenon known as sound buzz. Even with exceptionally careful p.c.b. layout, an I/C modulator with L-C oscillators can expect the pix carrier frequency to change with modulation depth. Fortunately, by modulating the sound signal as a 4.5 MHz intercarrier onto the pix carrier, the ICPM occurs equally in both R.F. carriers and will not be detected by the intercarrier receiver.

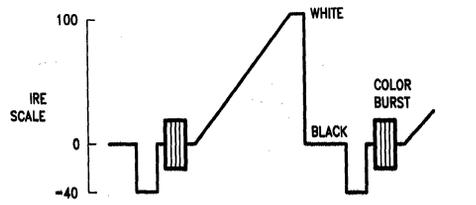
Video Modulation

The baseband input to the modulator is in an easily recognized composite format and this is a convenient point at which to introduce the I.R.E. scale. This is an oscilloscope scale divided into 140 units. The video portion of the signal representing the scene (picture) brightness levels will occupy the 0 to 100 I.R.E. portion of the scale, with 0 I.R.E. as black level and 100 I.R.E. as peak white level. From 0 to

-40 I.R.E. is the synchronization portion of the signal. The usefulness of this scale is that the standard composite video signal will always have a sync amplitude that can be normalized to 40 I.R.E. Similarly the color burst amplitude is always 40 I.R.E. For a 1V (p-p) video signal, an I.R.E. unit is equivalent to 7.5 mV.

Although the video is amplitude modulated on the carrier waveform, the carrier amplitude only decreases from the unmodulated level. This contrasts with standard AM where the carrier level alternately increases and decreases about the unmodulated level. For a television signal, the peak unmodulated level corresponds to sync tip level and increasing brightness levels cause decreasing carrier levels. To prevent complete suppression of the carrier (and consequent loss of the sound intercarrier in the receiver) the peak white signal is limited to a maximum modulation depth of 87.5% of the peak carrier. Returning to our I.R.E. scale we can see that from peak carrier to zero carrier is equivalent to 160 I.R.E. ($140/0.875 = 160$). One obvious consequence of this modulation scheme is that the video signal MUST BE dc coupled to the modulator. AC coupling will cause the peak carrier level to change with modulation scene brightness (standard AM) and the sync modulation amplitude will change. This spells trouble for the receiver sync circuits and the changing R.F. carrier black level will cause errors in displayed brightness—the picture will “wash out” or disappear into black.

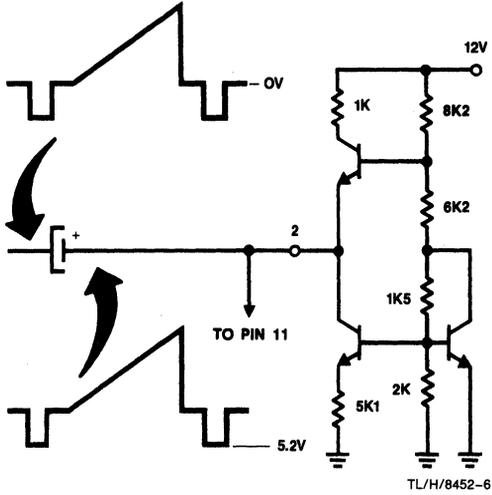
The LM2889 uses doubly balanced modulator circuits with an L-C oscillator switching the upper transistor pairs. The signal is applied across the lower transistor pairs. If the signal input pins 10 and 11 are at the same dc potential, the



TL/H/8452-5

FIGURE 5. Video Modulating Signal (In terms of the I.R.E. Scale)

Video Modulation (Continued)



TL/H/8452-6

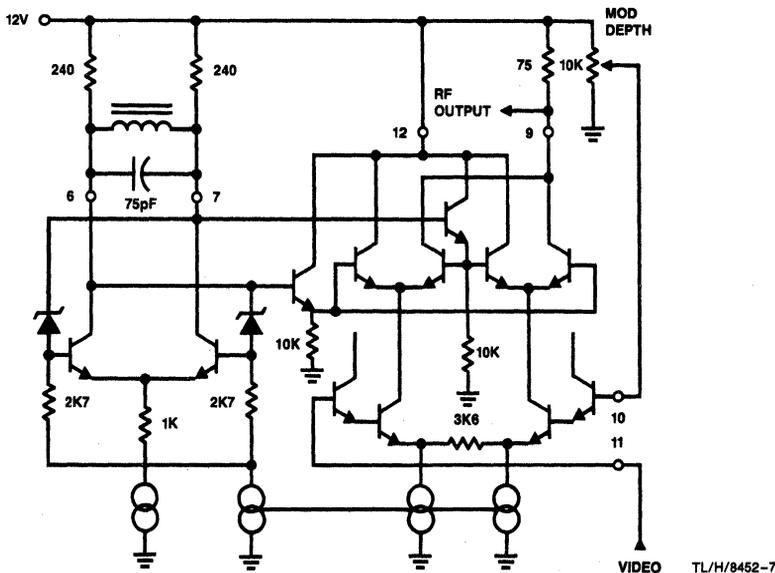
FIGURE 6. LM2889 Video DC Restoration Clamp

carrier is completely suppressed. As the offset voltage between pins 10 & 11 is increased, the carrier output level increases. With a 75Ω output load resistor, the conversion gain of the R.F. modulator is 20 mVrms/volt. A dc restoration circuit at pin 2 of the LM2889 allows the composite video to be ac coupled from the preceding stages, giving the designer flexibility in the video processing circuits (unless an LM1886 is being used as a video source, it is unlikely that the composite video dc level will be correct, even with dc coupled video sources). On a 12V supply, pin 2 clamps the sync tip of the video waveform to 5.1 VDC. Therefore, if we have a 2V (p-p) signal, one I.R.E. is equivalent to 14.3 mV and 160 I.R.E. is 2.29V. This is the required offset across the modulator input pins and since pin 11 will

be clamped to 5.1 VDC by the dc restorer circuit, pin 10 should be biased at $5.1V + 2.29V = 7.4$ VDC. A look at the R.F. carrier output will confirm that now the syncs occupy from 100% to 75% of the peak carrier, and that white modulates the carrier down to 12½% of the peak. To maintain the proper modulation depth the clamp at pin 2 will track with supply voltage changes, allowing the bias at pin 10 to be set with a resistive divider connected between the supply and ground.

If the video signal polarity is reversed with positive syncs, either a dc coupled signal or an external dc restorer should be used that places the signal sync tip voltage towards the upper end of the common-mode input range at pin 11, which is 9 VDC with a 12V supply. Pin 10 is then offset below pin 11 voltage by the required amount for proper modulation. An input level of 2V (p-p) is optimal. Signal amplitudes of less than 1V (p-p) are also useable but internal offset voltages and the potential for carrier feedthrough or leakage to the output stage may make it difficult to maintain good R.F. linearity at peak modulation depths. Signal swings larger than 3V (p-p) should be avoided since this will produce relatively large AC/DC current ratios in the modulator and the resulting modulator non-linearities can cause a 920 kHz beat between the chroma and sound carriers.

Although only one video input is required, the LM2889 has two balanced R.F. modulators and two R.F. carrier frequency oscillators. Selection of the carrier frequency is by dc switching the supply voltage to the relevant oscillator tuned circuit. This automatically shuts off the other oscillator and modulator circuits. For test purposes when an output R.F. VSB filter isn't used, or when only one carrier frequency is needed, the output pins 8 and 9 can be wired together with a common load resistor. Providing two channel operation with two independent oscillator/modulator circuits is much superior to using a single modulator and attempting to change carrier frequency by switching the tuning components of a single L-C oscillator. The latter method involves



TL/H/8452-7

FIGURE 7. LM2889 R.F. Modulator and Oscillator (one channel)

Video Modulation (Continued)

use of isolating diodes (if unbalanced operation with attendant feed through problems is to be avoided) and expensive trimmer capacitors for tuning the second carrier frequency. A further disadvantage is the need to switch the VSB filter at the R.F. output.

The LM2889 oscillator configuration is the familiar cross coupled differential amplifier type, with level shifting zener diodes used to prevent the transistors from saturating with large oscillator output swings. The oscillator frequency is set by the tuned circuit components ($f = 1/2\pi\sqrt{LC}$), and the load resistors connected to the supply will set the oscillation amplitude and drive level to the modulators as well as determining the circuit working Q.

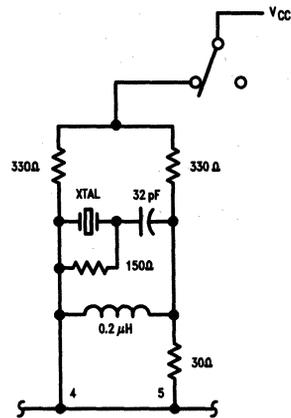
As might be expected, there are conflicting requirements on the practical range of working Q's. A high Q is desirable from the viewpoint of stability, but higher working Q's (set mainly by larger load resistors) increase the drive level to the modulator. Above 350 mV (p-p) the modulator will have attained full conversion gain and the R.F. output level will be determined by the amplitude of the video input signal. Unfortunately increased drive levels will also increase the carrier frequency second harmonic output from the modulator. Although a fully balanced design is used, parasitic capacitances on the emitters of the switching transistor pairs will rectify the oscillator waveform and this produces high levels of second harmonic. Load resistors much larger than 240Ω can produce a level of second harmonic matching the fundamental. Since relatively small load resistors are required (much smaller than the tuned circuit dynamic resistance) the working Q will be dominated by these resistors.

The acceptable degree of frequency stability will depend on the intended application, but L-C oscillators have proven to be adequate for most purposes. We can gain an idea of the frequency stability that is possible by considering the frequency drift produced by changes in the oscillator internal phase. A change in internal phase shift can be caused either by temperature or supply voltage changes but, as the LM2889 data sheet shows, the supply voltage dependency is low. Between 12V and 15V the frequency is essentially constant and changes by less than 30 kHz over the entire supply voltage range. With temperature, the internal oscillator phase shift changes by about 2 degrees over a 50 degree Celsius temperature range. If the tuned circuit Q is 15, then at 61.25 MHz (Ch 3 pix carrier) the oscillator frequency must change by -92 kHz to produce a compensating 2 degree phase shift. If the Q is 30, then the frequency would change by less than -45 kHz etc.

For high circuit Q, a large capacitance is desirable, but the inductor cannot be made too small if it is to remain the tuning element. This keeps the practical range of capacitance values to between 50 pF and 100 pF. Using a 75 pF capacitance, at 67.25 MHz the required inductance is just under 0.08 μH and the working Q is 15 with 240Ω resistors connected on either side of the tuned circuit to the supply voltage. Depending on the coil type, the number of turns for this inductance will be from 1½ to 3½ giving over 10 MHz tuning range. This is more than enough to compensate for component tolerance and variations in overall internal phase lag from 1/C to 1/C.

If better frequency stability of the carrier frequency over that provided by an L/C circuit is needed, then crystal control of the oscillators can be used. It is necessary to retain the inductor, since a dc short is required across the oscillator pins to avoid a collector current imbalance off-setting the

oscillator differential pair and preventing start-up. The inductor value is chosen to resonate with the capacitor in series with the crystal at slightly less than the desired operating frequency. About 20% less will allow the inductor to be fixed tuned. Close to its series resonant frequency (normally the 3rd overtone) the crystal will provide the additional inductive reactance necessary for the circuit to oscillate. The equivalent resistance of the crystal at the operating frequency will affect the tuned circuit Q and hence the peak-to-peak drive to the modulator circuit. Smaller capacitors in series with the crystal (with corresponding changes in the inductor value) will push the operating frequency closer to anti-resonance and produce large equivalent resistances dropping the oscillator drive level. Larger capacitance values cause the operating frequency to approach series resonance and a lower equivalent resistance (approaching R_S for the crystal, which is of the order of 40Ω to 100Ω at 60 MHz). This can produce higher drive levels but risks operation at the lower overtones. To prevent lower frequency oscillation a resistor can be connected across the crystal. Also a small resistor in series with one of the collector leads will form a low pass filter with the output capacitance and suppress spurious oscillations at higher frequencies. If this is needed, resistor values less than 30Ω should be used, so that dc offsets will not prevent the oscillator from starting. For the circuit of Figure 8, capacitor values between 20 pF and 56 pF, with the appropriate inductor value, work well with only slightly reduced oscillator drive compared to the conventional L/C circuit.



TL/H/8452-8

FIGURE 8. R.F. Crystal Oscillator Circuit

The Sound Carrier Oscillator

Before moving to the R.F. output and the VSB requirements, we need to look at another signal that will be added to the baseband video—the aural intercarrier. Both the LM1889 and the LM2889 have L-C sound carrier oscillators operating at 4.5 MHz. Frequency modulation of the LM1889 sound oscillator is achieved by an external varactor diode which alters the tuning capacitance in response to the amplitude of the audio signal. The LM2889 has a similar tuned L-C oscillator but the frequency deviation is obtained by internally phase shifting the oscillator current. This is done by a low pass filter connected to the oscillator which provides a lagging phase voltage component of the oscillator waveform at the input to a differential amplifier. The current output from

Sound Modulation

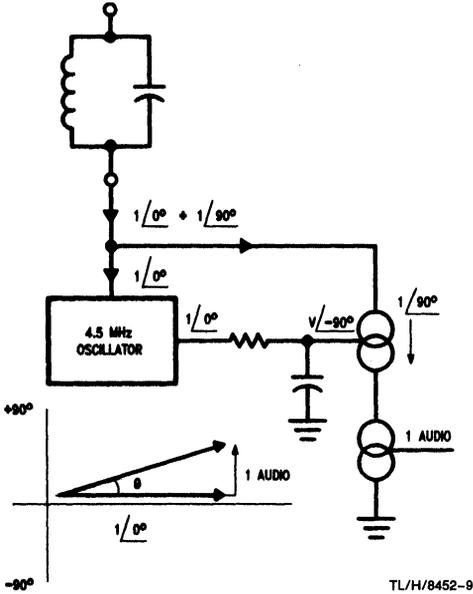


FIGURE 9. LM2889 Sound Carrier FM Modulator

this amplifier is controlled by the audio signal amplitude so that more or less of the current (now in quadrature to the original oscillator current) is added back to the tuned circuit producing the desired shift in the output frequency. Phase offsets of up to +12 degrees with increasing audio input levels will yield very low audio distortion (less than 0.2%). Also the use of a lagging oscillator waveform component reduces harmonic levels within the oscillator and a reduced possibility for undesired signals contaminating the R.F. waveform.

The tuned circuit operating Q is important in two respects. Similar to the R.F. oscillator tuned circuits, the 4.5 MHz tuned circuit should have a high loaded Q for stability, but the circuit bandwidth must also be wide enough to accommodate the FM sidebands produced by the audio modulation. For a maximum frequency deviation (Δf) and maximum modulating frequency f , the minimum bandwidth is given by Equation (1).

$$B-W \geq \Delta f (2.5 + 4f/\Delta f) \quad (1)$$

The other requirement is that the maximum phase deviation of the oscillator current is able to produce the maximum frequency deviation (Δf) of the carrier. This is given by Equation (2).

$$\Delta f = 4.5 \times 10^6 \times 0.12 / Q \quad (2)$$

Table I summarizes the results of calculating the maximum circuit Q that satisfies Equations (1) and (2) for the various monaural sound modulating standards used in the U.S. and Europe.

TABLE I

System		Δf	Modulation Bandwidth	Q_{max}	
				Modulation	Deviation
USA	Mono	25 kHz	125 kHz	≤ 36	≤ 21
	Stereo	73 kHz	400 kHz	≤ 12	≤ 7.4
UK		50 kHz	200 kHz	≤ 30	≤ 15
Continental Europe		30 kHz	150 kHz	≤ 36	≤ 22

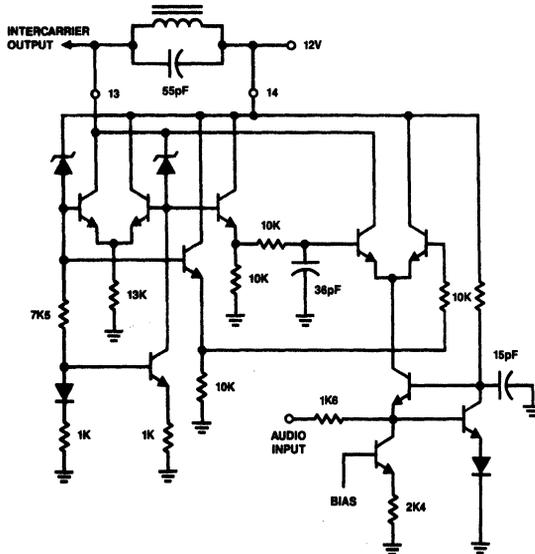


FIGURE 10. LM2889 4.5 MHz Sound Oscillator and Modulator

TL/H/8452-10

Sound Modulation (Continued)

Clearly the deviation phase offset dominates the circuit Q requirement.

If we choose a Q of around 10 then the oscillator drift with temperature (assuming a 2 degree phase change in oscillator current with a 50 degree rise in temperature) is of the order of -9 kHz. A typical receiver will generate less than 3% distortion at peak deviations with this much frequency drift but if better performance is required, then the circuit Q can be raised. High modulation linearity will still be retained with a Q of 20 and the oscillator maximum frequency drift will be halved. Alternatively temperature compensated tuning capacitors can be used (between N20 and N75). When higher circuit Q's than 20 are employed, increased audio input levels will produce the desired peak frequency deviations but with the possibility of increased modulation distortion. The actual operating parameters that are selected can be balanced between distortion as a result of modulation, and distortion in the receiver circuits as a result of oscillator frequency drift.

To ensure that we have a sufficient 4.5 MHz oscillator level to provide enough drive to the internal phase shift circuit, the load impedance at pin 13 should be greater than 3.5 k Ω . A second requirement is that we have enough oscillator level to generate the desired aural carrier amplitude when modulated on the picture carrier. This means that load impedances greater than 6 k Ω are desirable. At 4.5 MHz, a typical oscillator coil of 23 μ H will have an unloaded Q of 55 and tune with 55 pF. For a working Q of 10, the external damping resistor is 7.5 k Ω .

Stereo Sound

The introduction in the U.S. of a multiplex stereo sound system (the BTSC system combining the Zenith MCS proposal with dbx noise reduction in the stereo difference channel) with peak carrier deviations in excess of 73 kHz puts even larger constraints on the tank circuit Q. Following the same rules as before, the maximum allowable Q for low distortion is now less than 7.4—with a loaded Q of 5 being likely. With this loaded Q, maintaining a carrier center frequency accuracy better than 5 kHz with an L/C circuit becomes impractical

and other methods to set the oscillator frequency must be used. Since a crystal will provide the necessary temperature and voltage stable reference frequency a PLL is a useful solution (see *Figure 11*). Either the widely available 3.58 MHz crystals or a 4.5 MHz crystal can be used, but in either case, the L/C tank circuit frequency must be divided down before application to the phase detector. This is because frequency modulation of the sound carrier will produce many radians of phase deviation at the phase detector input—for a modulation frequency of 100 Hz and a peak deviation of 73 kHz the carrier phase change is given by Equation 3.

$$\theta = \Delta f / f_m = 73 \times 10^3 / 100 = 730 \text{ rads} \quad (3)$$

Since the linear input range of most phase detectors is less than 2π radians, the modulated carrier input must be divided down by at least 233 to keep the phase deviation within this linear range. For a 4.5 MHz crystal, the reference frequency divider M and the sound oscillator divider N are the same. Available ripple counters such as the 74HC4040 and 74HC4060 can easily divide by 128 (for monaural) or by 256 for stereo. If a 3.58 MHz crystal is used the M:N divider ratio is 35:44 requiring substantially more packages, and the odd numbered divider must be followed by an even divide of 2 or 4 to "square up" the input waveform to the phase detector. Also, since the video will include a chroma subcarrier, good isolation is needed to prevent the reference oscillator beating with the chroma sidebands.

A suitable phase detector is the 74C932 Exclusive-Or type with a sensitivity of 1.6 volts/radian. The filter at the detector output prevents the input modulation from reaching the varactor diode and distorting the audio. Even so, the loop filter must have some ac bandwidth for a reasonable acquisition time and other dynamic characteristics. The components shown in *Figure 11* have been chosen such that with a varactor sensitivity of 100 kHz/volt the loop has a hold-in range of over ± 150 kHz, with a lock-up time of less than 0.5 seconds. The T.H.D. is less than 1% for a 400 Hz modulating frequency producing 25 kHz deviation of the carrier. The accuracy of the sound carrier frequency is, of course, that of the crystal used for the reference oscillator.

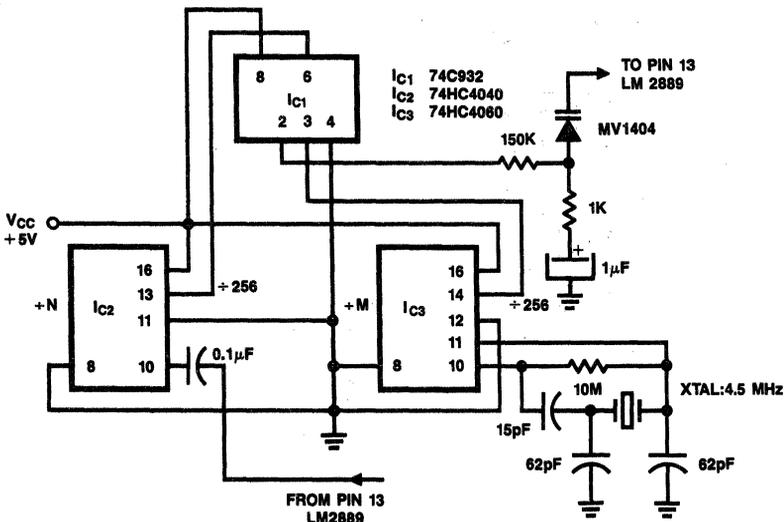
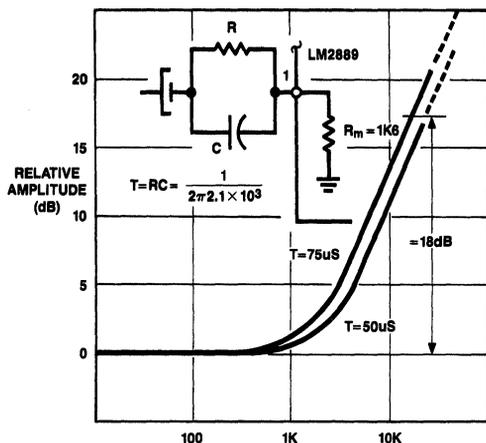


FIGURE 11. 4.5 MHz Crystal Reference Circuit

TL/H/8452-11

Audio Processing For Sound Carrier Modulation

With the proper tuned circuit Q (see Table I), a linear increase in the amplitude of the audio signal will produce a correspondingly linear increase in the frequency deviation. Television receiver sound circuits in the U.S. have a $75 \mu\text{s}$ de-emphasis and in Europe frequencies above 3.2 kHz ($50 \mu\text{s}$) are de-emphasized at a 6 dB/octave rate. This is done to help improve the S/N ratio of FM reception and the transmitter incorporates the complementary pre-emphasis characteristic—above 2.1 kHz the audio frequencies are boosted at a 6 dB/octave rate. The consequence of this modulation scheme is that if a 0 dB peak signal amplitude at 15 kHz is capable of producing a 25 kHz deviation than a similar amplitude signal at 400 Hz will produce a peak deviation of only 3 kHz—a loss of some 18 dB in S/N ratio for the midband frequencies. Broadcasters usually employ compressors to enable high modulation levels to be obtained at mid-band frequencies without overmodulating high frequencies. If the audio input to the LM2889 is being sourced from an original broadcast (a scrambled signal decoder output for example) than this audio—without de-emphasis—can be directly applied to pin 1 of the LM2889, and the overall input level is adjusted so that the modulation limits are not exceeded except for brief intervals (less than 10 instances per minute). When the audio has not already been processed a different set of conditions will apply and an audio pre-emphasis network is required at pin 1.



TL/H/8452-12

FIGURE 12. Audio Pre-emphasis

Since the audio source is likely to be at a relatively low impedance (a pre-amplifier output), the pre-emphasis network will also be used to attenuate the level of the average audio input to the LM2889 as well as providing a relative boost to the higher frequencies. The input sensitivity of the audio modulator is 150 Hz/mV which means that 118 mVrms will give a peak deviation of 25 kHz.

Next we have to decide what signal frequency and amplitude to use in calibrating the audio input. Unfortunately the $75 \mu\text{s}$ time constant for FM broadcasting was chosen at a time when equipment limitations meant there was relatively low spectral energy at higher frequencies. Today, modern audio material is not well suited to boosting above 2.1 kHz since energy peaks at only -6 dB can be obtained at 10 kHz. A further complication is the ability of the audio level meter to predict high energy peaks. If a conventional VU

meter is used, peak levels of $+10 \text{ dB}$ are possible while the meter is indicating OVU. Obviously without processing the audio to keep it within predetermined limits, the input level calibration will be somewhat empirical in nature.

If we assume the decrease in spectral energy above 10 kHz is such that overmodulation peaks above this frequency are unlikely to occur, then we can allow a signal at 10 kHz to produce full modulation deviation. Since the amplitude of most audio signals at 10 kHz is at least 6 dB below the midband frequency level, we can calibrate the audio input with a -6 dB amplitude, 10 kHz tone to produce 100% deviation. As we shall see later, a frequency close to 10 kHz will make the measurement of actual peak deviations very easy indeed. With the standard pre-emphasis network, at signal frequencies less than 2 kHz, the modulating signal amplitude at pin 1 will be -8 dB below the anticipated peak 10 kHz level producing 100% modulation. This corresponds to a modulator input level of $118/2.2 = 45.4 \text{ mVrms}$. The

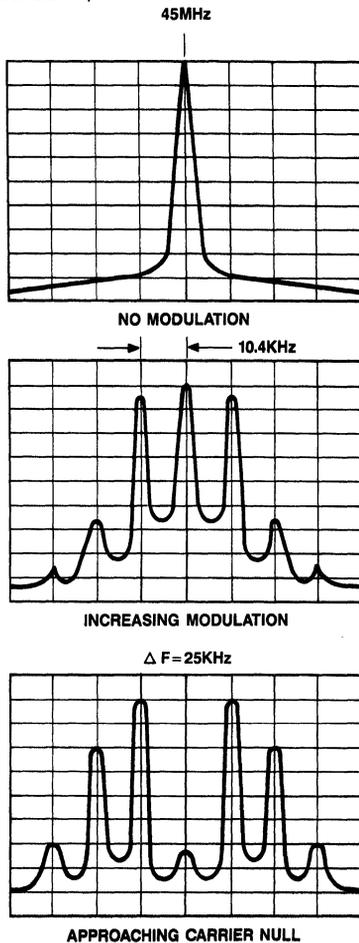


FIGURE 13. FM Spectrum with Increasing Audio Amplitude ($f_{\text{mod}} = 10.4 \text{ kHz}$) 4.5 MHz Sound Carrier Level

TL/H/8452-13

Audio Processing For Sound Carrier Modulation (Continued)

input resistance at pin 1 is $1.5\text{ k}\Omega$ so $R1 = 30\text{ k}\Omega$, if we assume an input source level of 1 V_{rms} at 400 Hz . For a 2.1 kHz breakpoint, $C = 0.0027\text{ }\mu\text{F}$.

Anyone who has observed the output from an FM circuit with a spectrum analyzer will know that for a fixed modulating frequency the output spectrum will consist of the carrier frequency component and sidebands spaced by the modulating frequency from the carrier. As the modulation amplitude is increased (the modulation index m becomes larger), the carrier decreases to a null and then increases again. The modulation indices for which carrier nulls occur can be calculated and for our purposes it is important to know that the first carrier null occurs at $m = 2.4048$. For a system maximum deviation of 25 kHz the modulating frequency f is given by:

$$f = 25 \times 10^3 / 2.4048 = 10.4\text{ kHz} \quad (4)$$

Therefore, if we use an input frequency of 10.4 kHz , as the input amplitude is increased the first carrier null will indicate peak deviation. If we continue with our assumption of a -6 dB level at 10 kHz , calibration consists of adjusting the audio input so that a -6 dB , 10.4 kHz signal causes the first carrier null. With the above pre-emphasis network, this should correspond to 500 mV_{rms} at 10.4 kHz .

We have already looked at the tuned circuit parameters at pin 13 in terms of deviation linearity and oscillator stability. With a working Q of 10, the effective load at pin 13 is $6.2\text{ k}\Omega$. The oscillator current is 0.45 mA so that the output amplitude at 4.5 MHz is 3.6 V (p-p). Some portion of this oscillator signal level is coupled over to pin 10 to set the sound carrier level and this can be done by splitting the external $7.5\text{ k}\Omega$ damping resistor into two parts. The picture carrier level is set by the offset voltage between pins 10 and 11 as described earlier. For a 2 V (p-p) video signal this offset is 2.3 V . Since the 4.5 MHz signal will be ac coupled over to the bias pin 10, it will amplitude modulate the picture RF

carrier. This is conventional AM and a 4.6 V (p-p) signal will yield sound carrier sidebands at -6 dB relative to the picture carrier. If we require a sound carrier amplitude at -17 dB , the signal coupled to pin 10 must be 11 dB below 4.6 V (p-p), or 1.3 V (p-p). This is obtained by using a $4.7\text{ k}\Omega$ resistor coupled through a $0.1\text{ }\mu\text{F}$ capacitor to pin 10, and a second $2.7\text{ k}\Omega$ resistor connected to the wiper arm of the potentiometer used to set the video modulation depth. The effect of the potentiometer setting on the aural carrier level is eliminated by a $0.1\text{ }\mu\text{F}$ capacitor connected from the wiper arm to ground. However, since the impedance presented by the potentiometer will, for all practical purposes, be relatively constant, the capacitor could be removed and the parallel resistance of the upper and lower arms of the potentiometer network used to provide the second resistor of $2.7\text{ k}\Omega$. If the video input level is well controlled, it may be possible to replace the potentiometer with a fixed divider.

The final part of the design concerns the output stage, and involves meeting the constraints applied by any regulatory agency. In the U.S., apart from the need to restrict the peak carrier output level to less than 3 mV_{rms} in 75Ω , we have two signals present in the output whose level will exceed the spurious emission limit of -30 dB with respect to the peak carrier level. One of these signals is the result of amplitude modulating the 4.5 MHz intercarrier audio on the picture carrier. Apart from the desired -17 dB sound carrier amplitude (upper sideband) an equal amplitude lower sideband will be present. For channel 3 this is at a frequency of 56.75 MHz —which is 250 kHz outside our channel lower limit. Therefore we need to provide at least 13 dB more attenuation at this frequency in the output filter. The second unwanted emission (or emissions) is the result of carrier frequency harmonics—specifically the 2nd harmonic level produced by high modulator drive. To suppress this, from -18 dB to -30 dB attenuation at 123 MHz is required.

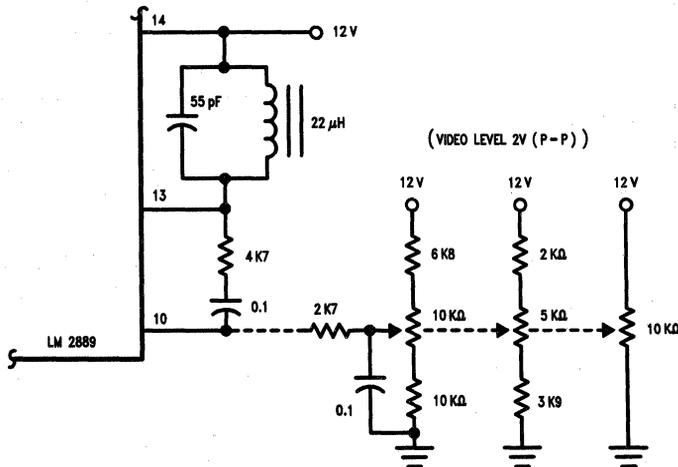


FIGURE 14. Audio Intercarrier Coupling to the Video Modulator R.F. Output and V.S.B. Filter

TL/H/8452-14

Audio Processing For Sound Carrier Modulation (Continued)

With a properly constituted baseband signal modulating the carrier, these are the only intrinsic unwanted emissions we are concerned with. Normal video modulation components appearing in the lower sideband will not have sufficient amplitude and do not extend beyond the lower channel limit. Even so, the filter requirements are not trivial.

If L-C filters are used, this can be done with three coils per channel but some alignment procedure will be required. Fortunately SAW filters are available from several sources which, although more expensive than the equivalent L-C filter, avoid the cost of production alignment. Usually the SAW filter will have a substantially greater insertion loss, but the LM2889 has enough output level to compensate for this. Both single channel and dual channel filters are available and in the latter case the LM2889 dual oscillator/modulator configuration enables easy dc switching between channels. A coil may be required, connected across the SAWF input, to tune out the SAWF input capacitance.

The load resistors connected to pins 8 and 9 will set the LM2889 conversion gain, which for 75Ω is typically 20 mVrms R.F. carrier per volt offset at the input pins 10 and 11. The actual load will include the input resistance of

the filter. Since the output of the filter will normally be terminated in 75Ω to match the cable (and provide triple transit echo suppression for a SAWF), the best way to choose the load resistor is to monitor the output to the cable and apply a dc offset between pin 10 and 11 that is equivalent to the expected video input. The resistor is then chosen to give the desired peak carrier level of 2.5 mVrms. The carrier should be unmodulated since downward modulation will reduce the mean carrier level by as much as 2-3 dB.

If the offset voltage between pin 10 and 11 is reduced, a check can be made on the residual carrier level at the output. This residual level is the result of oscillator feedthrough in the modulators and external coupling from the oscillator tuned circuits. The residual carrier level is normally better than -26 dB below the peak carrier level, ensuring good modulation linearity. High levels of residual carrier can be caused by coupling through ground or power supply leads. A good technique to minimize the effect of unwanted pick-up is to decouple the supply voltage to pin 8 and 9 load resistors over to the output connector shield ground. This removes at the output any carrier signal on the supply line to the load resistors.

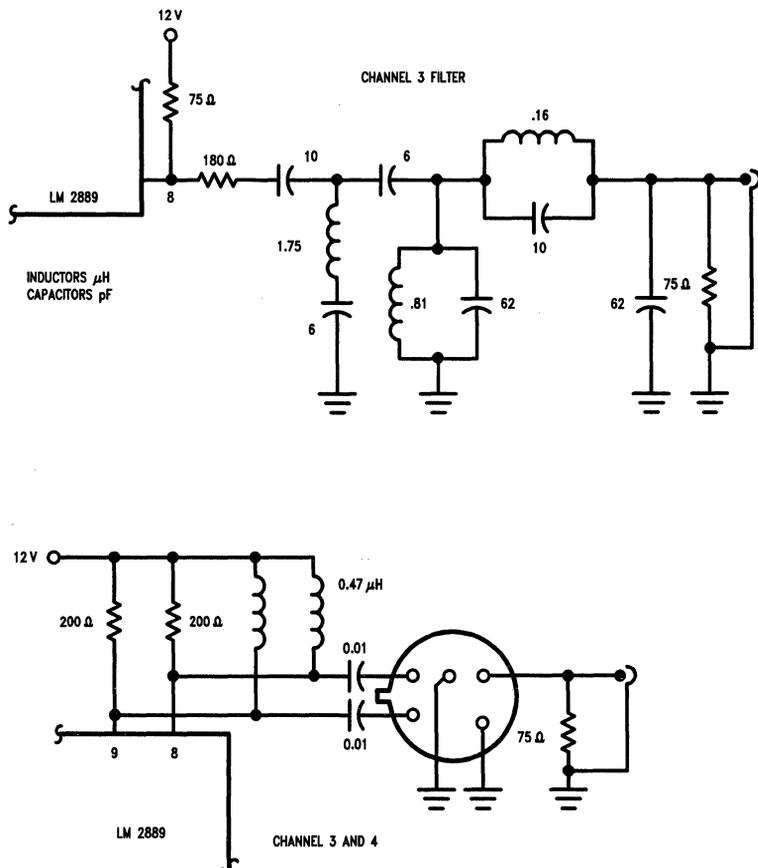


FIGURE 15. Vestigial Sideband Filters

TL/H/8452-15

Audio Processing For Sound Carrier Modulation (Continued)

Sources: SAWFs

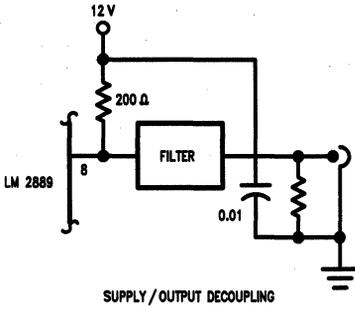
Crystal Technology, Inc.
 1035 E. Meadow Circle
 Palo Alto, CA 94303
 Kyocera International, Inc.
 8611 Balboa Ave.
 San Diego, CA 92123
 MuRata Corp. of America
 1148 Franklin Rd. S.E.
 Marietta, GA 30067

CRYSTALS

Saronix
 4010 Transport at San Antonio Rd.
 Palo Alto, CA 94303

COILS

Toko America, Inc.
 5520 W. Touhy Ave.
 Skokie, Ill. 60077



SUPPLY / OUTPUT DECOUPLING

TL/H/8452-16

FIGURE 16. R.F. Decoupling at the Output

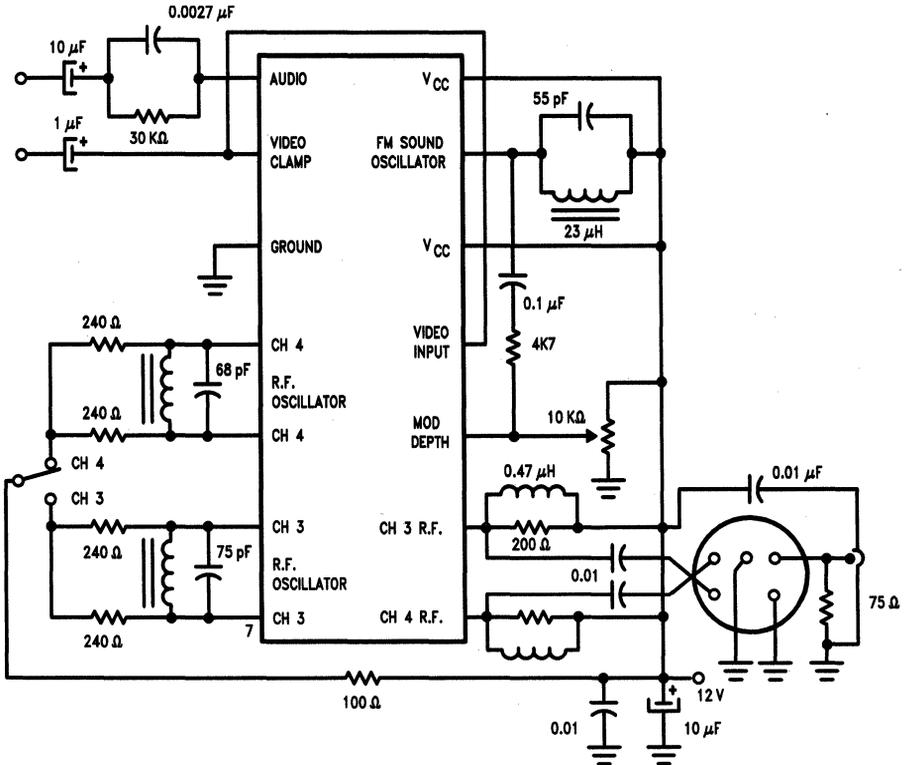


FIGURE 17. Complete R.F. Modulator External Circuit

TL/H/8452-17

A 16-Bit Video Shift Register with On-Board FIFO Operates at Rates Up to 350 Million Pixels Per Second

National Semiconductor
Application Note 580
P.H. Yeung, R. Shergill,
T.M. Wang, P.A. Tucci



INTRODUCTION

The evolution of graphics systems has, in part, been driven by the requirements for higher system resolutions. High performance raster scan graphics systems often require a resolution that exceeds 2k x 2k pixels. For non-interlaced raster scan monitors with a vertical refresh rate of sixty frames per second, the above screen size translates to a pixel rate of 300 MHz. Many systems being designed today require pixel rates significantly exceeding 100 MHz. Although this high data rate is only present over a small portion of the graphics subsystem, implementation requires significant modification to the architecture and hardware of the graphics subsystem.

SYSTEM OVERVIEW

Figure 1 shows a simplified block diagram of a graphics subsystem. The system processor interacts with the graph-

ics processor via the address and data buses and transfers the data to the display memory. The graphics processor controls the graphics subsystem; its functions include implementing the drawing algorithms, providing the CRT format (sync, blank, and size), and generating the addresses for the display memory. A BITBLT processing unit, such as National Semiconductor's DP8511 implements all of the classical BITBLT (Bit boundary Block Transfer) functions. Under the control of the graphics processor, it performs all the data path operations. The video shift register accepts the data from memory (typically dynamic RAM or video RAM) in parallel form and converts it to high speed serial data for the video DAC. The timing for this subsystem is provided for by a high performance video clock generator such as National's DP8512.

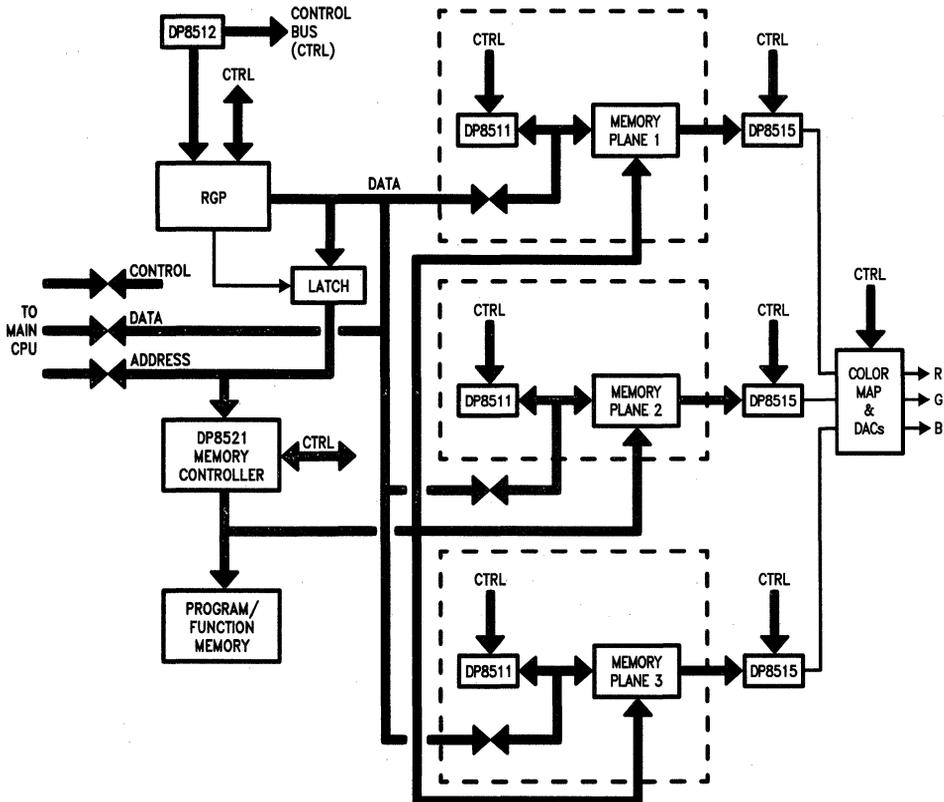


FIGURE 1

TL/F/10393-1

To meet the requirement of a pixel rate exceeding 100 MHz, graphics designers have to rely upon ECL logic circuitry that is costly, in PC board space, power consumption, and component cost, even though the remainder of the system operates at much lower data rates. Since it is unnecessary and economically impractical to build the entire graphics system in ECL logic, conversion circuitry is required to translate TTL logic levels to ECL logic levels. Furthermore, since most off-the-shelf ECL circuits and TTL to ECL translators operate with a negative supply voltage, an extra, negative power supply is required.

Design of multiple plane/board systems becomes especially difficult when high pixel rates are employed. The high pixel rates translate to high shift register parallel load rates. Looking at *Figure 1*, in multiple board systems, proper timing between data from memory (shift register parallel input data) and the load clock is difficult to maintain especially since data is arriving at the shift register from various boards while the clock (DP8512) is originating from one source. Even if variable delays are employed for adjusting the timing of the data arriving at the shift register, the total delay cannot exceed one load clock period without employing a FIFO. Again, looking at *Figure 1*, from the appearance of a clock signal on the bus, data from memory on one of the planes will not arrive at the shift register until two receiver delays, one RAM access time, and one driver delay later; at high data rates this could easily exceed one load period.

DP8515/16 EASES SYSTEM DESIGN

National Semiconductor's DP8515/DP8516 was designed with several key goals in mind. Most important was the goal to ease the system designer's job of designing the parallel to serial conversion section of the graphics subsystem. The second goal was to reduce the cost, power consumption, and board space currently required for the parallel to serial conversion in mid- to high-performance graphics systems. The final goal was to develop a flexible part, one that could easily be configured to fit well in any graphics system.

The DP8515/DP8516 Video Shift Register integrates the functions of a 16-bit 350 MHz shift register, all the TTL to ECL translators, and a 4-word deep FIFO. The integrated circuit is designed to operate with a standard TTL supply and optionally either a positive or negative ECL supply. The DP8515 meets both MECL 10K and F10K specifications while the DP8516 is both MECL 100K and F100K compatible. Interfacing the DP8515/DP8516 with a video DAC is made easy with the differential outputs SO/\overline{SO} and $S8/\overline{S8}$. With both the last bit and the 8th bit shift register outputs available, the integrated circuit can be used as two shift registers in an 8-bit system, thus cutting the component count in half. As will be shown later, the two outputs result in a lower component count for various word length systems.

System expansion to 32-bit or other word lengths is possible with the DP8515/DP8516 by using the shift register serial inputs, SI/\overline{SI} . All of the ECL inputs are differential in order to improve noise immunity; however, for single-ended use, one of the differential inputs can be wired to an on-chip reference, V_{BB} , that is compatible with standard ECL levels.

The maximum pixel rate is guaranteed to be a minimum of 350 MHz. This is achieved with a relatively low supply current of 130 mA through the use of National Semiconductor's combination Bipolar/CMOS process.

FIFO EASES TIMING PROBLEMS

As mentioned before, in mid- to high-end graphics systems the pixel rate may range from 100 MHz to over 300 MHz. The timing problems discussed above have been addressed by the DP8515/DP8516. The DP8515/DP8516 contains an on-board four word by sixteen bits FIFO. This FIFO is placed after the input latches and in front of the shift register as is shown in *Figure 2*. By setting two input pins, M0 and M1, the part may be configured in three separate ways: the FIFO in front; the FIFO bypassed with the data inputs configured as transparent latches; or the data inputs configured as edge-triggered flip-flops.

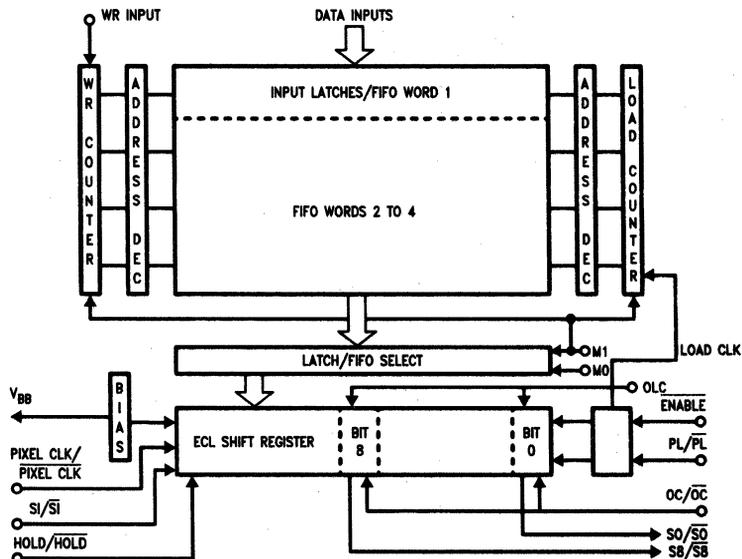


FIGURE 2. DP8515 Video Shift Register Block Diagram

TLF/10393-2

With the inclusion of the four word FIFO, timing problems between the data from memory and the shift register load clock have been eliminated. *Figure 3* shows a multiple plane graphics subsystem. The timing for this system is being provided by two of National Semiconductor's video clock generators, the DP8513 (multiple board version of the DP8512), and the DP8514. As mentioned above, data will appear at the video shift register inputs significantly delayed from the arrival of a clock signal, LCLK1, on the bus. This delay is comprised of the LCLK1 receiver delay (on P0 for example), the video RAM access time, the driver delay for the video RAM data going onto the bus, and the receiver delay for the video RAM data coming onto the board containing the video shift register. In high pixel rate systems, this delay may easily exceed the period of the data being input into the video shift register. A setup or hold time violation could then occur which may cause data to be lost or appear at the wrong time on the screen.

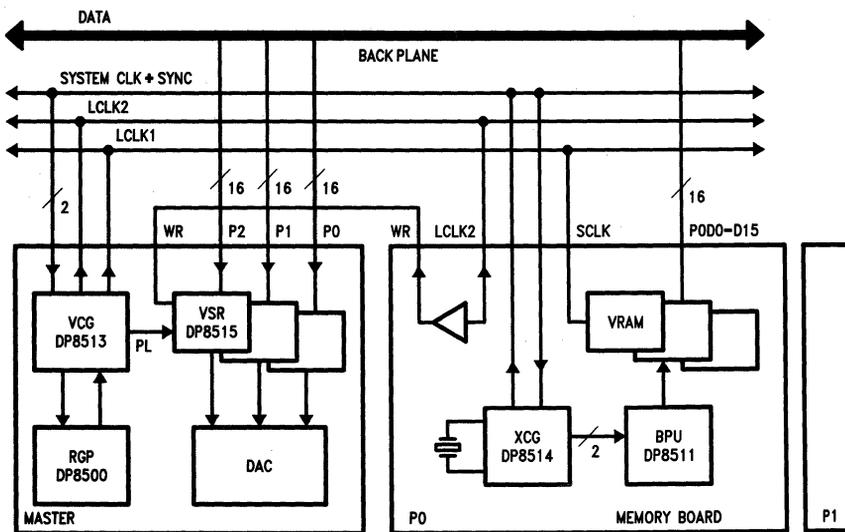
This timing problem is easily solved with the FIFO on the DP8515/16 Video Shift Register. First, by running the video shift register write clock to the various boards containing the video RAMs, any potential skew between the shift register input data and its write clock is eliminated. The fact that the total delay in the data path may exceed one data period is no longer a concern since the FIFO will act as a buffer.

With a data rate of 14 MHz (71 ns period) and a total delay of 100 ns, the system designer would have severe problems without a FIFO. Using a delay line, he might try to delay the shift register parallel load by the same amount as the data, 100 ns, however the parallel load needs to maintain a well-defined relationship with the pixel clock. For a pixel rate of 224 MHz (14 MHz data rate and 16 bit word), a setup time of about 2 ns is necessary for the parallel load. It is very diffi-

cult to maintain this setup time while delaying the parallel load by 100 ns. Another possible solution might be to delay enabling the shift register parallel load by two data periods since the data is delayed by approximately one and one-half periods. Since no delay is inserted in the shift register parallel load path, the setup time to the pixel clock is maintained. The problem with this approach, also present in the first approach, is that the data path delay must be known over temperature, supply, and part-to-part variations. If the data delay of one and one-half periods increases to more than two periods or reduces to less than one period the data is lost since a shift register parallel load will not occur during that period.

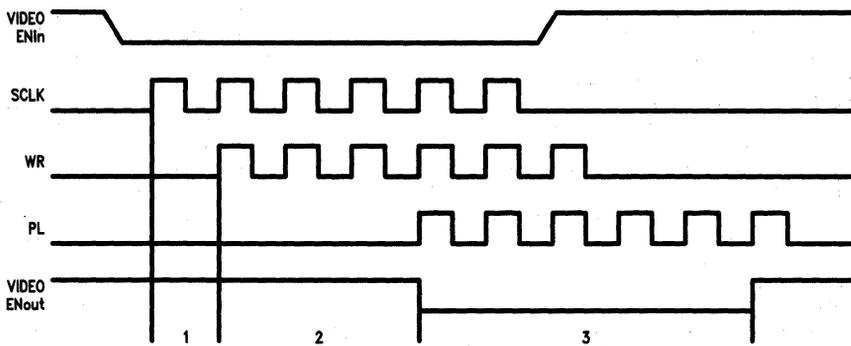
When using a FIFO, this problem does not exist. The enabling of the shift register parallel load can be delayed by up to four data periods. As long as the data delay does not exceed four periods, no data is lost since data will be written into the FIFO as soon as data and write clock reach the FIFO. Any data delay variations can be accommodated by the FIFO.

Figure 4 shows a possible timing scheme for the Video Clock, Video RAM, and Video Shift Register interface. As is shown in the figure, time period 2 is used to write the data from the Video RAM into the Video Shift Register and this is the time which will vary depending on the delays. Whether the delay in the data path is less than one data period or greater, there is no problem since data will not be read from the FIFO into the ECL shift register until time period 3. It is of no concern to the ECL shift register whether one, two or three WR clocks, along with data, occur before the Parallel Load, PL, because this will just vary the number of words written into the FIFO. Although the FIFO is absolutely essential for high speed systems, it will significantly ease design in any system.



TL/F/10333-3

FIGURE 3. Multiboard Graphics Interface



TL/F/10333-4

- 1: Transfer VRAM S/R Data to Serial Output
 2: Write VRAM Data into VSR FIFO
 3: Read VRAM Data from VSR FIFO into High Speed Shift Register

FIGURE 4. VCG/VSR/VIDEO RAM Timing Diagram

SCROLLING SIMPLIFIED WITH CHIP

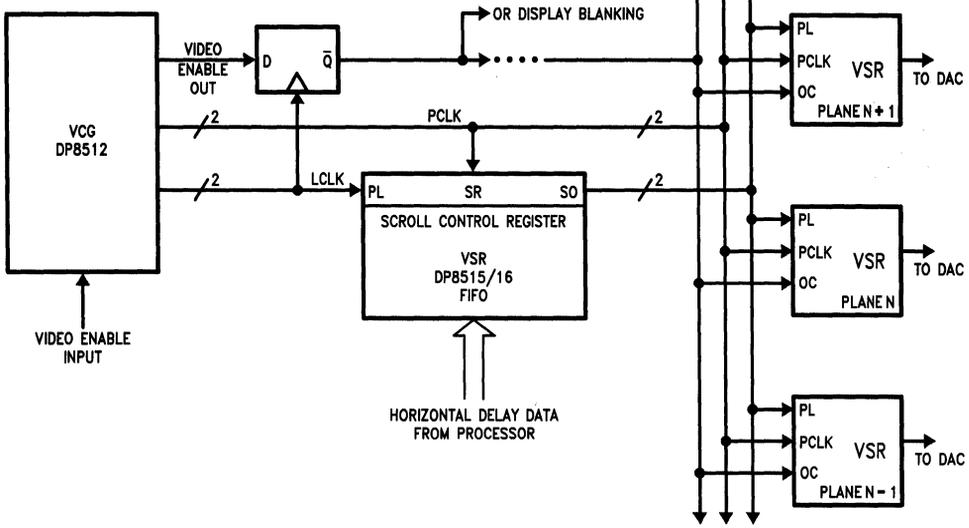
Two pins on the DP8515/16, OUTPUT CONTROL and OUTPUT LEVEL CONTROL, allow the user to manipulate the shift register data if he so chooses. With the OUTPUT CONTROL input at a high level, the last bits of the shift register, that is the SERIAL OUTPUT, SO, and the S8 bit, are prohibited from shifting while all of the other bits are being shifted by the pixel clock. This has the effect of performing a basic scrolling function on the screen. The OUTPUT LEVEL CONTROL input determines to what level the last bit is set. Even though the eighth bit is held at the output, internally information will continue to be shifted through this bit when using the OUTPUT CONTROL function.

When doing a horizontal scroll, the user wants to be able to shift the data appearing at the beginning of the screen by a certain number of bits. Ideally the data can be shifted in single-bit increments thus giving the perception of smooth scrolling rather than the "jumping" data which results from scrolling in word length increments. Since data is read out of memory in whole words, the user cannot perform a shift less than a word length, by changing the data read out of memory. He must manipulate the data coming out of the shift register which is costly in terms of component count and board space, or he must rewrite the data in the buffer memory which is costly in terms of required time to perform the operation.

Figure 5 shows how DP8515/16s can be used to perform scrolling. The DP8512 video clock generator supplies a video enable signal at the beginning of each screen line and a video shift register parallel load signal, LCLK, for each data

word. In the circuit shown in Figure 5, the video enable signal is slightly delayed with respect to the LCLK signal. Thus, the output of the D flip-flop being input to the OUTPUT CONTROL input of the video shift registers will occur up to one word after the shift register parallel load signal for these registers. The LCLK signal from the DP8512 is applied to the parallel load input of another DP8515/16 the "Scroll Control Register" in Figure 5. This shift register acts as a delay for the parallel load signal being sent to the other DP8515/16s. The value of the delay is programmed by the information loaded into the parallel inputs of the Scroll Control Register. The LCLK signal is replicated by this register by inputting the appropriate ones-zeroes pattern into the parallel inputs. The delay is achieved by properly positioning the beginning of the ones pattern within the input stream. Since the video shift register has an on-board four word FIFO, this FIFO may be used to replicate the delayed version of the LCLK signal. Also, as a result of the FIFO, the input pattern needs to be loaded only once. After the information is in the FIFO it will be cycled over and over.

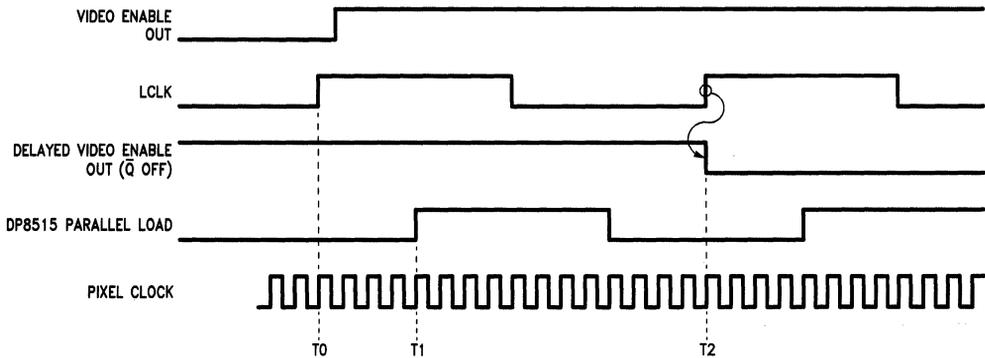
Figure 6 is a timing diagram for the circuit in Figure 5. At time T1, data is loaded into the video shift registers and is shifted on the subsequent pixel clocks. At this time, however, no data is output from the shift registers to the DACs since the delayed video enable signal (OUTPUT CONTROL) is high. At time T2 the delay video enable signal (OUTPUT CONTROL input) goes low and data is output from the video shift registers. The number of pixel clocks occurring between T1 and T2 determine by how many bits the data is scrolled.



TL/F/10393-5

Uses DP8515/16 to delay parallel load for data being shifted to color palette/DAC.

FIGURE 5. Horizontal Scroll



TL/F/10393-6

FIGURE 6

SHIFT REGISTER TAP REDUCES COMPONENT COUNT

The ECL shift register inputs and outputs have been designed to accommodate cascading and for ease of interconnection with DACs. The inputs and outputs may be used differentially or single-ended. If single-ended use is desired, the on-board V_{BB} reference is provided for the unused inputs. Differential inputs and outputs significantly ease board design by reducing the circuit's susceptibility to noise picked up while routing the traces on the printed circuit board.

The tap at the eighth bit enables the user to "customize" the part for various word lengths. For an eight bit system the component count, and thus power dissipation, is cut in half by the availability of the eighth bit; one part may be used for two words. *Figure 7* shows how the tap results in component count reduction for other word lengths. In this example a twenty-four bit word is used. Three video shift registers can perform the parallel to serial conversion for two words. If the tap was not present, another video shift register would be necessary. This results in a savings of four video shift registers (12 versus 16) in an eight plane system. This flexibility reduces the total cost of components, board space, and power.

ADDITIONAL FEATURES ENHANCE FLEXIBILITY

Two additional features provide the system designer with even more control over the operation of the DP8515/16. A HOLD input is provided which, when held high, inhibits the shifting of all of the bits of the shift register. This is in contrast to the OUTPUT CONTROL input which only inhibits the

shifting of the SO and S8 bits. The HOLD input is ECL compatible and may be applied for as short a time as one pixel clock.

An ENABLE input is also provided on the DP8515/16. When held high, this TTL compatible signal will inhibit the parallel loading of the data into the shift register.

ECL POWER SUPPLIES NO LONGER A PROBLEM

As mentioned earlier, the DP8515/16 was designed with several goals in mind; one of these was to develop a flexible part that could easily be configured to fit well in any graphics system. The DP8515/16 power supplies are specified to meet this goal. The DP8515/16 has separate pins for the TTL supply and the ECL supply. However, the ECL supply is specified so that it may be shared with the TTL supply. Both the 10K (DP8515) and 100K (DP8516) versions have all specifications guaranteed with supplies varying from 4.2V to 5.5V. Also the ECL supply may be either positive or negative. This allows the user to configure this part around his system, not the other way around. Also, the on-chip V_{BB} reference eliminates the problem of having to generate different references for positive and negative supplies if single-ended signals are being used.

DP8515/16 IDEAL FOR MANY APPLICATIONS IN ADDITION TO GRAPHICS

The DP8515/16 video shift register is well-suited for any high-speed application where parallel to serial conversion is required. Any high-speed system will have the timing problems described above which can be eliminated by placing a FIFO in front of the shift register.

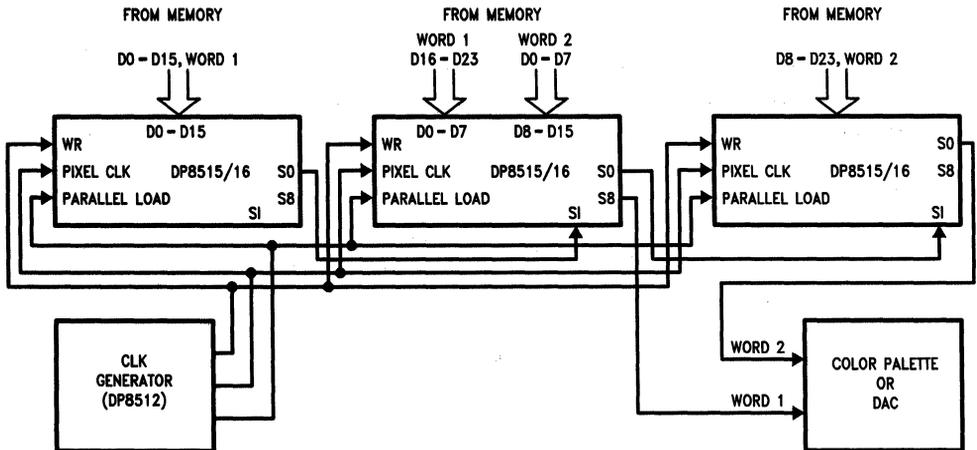


FIGURE 7. DP8515/DP8516 Interconnection for 24-Bit Word

TL/F/10393-7

The DP8515/16 is ideal for the generation of any high-speed pattern for ECL circuitry. The pattern can be parallel-loaded at a low rate into the FIFO. The serial pattern can then be shifted out at sixteen times this rate and at ECL levels. If the DP8515/16's are cascaded, higher differentials between input and output data rates are possible. A pattern of up to sixty-four bits can be stored in the FIFO and, by cycling through the FIFO, continuously shifted out of the shift register in serial form. If longer patterns are desired, the FIFO is periodically updated. The serial data from the shift register can be clocked by a synchronous clock source which was divided down to generate the write clock for the FIFO.

SUMMARY

The DP8515/16 video shift register has been designed with the intent of significantly easing graphics subsystem design and at the same time reducing the cost of the parallel-to-serial data conversion section of the graphics subsystem. The chip can be configured to meet the needs of the system. A four-word FIFO is available and when put in front of the high speed shift register, eliminates the serious data timing problem which arises when designing multiple board systems. This video shift register may operate off of positive or negative power supplies, is available in 100K and 10K options, and includes features which allow smooth panning and pixel replication zoom functions. Also a tap at the eighth bit allows the chip to operate efficiently with any word length.

Mid-Scan-Line Load Techniques

Using the DP8500 Raster Graphics Processor

National Semiconductor
Application Note 553
Gary Betz



A video graphics system's main task is to perform drawing operations with the highest possible performance. In order to achieve high drawing rates the drawing processor must have access to the frame buffer via the system bus with as little interruption as possible from other system activities. Unfortunately, all video systems are required to perform some level of display refresh which, unless the frame buffer is dual-ported, will require the refresh controller to fetch data from the frame buffer utilizing the same bus that the drawing processor is using. As a result, the drawing process will be interrupted in order to perform the display refresh operations which will ultimately hinder the drawing performance of the system.

VDRAM TECHNOLOGY

The bus bandwidth overhead associated with display refresh has been greatly reduced with the advent of the dual-ported Video Dynamic RAM (VDRAM). Shift registers tied to a separate video data port have been incorporated into conventional DRAM devices providing a convenient way to transfer frame buffer data to the video output section. Data is transferred to the shift registers a row at a time and then clocked out of the serial port as needed by the video refresh section. The activity of transferring data from the random memory array to the shift registers is commonly referred to as a data transfer cycle.

Utilizing VDRAM devices in a RGP-based graphics system inherently increases the system bus bandwidth for other bus activities not related to display refresh. The time ordinarily needed for fetching display data, particularly in a system with a frame buffer consisting of conventional DRAM devices, can occupy greater than 50% of the available bus bandwidth. In contrast, a frame buffer consisting of VDRAM devices in a RGP-based system are accessed for display update at the most once every 4096 pixels displayed, so the system bus is freed up for other system activities, such as drawing.

Display refresh using VDRAM devices is accomplished by performing a read cycle on the random access port in the data transfer mode ($\overline{DT}/\overline{OE}$ before \overline{RAS}), which enables the transfer of a row of frame buffer data to its internal 256-bit shift registers. Once the read operation has completed, the video data is clocked out of the serial access port with the *Shift Clock (SC)* input on the VDRAM. The advantage of this dual-port arrangement is that the system bus is utilized only once for the transfer cycle and then the shift registers take over delivering the frame buffer data to the display.

The timing requirements of VDRAMs with respect to the data transfer cycle, however, are rather rigid. The rising edge of the $\overline{DT}/\overline{OE}$ input, which controls when the data is loaded into the internal shift registers must occur at a point in time when the last bit of data is clocked out of the serial port but before the next rising edge of *Shift Clock*. At video rates, this can become difficult to control.

SIMPLIFYING THE DISPLAY REFRESH TIMING

Because of the precise timing requirement required for display refresh, it is not uncommon to feed the outputs of multiple VDRAM devices to a high speed parallel-to-serial shift register so that a slower clock can be used to clock to VDRAM shift register. This essentially relaxes the window for performing the transfer cycle as well as extending the time interval between transfer cycles.

In some systems this solution already exists because of the word width imposed by the drawing processor. For example, an RGP-based system utilizes a 16-bit data bus which implies that four VDRAM devices (i.e., 64k x 4 devices) be used to form the required data width on the random access port. The serial port word width matches the random access word width of 16 bits. The period (in pixels) between data transfer cycles can be calculated by multiplying the number of bits used to create the data word, which, in the RGP's case is sixteen, by the length of the VDRAM's serial shift registers. Using the RGP-based system as an example the period can be calculated as $16 \times 256 = 4096$ bits.

This effectively creates a longer serial word per transfer cycle performed, which results in extending the maximum period between data transfer cycles to be equal to some multiple of the horizontal line length. With some display widths, for example, a 1024-bit wide format, this would allow the data transfer cycle to occur during the blanking period, eliminating the need for generating the precise timing that otherwise would be required if performing the data transfer cycle during active video. This method provides a reasonable solution for some system designs which can match the display format to the frame buffer's effective width or some multiple of it.

SIMPLICITY WITH COMPROMISE

A problem with the above approach is that not all display's horizontal dimensions can conform to the implied boundaries of the frame buffer. A good case in point is the 1280 x 1024 display. Given a 16-bit wide RGP data bus and using 64k by 4 VDRAM devices, a simple frame buffer could be implemented using two banks of 4 devices configured as illustrated in *Figure 1*. Each VDRAM is responsible for providing 4 bits of video data which is fed to a 16-bit shift register. This configuration would allow for a data transfer cycle to occur at a maximum of once every 4096 pixels displayed ($16 \times 256 = 4096$).

In order to guarantee that the display refresh cycle does not occur during active video, a value for the word offset between vertically adjacent pixels in the frame buffer called the display buffer warp, must be chosen so that the shift registers never become empty before reaching the end of the scanline.

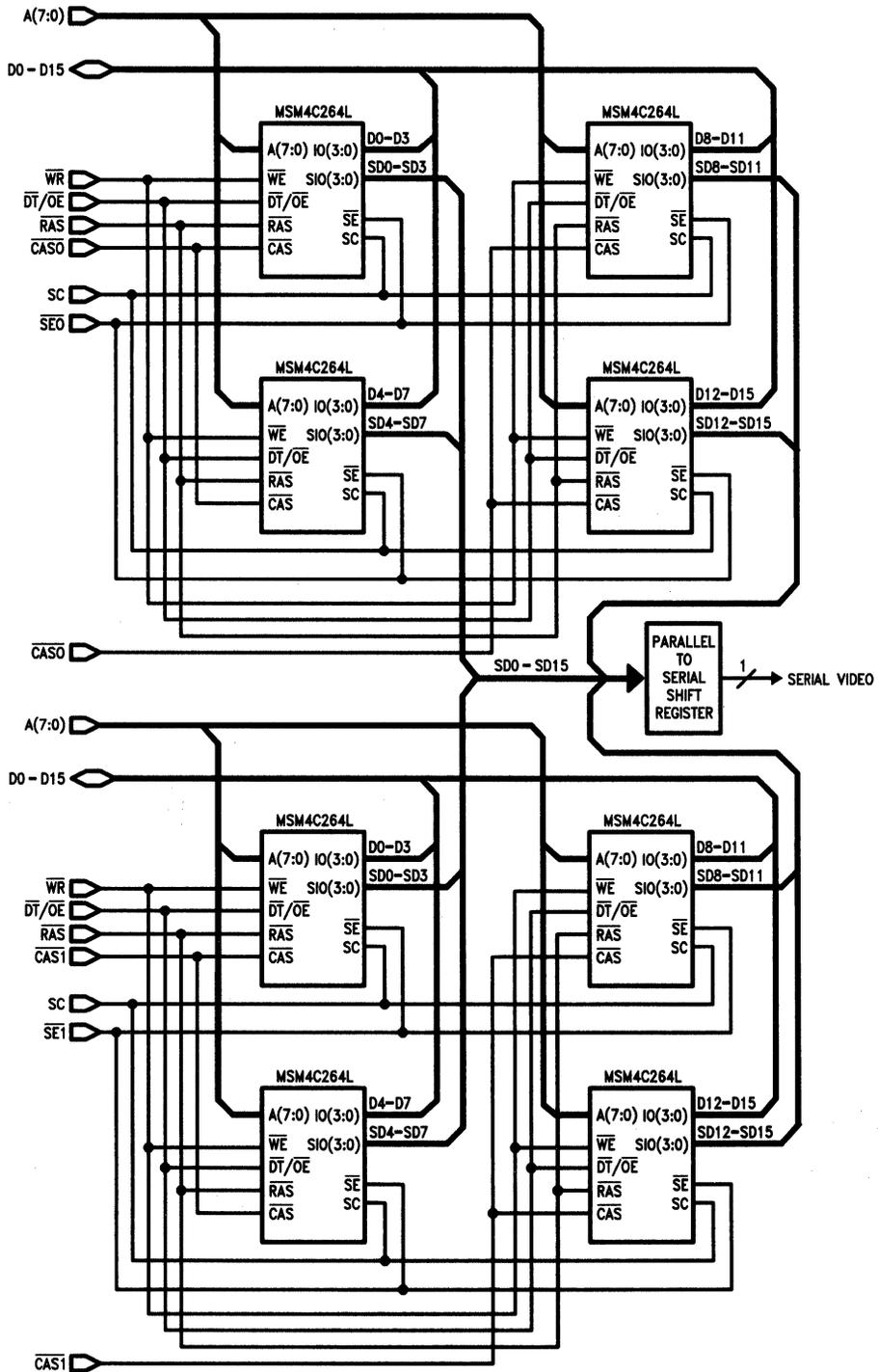


FIGURE 1. 2k x 1k x 16-Bit Frame Buffer

TL/F/9884-1

When the RGP generates a display refresh cycle at the start of each scanline, the address generated by the RGP is calculated by adding the display buffer warp value to the display buffer starting address each time a new scanline is started. This requires that the display buffer warp be set so that successive scanline's starting addresses can be divided into the shift register's effective length evenly. In this case, the display buffer warp is calculated as being $\frac{1}{16}$ th (to convert to words) the value closest to the horizontal line length of 1280 which can be evenly divided into 4096, which is 2048. In this case, each time a new scanline is started, the new refresh address would be the previous line's starting address plus 128 (2048/16). In this way the VDRAM's shift register, which is really 4096 bits long, would never be exhausted during active video.

An apparent disadvantage of using this type of refresh configuration is that it wastes a large percentage of the frame buffer. The memory between pixels 1280 and 2048 on each scanline cannot be used in a linear fashion because it is in fragments throughout the frame buffer. In this example, a total of 48k words of random access memory which could be utilized for other system tasks would be forfeited to simplify the data transfer logic timing.

A more efficient design would therefore allow the display buffer warp to be set such that the ending address of each scanline is adjacent to the beginning address of the next scanline in the frame buffer, eliminating the wasted buffer area. As a tradeoff the system would be required to perform a new data transfer cycle on demand, regardless of being in active video or not, when the VDRAM shift register's contents are about to become exhausted. This operation is called a *Mid-Scan-Line Load Operation*.

A BETTER SOLUTION USING THE RGP

The DP8500's refresh logic was designed with VDRAM technology in mind and provides an easy way to implement mid-scan-line load with a minimum of external components.

The Video Refresh Control (VRC) block of the RGP consists of a refresh address counter which is clocked by the *LCK* input, and the timing chains that ultimately produce the sync and display refresh request signals. When the refresh counter is either clocked to 255 or when the end of a scanline is reached, the VRC block generates a signal, \overline{DRREQ} , which indicates that the RGP will perform a display refresh cycle when the current bus cycle completes. This signal can be used to flag the system memory controller which can generate the appropriate data transfer cycle timing in the next bus cycle.

One *LCK* cycle after the RGP's internal refresh counter clocks to 255, the RGP will internally request a display refresh cycle. If there is a bus cycle currently in progress, the RGP will continue executing the current cycle, regardless of its length, until the *WAIT* input is sampled high. At this point, the RGP will initiate a display refresh cycle and allow the system to perform the VDRAM data transfer cycle.

The criteria for performing a successful data transfer cycle during active video is to start the cycle as normal by subse-

quently asserting $\overline{DT/OE}$, \overline{RAS} and \overline{CAS} low and then position the rising edge of the $\overline{DT/OE}$ signal so that it transitions after the last word of serial port data is transferred to the 16-bit output shift register but before the next rising edge of *SC* (shift clock). This will ensure that the last word of video data is not corrupted before being transferred to the 16-bit output shift register.

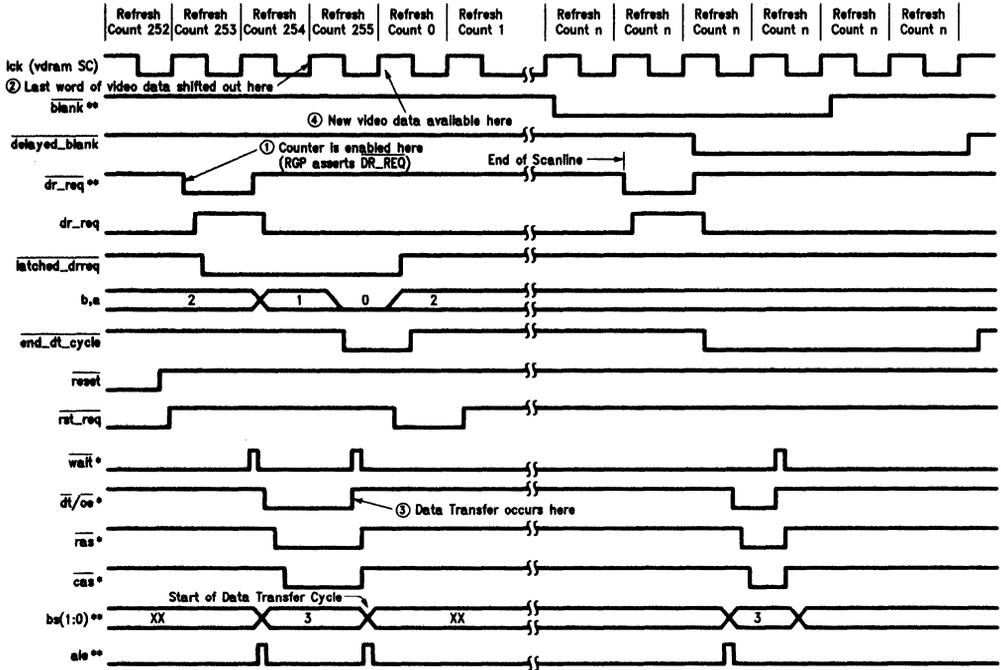
Because the period between the \overline{DRREQ} transition and when the address counter rolls over is fixed at 1 *LCK* cycle, a problem may arise if the display refresh request from the RGP comes during a bus cycle which is longer than the period of time that is needed to shift out the remaining 16-bit word of display data. There is a possibility that the data transfer cycle would occur too late, resulting in an interruption in the process of delivering video data to the display.

This problem can be overcome by delaying the blanking signal by "n" *LCK* periods which, in effect, allows the RGP's internal refresh counter to start clocking "n" *LCK* periods before active video. This causes the RGP to assert the \overline{DRREQ} signal "n" *LCK* periods before the VDRAM shift register's contents are exhausted allowing the RGP ample time to complete the current bus cycle and start the display refresh cycle. The number of *LCK* periods to delay the blanking by can be determined by summing up the time required to execute the longest bus cycle and the period with in the display refresh cycle just prior to when the $\overline{DT/OE}$ is to be clocked high and dividing that total by the *LCK* period.

The period of time between when the RGP asserts \overline{DRREQ} and when the VDRAM shift registers are empty is determined by the "n" *LCK* delay of the \overline{BLANK} signal plus the time it takes to perform the display refresh cycle. It is important to note that the timing relationship of \overline{DRREQ} and $\overline{DT/OE}$ must remain constant. The bus state machine must guarantee that this period is maintained regardless of the length of the current bus cycle. This can be done by using the \overline{DRREQ} signal to enable a down counter being clocked by *LCK*. The count length is set to match the delay of the \overline{BLANK} signal. The counter's output is sampled by the bus state machine to hold off the completion of the display refresh cycle by inserting wait states to the RGP during the display refresh cycle. The $\overline{DT/OE}$ signal is set low but is not released high again until the counter reaches zero. At this point, the bus state machine asserts both $\overline{DT/OE}$ and *WAIT* high and finishes the display refresh cycle. In this way, the period between \overline{DRREQ} and $\overline{DT/OE}$ transitions remains constant, guaranteeing the transfer of data to the shift register at exactly the right time.

IMPLEMENTATION

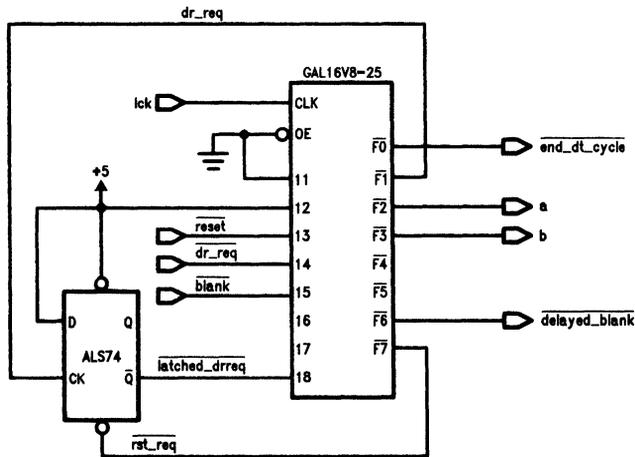
The logic for the above circuit was implemented in a single National 16V8 Generic Array Logic (GAL[®]) device. The logic diagram, timing, and GAL equations can be found in *Figures 2, 3, and 4* respectively. The circuit was tested on National's DP850EB 4-Plane Evaluation Board. Please consult the *DP850EB 4-Plane Evaluation Board System User's Guide* for further information on designing a graphics system around the DP8500 Raster Graphics Processor.



TL/F/9984-2

- * Generated by the Bus State Machine
- ** Generated by the RGP

FIGURE 2. Mid-Scan-Line Load Timing



TL/F/9984-3

FIGURE 3. Logic Implementation for Mid-Scan-Line Load Circuit

```

module      ms14
title      'Mid_scan_line_load logic
Gary Betz National Semiconductor Corporation, Santa Clara, Ca.
June 8, 1988

```

This device delays blanking by two clock periods. It also contains a 2-bit counter which is used to hold off finishing the display refresh cycle. Please see the timing and schematic diagrams for more details on how this circuit is used.

```

IC1      device      'pl6v8r';
        clk          pin      1;      "LCK input.
        I1           pin      2;
        I2           pin      3;
        reset~      pin      4;      "Reset in from RGP.
        dr_req~     pin      5;      "Inverted dr_req~ for flip/flop.
        blank~     pin      6;      "Blanking input.
        I6          pin      7;
        I7          pin      8;
        latched_drreq~ pin      9;      "Starts the down counter.
        GND         pin      10;
        OE          pin      11;
        end_dt_cycle~ pin      12;    "Used to signal state machine to
        "finish the display refresh cycle.
        dr_req~     pin      13;    "Dr_req~ input from RGP.
        a           pin      14;    "2-bit counter LSB.
        b           pin      15;    "2-bit counter MSB.
        F4          pin      16;
        F5          pin      17;
        delayed_blank~ pin      18;  "Blanking delayed by 2 LCKs.
        rst_req~   pin      19;    "Clears the latched dr_req~
        VCC         pin      20;

equations      !end_dt_cycle = (!a & !b) # !delayed_blank ~;
              !a :=          (a & !latched_drreq~)
              # (I2 & latched_drreq~);
              !b :=          (b & !a & !latched_drreq~)
              # (!b & a & !latched_drreq~)
              # (!I2 and latched_drreq~);

              !dr_req = dr_req~;
              !rst_req~ := !end_dt_cycle # !reset~;
              !delayed_blank~ := F5;
              !F5 := blank~;

end

```

FIGURE 4. GAL Equations

Accurate Timing for Multi-Board Graphics Systems

National Semiconductor
Application Note 554
Craig Davis



Two members of National Semiconductor's Advanced Graphics Chip Set are the DP8513, Multiboard Video Clock Generator, and the DP8514, Crystal Clock Generator. These devices are used in multiple board/plane graphics systems to generate and synchronize the clocks which drive the graphics processors and transfer VRAM information out to DACs for CRT display updates. Synchronizing the signals involved in these operations is becoming increasingly difficult as display resolutions increase and as more color planes are offered. The higher display resolutions increase the rate which information must be transferred from memory to the display. The increasing number of color planes force VRAM information to reside off of the mother board thereby introducing timing constraints associated with back plane data transfers. Utilization of the DP8513 and DP8514 greatly simplify these clock generation and synchronization problems.

Updating a CRT display involves a parallel to serial conversion of screen information from VRAM memory into a video shift register which then serially feeds a DAC. Pixel and load clocks are used to coordinate the transfer of the VRAM data into the video shift register and also the shifting of the data out to the DAC. Present day screen resolutions of 1280 x

1024 pixels and beyond are becoming commonplace. This translates to a pixel rate exceeding 100 MHz for the conventional non-interlaced raster scan monitor. Manipulation of the screen information at these high data rates requires that the video shift register and DAC portions of the graphics system be implemented in high speed Emitter Coupled Logic (ECL). The ECL pixel and load clocks associated with these data transfers must synchronize with the slower MOS clocks driving video memory and the graphics processors so that screen updating, blanking, and scrolling operations are properly carried out. The DP8513 and DP8514 generate these pixel, load, and processor clocks and provide the appropriate gating circuitry to simplify interconnection into either MOS, TTL or ECL based systems.

The DP8513 generates the pixel, load, and processor clocks using a digital phase lock loop as shown in *Figure 1*. The device takes an input reference signal provided by the DP8514 and uses it to precisely control a high speed voltage controlled oscillator from which the pixel, load, and processor clocks are derived. The phase lock loop attributes enable the input reference signal and all the output clocks it generates to inherently be phase aligned.

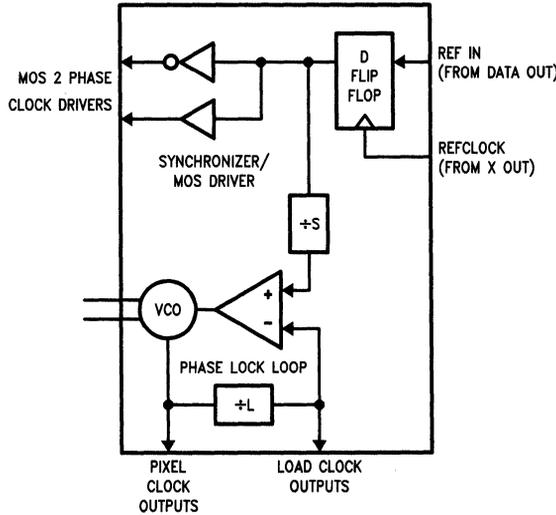
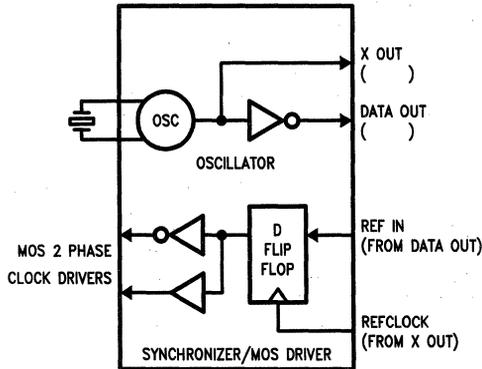


FIGURE 1. DP8513 Functional Diagram

TL/F/10005-1

The DP8514 is comprised of two independent functions as shown in *Figure 2*: an oscillator with TTL outputs and a synchronizer/MOS clock driver. The oscillator function provided by the DP8514 need only be implemented once in any system on any arbitrary board. The oscillator and oscillator divided by two outputs, X OUT and DATA OUT, are intended to be connected out to the backplane to drive all boards identically. (In a system where it is desired to lock to an external video source, this oscillator function should reside on the motherboard. In this configuration, the DP8513's PLL function can interact with the DP8514's crystal/ceramic oscillator to phase align itself to an external horizontal input signal.)



TL/F/10005-2

FIGURE 2. DP8514 Functional Diagram

The DP8514's synchronizer/MOS clock driver function would be required on every memory board in a system. The synchronizer function receives the DP8514 reference signals, X OUT and DATA OUT, off of the backplane and provides controlled duty cycle 2 phase MOS clock drive for the DP8510 or DP8511 Bit BLT Processing Units associated with each color plane. The DP8513 incorporates a synchronizer/MOS clock driver function identical to the DP8514's which is used to drive the Raster Graphics Processor. This configuration enables the RGP to maintain phase alignment with each of the remotely located Bit BLT units it commands.

Increasingly higher screen resolutions require higher pixel and video shift register load clock rates which in turn push the VDRAMs to be operated near their maximum specified frequency. In a multiple board system the video shift registers and DAC's must be contained on the mother board while their VDRAM data is received through a backplane from another board. This configuration is necessary because the ECL signals, which the video shift registers and DACs required, aren't suited for distribution out through the backplane. Clocking VDRAMs from a source on the motherboard and receiving data back from them through the back-

plane presents some significant timing problems. The backplane introduces clock waveform distortions and driver/receiver delays which limit data throughput. The total delay associated with the transfer of VDRAM data to the video shift register must not exceed one load period. This operation involves clock and data signals propagating through two receivers, one VDRAM access time, and one driver. As the load clock cycles approach 40 ns, satisfying the timing requirements becomes a significant challenge.

The DP8513 and DP8514 address these multiple board timing problems associated with the backplane. The DP8513 provides several different gated load clock outputs which help to eliminate the one load clock transfer constraint. The DP8514 contributes by providing a means of reducing the reference clock phase skews between boards and eliminating concerns about the clock waveform distortions introduced by the backplane.

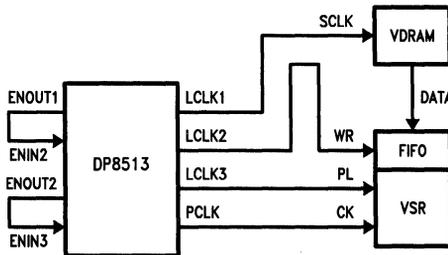
Figures 3 and 4 shows how the DP8513 works in conjunction with the DP8515 video shift register which has an on-board FIFO to simplify the load clock timing. The sequence begins with the processor issuing a video unblank signal to the DP8513's ENIN1 input. This begins a train of load clock pulses out of the DP8513's LCLK1 output synchronous with the next load clock transition. These pulses will travel down the backplane and result in data being strobed out of the VRAM's serial port. The DP8513's ENOUT1 pin to ENIN2 pin connection results in a train of load clock pulses (LCLK2) beginning one load clock cycle after LCLK1 which also travels down the same backplane path. This LCLK2 train of pulses will be routed back to the motherboard along the same path as the VDRAM's output data. The LCLK2 train of pulses thus strobe the VDRAM data into the video shift register's FIFO port with a setup time dependent only on the VDRAM access specification. The connection of ENOUT2 to ENIN3 on the DP8513 provides a train of load clock pulses (LCLK3) which lags the LCLK2 output pulse stream by 4 load clock cycles. The LCLK3 train of pulses are ECL and are used to load the FIFO data into the shift register. This offset enables the FIFO to fill itself up and allow VDRAM to FIFO data write transfers to operate asynchronously from data read transfers to the shift register.

The DP8514's oscillator outputs, X OUT and DATA OUT, which are routed out to the backplane are received and resynchronized on all the boards by a D flip flop's data and clock inputs residing within either a DP8513 or DP8514. This method of sending the reference clock around the system ensures that the receiving device can easily regenerate a synchronous output with a controlled duty cycle irrespective of the distortion introduced by the backplane drivers and receivers. This becomes increasingly important as MOS processors are driven closer to their rated operating frequency. The DP8514's architecture additionally minimizes system clock skews by virtue of its synchronizer/MOS clock driver functional block being identical to the DP8513's.

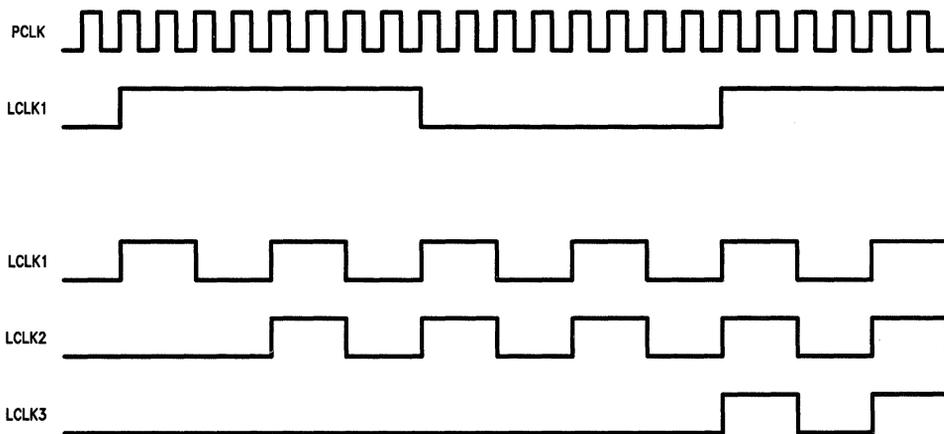
This configuration results in the MOS clock transitions coinciding with the pixel and load clock phase transitions generated by the DP8513's PLL functional block.

Utilization of a stable low cost crystal resonator as a reference source and matched internal delay paths enable the

critical graphics clocks generated by these devices to exhibit good temperature and supply voltage insensitivity. The combination of these two products provide system timing solutions which are difficult to obtain with off the shelf MSI components.

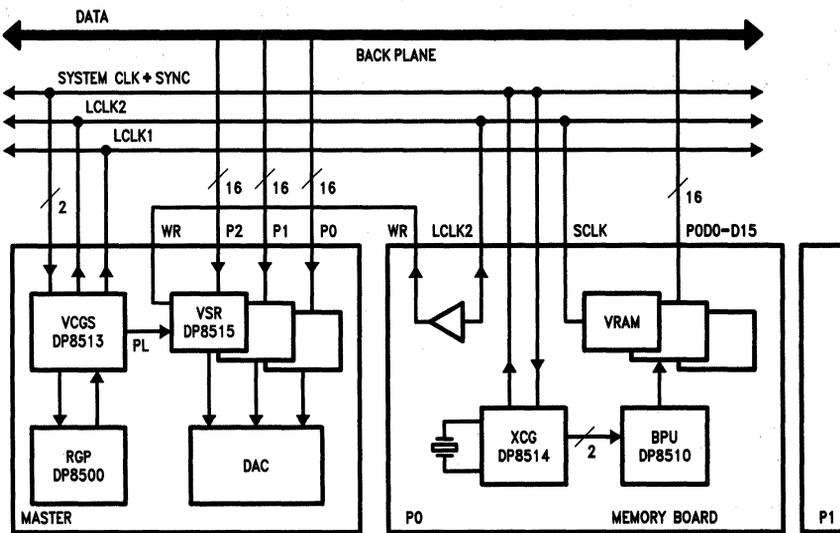


TL/F/10005-3



TL/F/10005-4

FIGURE 3. System Configuration and LCLK Waveforms Using a VSR Containing FIFO



TL/F/10005-5

FIGURE 4. System Interconnect Diagram

DP8512/13/14 Video Clock Generator Evaluation Board

National Semiconductor
Application Note 604
Craig Davis



GENERAL DESCRIPTION

This Evaluation Board was designed to demonstrate some key features of the Video Clock Generator family of integrated circuits. Three chip types will be discussed in this note, those being: DP8512, DP8513 and the DP8514 (DP8530 will not be discussed).

The DP8512 is a clock generator designed to drive high speed ECL shift registers and DAC's in a graphics system. It generates an ECL pixel clock (PLCK), load clock (LCLK) and MOS microprocessor clocks. The DP8512 includes an on-chip crystal oscillator circuit (XTLB, XTLC).

The DP8513 is identical to the DP8512 with the exception that the crystal oscillator inputs are replaced with TTL inputs designed to accept a pair of external reference signals.

The DP8514 is a crystal clock generator designed to generate a 2 phase non-overlapping MOS Clock derived from an on-board crystal oscillator circuit. It also includes a synchronizer circuit used to eliminate clock duty cycle distortion in multi-board applications. The DP8514 interfaces directly with the DP8513. Three configuration schemes for these clock chips are shown in *Figures 1* through *3*.

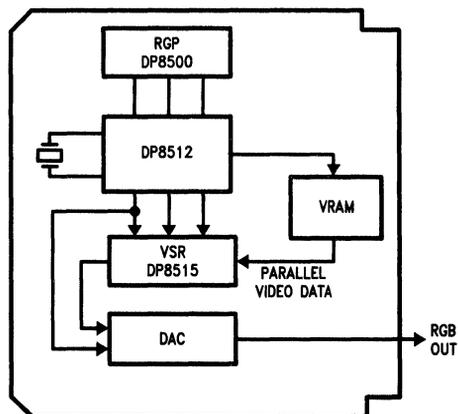
Figure 1 shows the selection of clock components for a single board design. The appropriate clock product is a DP8512 for a single board application. All the necessary clocks are generated on-chip in conjunction with an external crystal. All the outputs from the DP8512 are terminated within the domain of the single board.

Figure 2 shows the most common multiple board configuration. The appropriate clock products for a multi-board design include a DP8513 and multiple DP8514's. Alignment of clock edges from board to board is optimized by utilization of multiple DP8514 clock devices. This device helps to eliminate performance degradation due to delays and duty cycle variations down the backplane. The DP8513 is designed to accept the backplane reference signal generated by the DP8514. All the high speed ECL signals related to the DAC's and Video Shift Registers are confined to the mother board. No ECL signals need to be shipped across the backplane. Each of the daughter boards contains a DP8514 which drives its local BITBLT units and frame buffer memories. The portion of the DP8514 which is used on the daughter board is a re-synchronizer/MOS clock driver for driving the local BITBLT and frame buffer memories. The

DP8514 receives $1f$ and $2f$ reference clock signals from the backplane originating from a DP8514 on the mother board. The DP8514s on the daughter boards receive these two signals and output a $1f$, 2 phase non-overlapping MOS clock. This clock is now duty cycle corrected by the DP8514's resynchronizer circuit.

Figure 3 shows a configuration which enables a user to synchronize very high speed ECL signals in a multi-board application. The ECL signals are all phase aligned to a common low frequency clock derived from the backplane. This design eliminates the need for an ECL controlled impedance backplane bus.

The Evaluation board demonstrates attributes of each of the above three figures. The board is capable of exercising the DP8512/13 to its maximum pixel clock rate of 225 MHz. It contains one socket which accepts a DP8512 or DP8513 device (U1) and two DP8514 sockets. One DP8514 (U2) is configured as a crystal oscillator without using the MOS clock driver section. The other DP8514 (U3) is configured as a MOS clock driver. Jumpers on the board (J1 thru J4) allow flexibility of reference clock signal routing.



TL/F/10398-1

FIGURE 1. Clock Chip DP8512 In Single Board

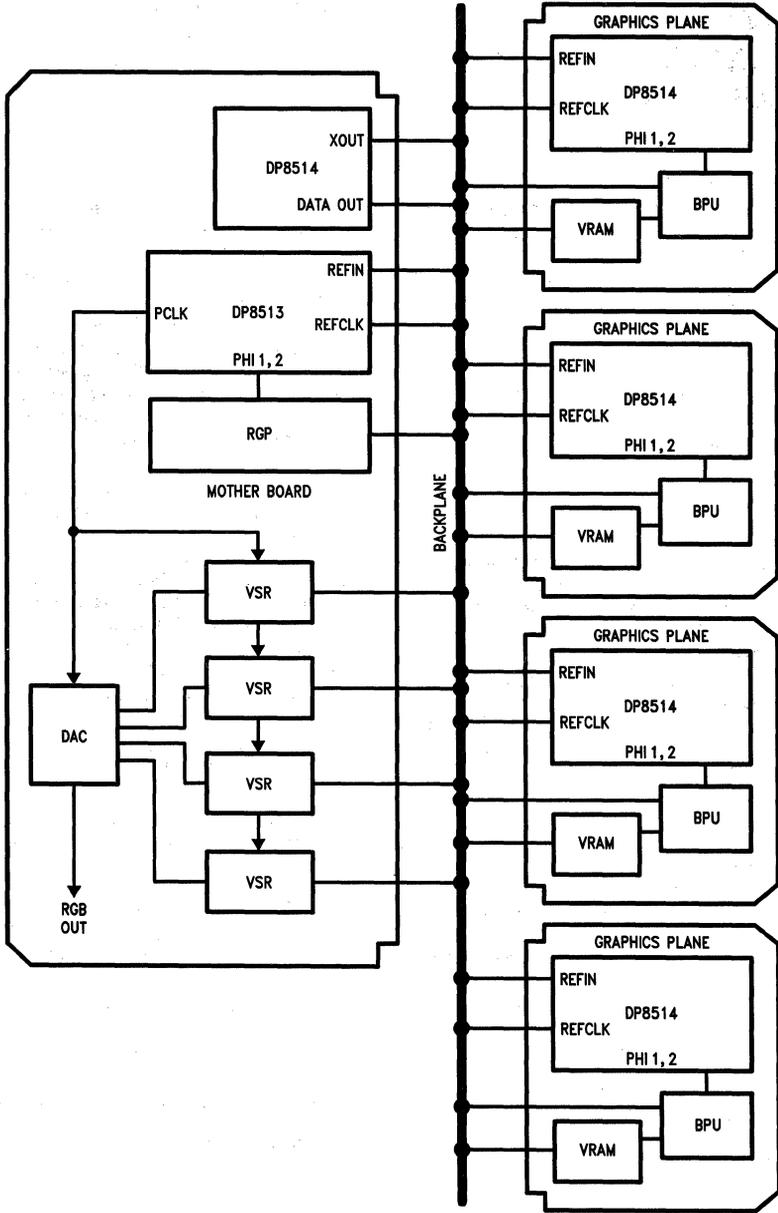


FIGURE 2. Multiple Board Configuration

TL/F/10398-2

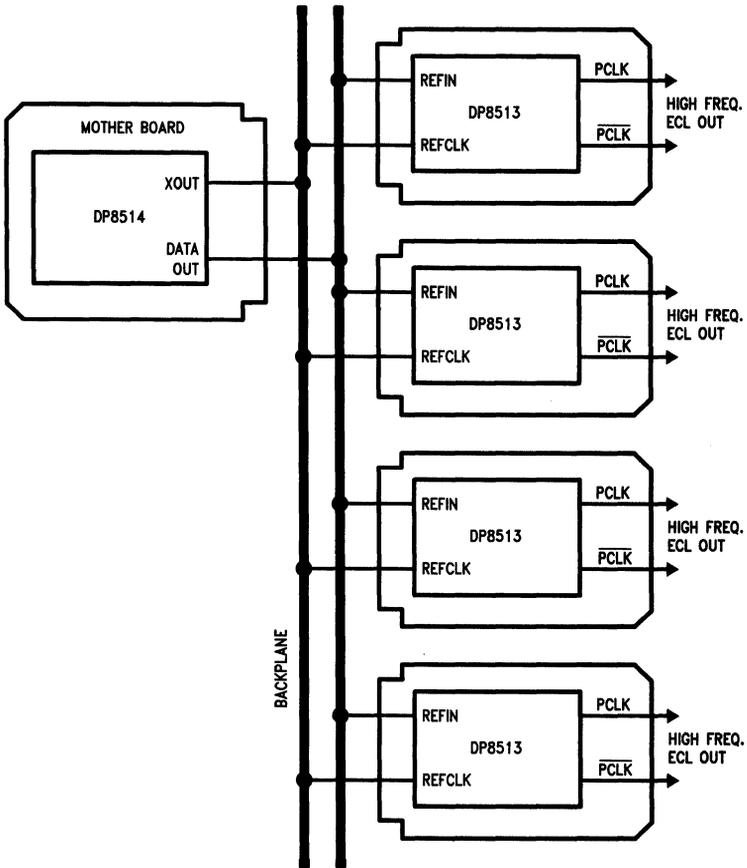


FIGURE 3. Special Multi-Board Configuration Using DP8513 Devices

TL/F/10398-3

CIRCUIT DESCRIPTION

The DP8512/13 devices are digital PLLs containing two phase comparators whose outputs are brought out at OPAMP1 and OPAMP2 respectively (see *Figure 4*). Only OPAMP1 is used in this Evaluation Board. The refer-

ence input to the phase comparator is SOUT, whose origin is either a crystal for the DP8512, or an external source for the DP8513.

3-72

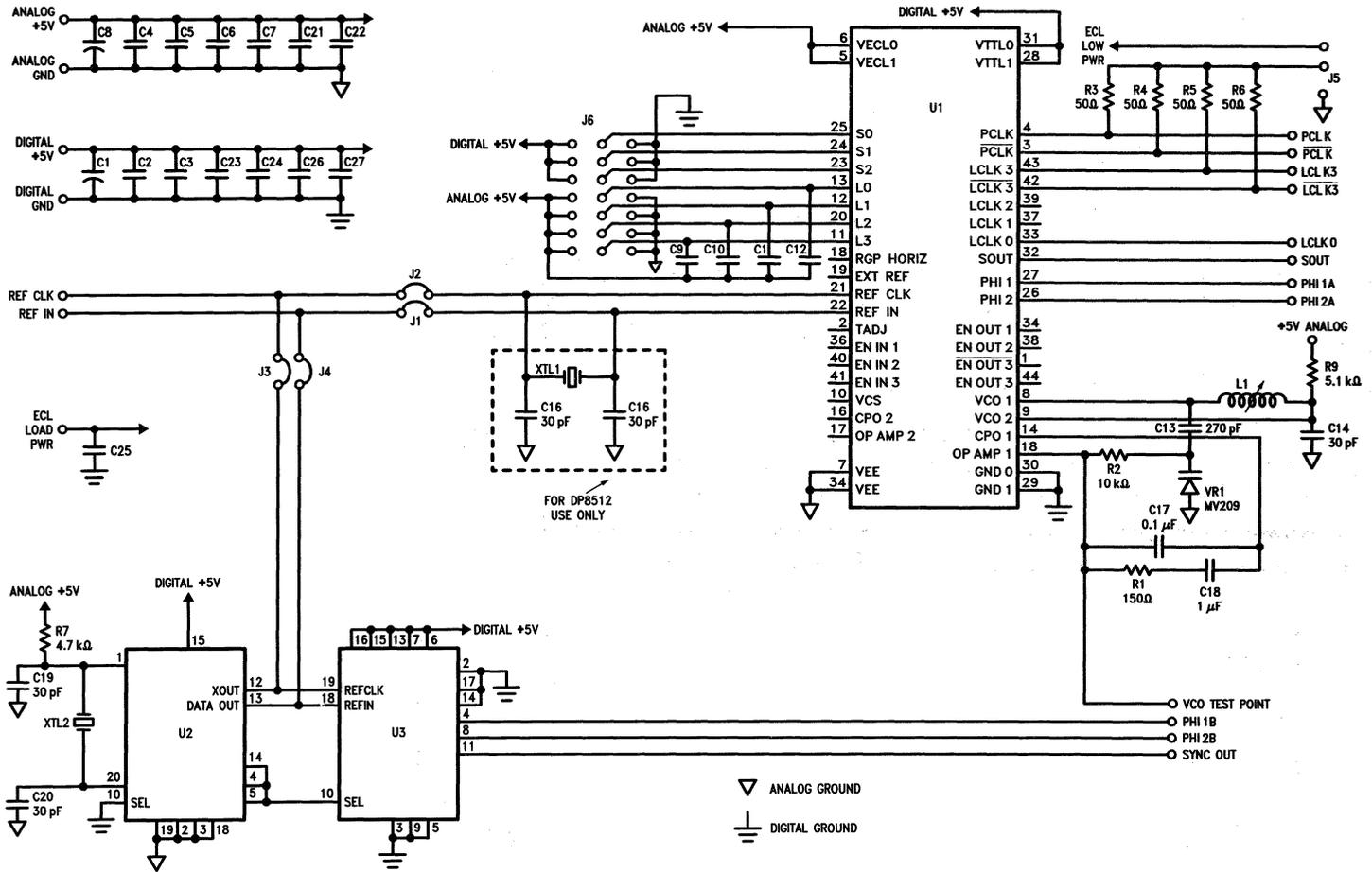


FIGURE 4. Schematic of DP8512/13/14 Evaluation Board

The other input to the phase comparator is LCLK0, which is the divided down VCO signal output of the L Counter. The LCLK0 signal is phase and frequency compared to SOUT. Any phase error between these two signals results in a correction of the voltage into the external VCO control element, varactor VR1, via the OPAMP1 output. This voltage is proportional to the amount of phase error and tends to drive the frequency of the VCO in the direction which, when divided down, minimizes the LCLK0 to SOUT reference signal phase difference. When the phase transition of LCLK0 occurs before that of the reference input (SOUT), the VCO frequency is sensed as being too fast and produces a negative going correction voltage to VR1. This in turn slows down the VCO and delays the following occurrences of the LCLK0's phase transition. The device's PLL architecture makes it possible to generate output clocks with frequency tolerances identical to the crystal reference. In addition, when the reference input signal is a backplane signal, the generated clocks are positive transition phase aligned within ± 2 ns to this reference due to the closely matched delays in the phase comparator.

Both the Primary and the Secondary phase detectors have negative edge sensitive inputs. However, the Primary phase detector inputs appear on LCKO and SOUT via inverters. This gives the appearance that the Primary loop phase detector is positive edge triggered.

Before beginning to describe the circuit, it must be noted that this application board only supports the major functions of the DP8512/13. The Secondary Loop function for Gen locking and various load clock enable functions is not demonstrated.

The DP8514, component U2, functions as a stand alone crystal oscillator. The crystal frequency and half crystal frequency are output on pins XOUT and DATA OUT. These two signals are intended to be sent out on to a backplane and used as a frequency reference by either clock generators or MOS clock drivers located on each of the boards in the system. When using the DP8514 as a frequency reference for the DP8513, it should be noted that the half crystal frequency is the effective input reference frequency. The DP8514, component U3, demonstrates the device used as a synchronizer and MOS Clock driver. The synchronizer is a D register which has a clock input, REFCLK, a data input, REFIN and a mode control input, SEL, which allows the REFCLK input to control the synchronizer's output. This feature allows either the clock or the resynchronized + 2 clock to be fed to the MOS clock driver. The DP8514's SYNC output is a TTL equivalent output in phase with the PHI1 MOS clock output. The SEL input is grounded on the board to demonstrate the resynchronized clock input mode.

The DP8512/13, component U1, is the Graphics Clock Generator. It utilizes a high frequency VCO (Voltage Controlled Oscillator) phase locked to a crystal reference fre-

quency. L1 is the major element in the VCO. Varactor VR1 and capacitor C14 complete the resonant circuit. The inductance value of L1 is chosen by the user to operate the VCO at the desired pixel clock frequency. When using a DP8512 in the U1 position, U2's reference frequency inputs should be disconnected by pulling out the jumpers J1, J2 and placing the crystal and capacitors C15 and C16 into their positions. Note that the DP8512 crystal frequency is not divided by two as in the case of the DP8513/DP8514 combination.

Varactor VR1 is controlled by OPAMP1, a phase comparator output. The voltage on this pin has a compliance of about 4V. This is sufficient to vary the capacitance of VR1 from 25 pF (4V) to about 40 pF (1.0V). C13 is a DC blocking capacitor. It should not be increased in value much above 470 pF, otherwise it may affect the loop response since R2 and C13 form a low-pass filter. C17, C18 and R1 form the primary loop filter. The value of these components are discussed in the Loop Filter Calculations section.

Jumpers (J6) are provided on the board to program the S and L counters. These jumpers are labeled L0, L1, L2, L3 and S0, S1 and S2. These jumpers program U1 to divide the reference frequency in accordance with Tables I and II. The S counter divides the crystal reference frequency down to a load clock rate compatible with VRAM cycle times. The L counter provides the appropriate division rate for the Video RAM parallel to serial conversion operation. Note that inputs L0 thru L3 are bypassed to Analog +5V by C9 thru C12. This is very important for correct device operation. They should be very close to the device. In some applications noise on supply or VCC lines can cause problems. Capacitors C9 thru C12 eliminate the problem. R3 thru R6 are provided to terminate the ECL pixel and load clocks. J5 allows the termination resistors to be returned to VCC - 2V or ground.

TABLE I. S Counter Division Codes

S MOD	S Counter Inputs		
	S2	S1	S0
1	L	L	L
2	L	L	H
3	L	H	L
4	L	H	H
5	H	L	L
6	H	L	H
7	H	H	L
8	H	H	H

L = TTL Level Low input
H = TTL Level High input

TABLE II. Counter Division Codes

L MOD	L Counter Inputs			
	L3	L2	L1	L0
4	0	0	0	0
8	0	0	0	1
12	0	0	1	0
16	0	0	1	1
20	0	1	0	0
24	0	1	0	1
28	0	1	1	0
32	0	1	1	1
36	1	0	0	0
40	1	0	0	1
44	1	0	1	0
48	1	0	1	1
52	1	1	0	0
56	1	1	0	1
60	1	1	1	0
64	1	1	1	1

0 = V_{EE} or OPEN
1 = V_{ECLD}

LCLK0, SOUT, PHI1A and PHI2A are available for on-board monitoring. During operation, LCLK0 is phase locked to SOUT.

USING THE DP8512/13/14 EVALUATION BOARD

There are three key items to adjust before using the evaluation board. The first adjustment involves setting the VCO to operate at the desired pixel clock frequency. The values of the tank components, L1, C14, VR1, determine the VCO oscillation frequency.

The second item involves setting the desired Video Shift Register word width. The width of the video word determines the modulus of the L counter. For example if the video word is 16 pixels long, then the L counter is set to divide by 16. The modulus of the L counter is settable in increments of 4 bits up to 64. Setting is done on the board by moving the jumpers on J6 (L0, L1, L2 and L3) to either the Logic "HI" or Logic "LO" position.

The third adjustment involves setting the S counter modulus to allow the graphics processor to operate at the highest possible rate which is some multiple of the load clock frequency. The PLL uses the LOAD CLOCK and SOUT frequencies internally as phase detector inputs for the main loop. Phase locked operation occurs when the frequency of these inputs are identical. As mentioned previously, the S counter can be programmed to divide the crystal reference frequency by any number from 1 to 8. For example if we select a 20 MHz crystal for the DP8512, a graphics processor can derive its system clock from the PHI1 and PHI2 outputs at 20 MHz. If the S counter is set to divide by 2, (SO = High, S1 = Low, S2 = Low) then the actual frequency available at SOUT and simultaneously applied to the PLL, will be 10 MHz.

Using the above example, let us choose a VCO frequency of 80 MHz and set the L counter accordingly.

$$f_{VCO} = 80 \text{ MHz}$$

$$f_{SOUT} = 10.0 \text{ MHz}$$

Therefore L counter modulus is $80 \div 10 = 8$. Once this has been done the board is ready to use.

INITIAL BOARD ADJUSTMENT PROCEDURE

Note: Although this procedure is specific to the DP8512/13/14 Evaluation Board, it can be used as a general procedure for setting up or trouble shooting most boards using the Video Clock Generator PLL circuits.

- Apply +5V Power to ANALOG +5V and DIGITAL +5V. For single supply operation, use the traces on the bottom of the board to short the Analog +5V and Digital +5V. Analog and Digital Grounds also have traces which can be shorted together.
- Apply +3V to the ECL Termination resistors (Load Power). Ensure that jumper J5 is shorting the middle pin to the topmost pin. This connects +3V to the common point of R3 thru R6, providing ECL loading for the pixel and load clocks.
- Connect a voltmeter to pin 15, the OPAMP1 output of the DP8512/13. This is the VCO control voltage applied to the varactor VR1. A test point is available on the board right above R1 for this purpose. While monitoring the VCO voltage, adjust L1 until the voltage is about 2.5 V_{DC}.
- If using the DP8513, ensure that J1, J2, C19, C20 and XTL2 are in place, while C15 and C16 are absent.
 - If using the DP8512, remove J1 and J2 jumpers and ensure that XTL1, C15 and C16 are in place.
- Check that SOUT and LCLK0 waveforms are phase locked. The pixel clock outputs will now also be phase locked since LCLK0 is directly derived from the VCO.

IN CASE OF DIFFICULTY

If the board does not operate correctly, check that the S and L counters are programmed correctly. There are several things that can be checked to quickly determine the cause of a problem. Following is a troubleshooting sequence which in most cases will lead to a successful resolution of clock generator locking problems.

- With an external power source, apply 2.5 Vdc to pin 15 of the DP8512/13. This overrides the OPAMP1 pin thus setting the varactor VR1 to its final desired operating condition.
- Check that the frequency of SOUT is the crystal frequency divided by the modulus of the S counter.
- Check the frequency of the LCLK0 output. It should be close to the SOUT frequency. Adjust coil L1 until the LCLK0 frequency is close to the SOUT frequency. If there is no output or an unstable wave form on LCLK0, indications are that the VCO tank circuit is not oscillating. Check the orientation of capacitors C13 and C14. Inductor L1 should be a high Q RF coil. If problems persist, ensure that C9 through C12 are in the circuit.
- Check the frequency and amplitude of the ECL PCLK outputs. They should be about 0.8V peak to peak, at the appropriate pixel rate. If not, check that the ECL outputs are properly terminated and that J5 is in the correct position. The pixel rate is SOUT times the L counter modulus.
- If L1 does not adjust to the correct frequency, check that C14 is connected and is the correct value (C13 should remain at 270 pF). Check that L1 is the correct inductance for the chosen pixel frequency.

6. When using the DP8513 and DP8514 combination, the XOUT output of the DP8514 can overshoot below ground due to extraneous lead inductance. This may cause erroneous phase lock. Placing a 100Ω series resistor and a 5 pF capacitor to ground will cure this anomaly (the capacitor connects to the REFIN side to ground).
7. Remove the 2.5 Vdc supply from pin 15 and connect in its place an oscilloscope. Adjust L1 until the OPAMP1 voltage is approximately 2.5 Vdc . LCLK0 should now be phase locked to SOUT. If the OPAMP1 voltage on pin 15 is stuck at 4V , either the coil has too much inductance or the L counter is set to a modulus that is too high. Conversely, if the VCO voltage is stuck at around 300 mV , the coil does not have enough inductance, or the L counter modulus is set too low. If the OPAMP1 voltage oscillates, the loop compensation components may be in error. Consult the Loop Filter Calculation section C18, the capacitor in series with R1, should be approximately 10 to 15 times the value of C17).

Shown in Figures 5, 6, and 7, are the typical waveforms.

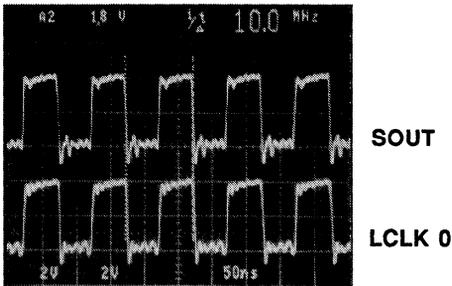


FIGURE 5. SOUT vs LCLK0 Waveforms

TL/F/10398-5

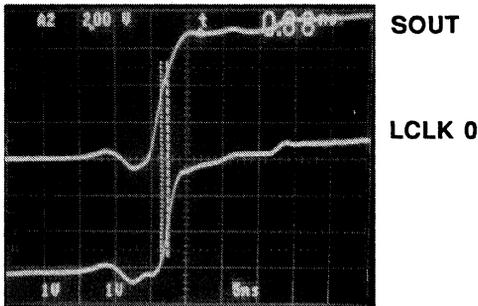


FIGURE 6. SOUT vs LCLK0 Close Up
(Note absence of any phase jitter)

TL/F/10398-6

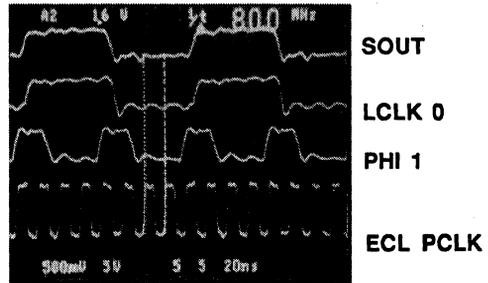


FIGURE 7. Typical Output Waveforms

TL/F/10398-7

DP8512 LOOP FILTER CALCULATIONS

Several constants need to be known in order to determine the loop filter components. They are the loop divide ratio N , the phase detector gain K_p , the VCO gain K_o , the loop bandwidth ω_o and phase margin ϕ .

The constant K_p is fixed at $80\ \mu\text{A}/\text{rad}$ for the DP8512. N is simply the L counter modulus for the main loop. For the secondary loop, N is the S counter modulus times any external division between the SOUT pin and the RGP HORIZ pin (i.e., if $S = 1$ and there is a division by 100 counter between SOUT and RGP HORIZ, $N = 1 \times 100$). A 60° phase margin is recommended, however the equations allow other values to be used if desired.

The oscillator gain constant K_o can be obtained from the VCO components table or determined experimentally. This is done by driving R2, the $27\text{ k}\Omega$ resistor which normally connects the varactor to the OP AMP output, with an external power supply. Set the supply to $\text{VEE} + 3\text{V}$ and note the PCLK frequency. Next, set the supply to $\text{VEE} + 2\text{V}$ and note the frequency again. The difference in these two frequencies (times 2π to convert to radians) is K_o . For optimum performance the desired PCLK frequency should be somewhere between the two frequencies measured above. This may require adjustment of the coil.

Before choosing a value of ω_o , one fact should be pointed out: R2, the $10\text{ k}\Omega$ resistor and C13, the 270 pF coupling capacitor between coil and the varactor, form a low pass filter with a cutoff frequency of about 60 kHz . Thus the loop bandwidth must be chosen to be less than this value. We recommend $(1000\text{ Hz to }30\text{ kHz times }2\pi)$ for ω_o .

Having found all these constants, the following equations are used to find the component values:

$$\begin{aligned} R1 &= 1.08 N \omega_o / K_p K_o \\ C18 &= 3.46 K_p K_o / N \omega_o^2 \\ C17 &= 0.27 K_p K_o / N \omega_o^2 \end{aligned}$$

To use a phase margin of other than 60° use the following:

$$\begin{aligned} R1 &= (N \omega_o / 2 K_p K_o) (\text{cosec } \phi + 1) \\ C18 &= (2 K_p K_o / N \omega_o^2) (\tan \phi) \\ C17 &= (K_p K_o / N \omega_o^2) (\sec \phi - \tan \phi) \end{aligned}$$

Example: Design a system with the following characteristics:

- External horizontal synch of 76.8 kHz
- 1024 pixels per line (1280 pixels including retrace)
- 8-bit wide video data
- 20 MHz processor rate
- 1024 X 1024 Monitor resolution
- 20 MHz processor clock rate
- 60 Hz refresh rate

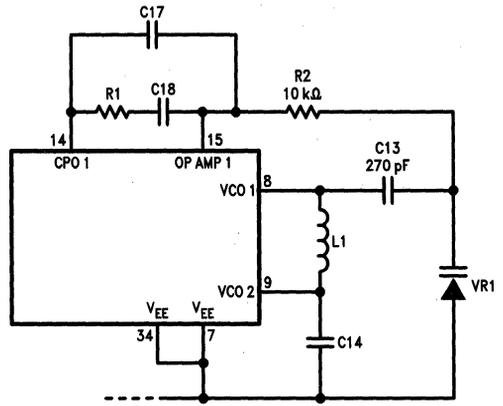
Note that this system will sync to an external source so that both loops must be used.

The PCLK frequency will be:

$$\begin{aligned}
 &60 \text{ Hz (Refresh rate)} \\
 &\times 1024 \text{ (Horiz)} \\
 &\times 1024 \text{ (Vert)} \\
 &\times 1.128 \text{ (H retrace)} \\
 &\times 1.128 \text{ (V retrace)} \\
 &= 80 \text{ MHz}
 \end{aligned}$$

The components in Figure 8 will be used. Note that $K_o = 19 \text{ Mrad/V}$. Because it is an 8-bit wide system the L counter modulus must be 8. By choosing $\omega_o = 2\pi$ times 30 kHz the equations give:

$$\begin{aligned}
 R1 &= 150\Omega, C18 = 1.0 \mu\text{F} \text{ and } C17 \\
 &= 0.08 \mu\text{F} \text{ (use } C17 = 0.1 \mu\text{F)}
 \end{aligned}$$



TL/F/10398-B

FIGURE 8. Circuit of VCO External Components

The Secondary Loop filter component values will be determined in a similar fashion. Note that the value for K_o for a crystal or ceramic resonator VCO are markedly smaller. In choosing ω_o , it should be noted that ω_o for the secondary loop should be smaller than ω_o for the primary loop, so that the main loop will be able to track the secondary without losing lock (Table III shows various C values versus pixel frequencies).

VCO INDUCTANCE VALUES FOR DP8512/13

$$FVCO = \frac{1}{2\pi \sqrt{LC_{tot}}} \quad C_{tot} = \frac{1}{\frac{1}{C14} + \frac{1}{CVR1}}$$

TABLE III. L and C Values versus Pixel Clock Frequencies

Frequency	L (μH)	Suggested Vendor	Part No.	C14 (pF)	Varactor VR1 (Motorola)	K_o Mrad/V
20	4.2	DELEVAN	9405-20	30	MV209	12
40	1.0	DELEVAN	9405-12	30	MV209	14
60	0.465	COILCRAFT	146-10J08	30	MV209	16
80	0.264	COILCRAFT	146-06J08	30	MV209	19
100	0.120	COILCRAFT	146-04J08	30	MV209	21
120	0.090	COILCRAFT	150-03J08	30	MV209	31
140	0.075	COILCRAFT	150-02J08	30	MV209	27
160	0.066	COILCRAFT	150-02J08	30	MV209	27
180	0.070	COILCRAFT	150-02J08	15	MV2205	26
200	0.037	COILCRAFT	150-01J08	15	MV2205	24
220	0.037	COILCRAFT	150-01J08	15	MV2205	34

COMPONENT SOURCES:

DELEVAN— AMERICAN PRECISION INDUSTRIES
 Electronic Components Group
 Delevan Division
 270 Quaker Road, East Aurora, NY 14052-0449
 Phone (716) 652-3600 Telex 91-293

COILCRAFT— COILCRAFT
 Cary, Illinois 60013
 Phone (312) 639-2361 TWX 910-651-2251

APPENDIX

DP8512/13 PCB LAYOUT CONSIDERATIONS

1. Use separate V_{CC} and GND branches for the analog (VECL 0, 1, V_{EE}) versus digital (VTTL 0, 1 and GND 0, 1) supply pins. Use of the ferrite beads between the branches is also highly recommended.
2. Bypass supplies right at the device. (Preferably right under the device using surface mount components.)
3. Ground unused inputs to reduce noise generation (eg. ENIN1, ENIN2).
4. Bypass L0, L1, L2, L3, to VECL0 close to the device.
5. Mount VCO coil L1, varactor VR1, C13 and C14 right next to VCO1, VCO2 pins. Ground for VR1 and C14 should return to a single point at V_{EE} pin 7.
6. Place varactor diode and associated DC blocking capacitor (270 pF) right next to the VCO coil.
7. Place capacitor from VCO2 to V_{EE} right next to VCO2 and VCO coil.
8. If using multi-layer layout with dedicated V_{CC} and Ground planes, ensure that:
 - a. For power, VECL1 and VECLO should have their own small isolated ground plane (Analog V_{CC}).
 - b. For ground, the two V_{EE} pins should have their own small isolated ground plane (Analog GND).

ADDITIONAL DP8512 APPLICATIONS

1. External Drive of VCO input.

Parasitic capacitance limits the upper frequency of the internal Pierce oscillator on the DP8512 to approximately 225 MHz. In order to operate above this frequency, an external oscillator must be capacitively coupled into the oscillator pins. When driven externally, operation to 300 MHz can be achieved.

The circuit in *Figure 9* can be used to drive the VCO input of the DP8512. Note that the VCO1, the base of the internal oscillator transistor, (see VCO INPUTS diagram, page 11 of DP8512 data sheet) must be pulled up to VCO2, the collector. A suggested value of resistance is 2.2 k Ω . The drive level into VCO2 should be in the range of -10 to 0 dBm.

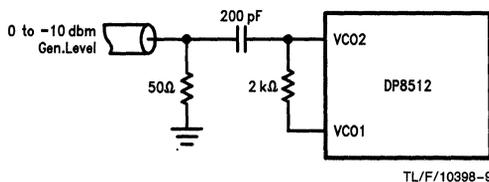


FIGURE 9. External Drive of DP8512 VCO

2. External Drive of VCXO input.

The DP8512 contains an internal oscillator designed to be used as a master clock from which the other frequencies are derived. This oscillator is generally used with an external crystal. This crystal may stand alone, or be phase locked to another source via the Secondary Loop within the DP8512. Locking is accomplished via a varactor diode within the crystal oscillator circuit (VCXO).

If the desired degree of frequency shifting lies beyond the range of the crystal, then other oscillator elements must be considered. An alternative to a crystal is a ceramic resonator. Conventional ceramic resonators are not suitable for VCO applications due to their spurious anti-resonant oscillation modes. Special low Q ceramic resonators are necessary in this application.

If desiring to use an external oscillator, special consideration must be given to interfacing the VCXO pins, namely XLTB and XTLC (pins 21 and 22). These pins are not TTL compatible. It is recommended that the XTLC pin be driven, while leaving XLTB open. The input signal must be in the range of 400 mV to 600 mV peak-to-peak. The circuit in *Figure 10* is suggested.

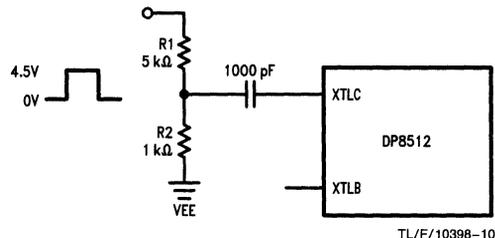


FIGURE 10. External Drive of DP8512 Crystal OSC. Inputs

Note that R1 and R2 form a voltage divider that reduces a TTL signal to an acceptable level. When combined with the effective input impedance of the XTLC input, the resulting signal level is approximately 500 mV p-p at 10 MHz, the frequency at which the above circuit was tried.

Several precautions are necessary in order to ensure that extraneous noise not be injected into this input. If an external reference signal is utilized, precautions should be taken to ensure that extraneous noise is not injected into the part as a result of dissimilar ground references. The internal oscillator is referenced to VEE and consequently R2 should be placed close to the DP8512's VEE pins. The circuit shown in *Figure 10* will provide approximately the same effective signal into the DP8512 XTLC input whether the device is being used in either common or split supply mode of operation.

3. Loop Jitter Considerations

The most critical layout area of the board is related to the VCO. It is very important that the analog supplies to which the VCO components are connected are isolated, as well as possible, from any digital noise sources. Digital noise impressed on the VCO signals has recently been observed to induce phase modulation onto the VCO's output frequency. Under certain conditions relating to layout, frequency of operation, power supply bypassing, and output loading, the digital noise may cause the loop to exhibit some amount of jitter. This can be observed by viewing LCLK0 on an oscilloscope while triggering the scope from SOUT. The result will be a blurry trace. Careful attention to proper supply bypassing and good RF layout techniques may eliminate the jitter. If it does not, the external VCO circuit shown in *Figure 11* can be used to further decouple the VCO from the power supply and reduce the jitter.

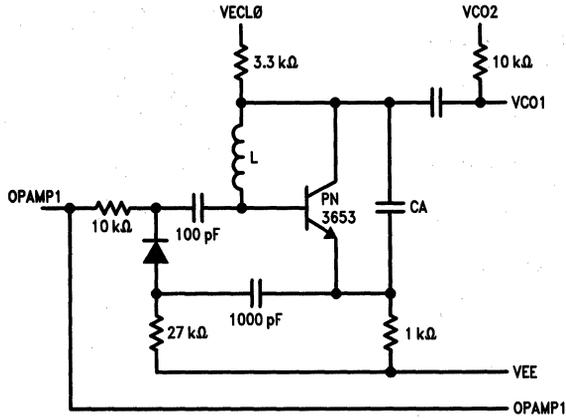


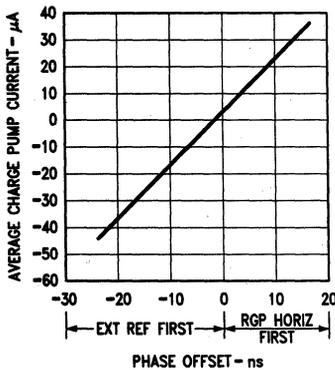
FIGURE 11. External VCO Circuit

TL/F/10398-11

The PN3653 is a high frequency oscillator transistor. To reduce component count the 1000 pF capacitor and the 27 kΩ resistor may be eliminated by simply returning the anode of the varactor to the emitter of the transistor. This will result in a control voltage range reduction of about 1V which may or may not be acceptable depending on system requirements.

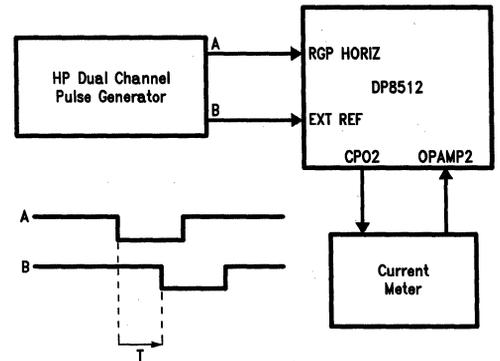
DP8512 PHASE DETECTOR/CHARGE PUMP LINEARITY TEST

Figure 12 demonstrates the zero dead band characteristics of the DP8512/13 phase detector or charge pump circuitry. Note the linearity at the zero phase offset crossover point. When the device is locked, the phase jitter is imperceptible.



TL/F/10398-12

FIGURE 12. DP8512/13 Phase Offset versus Charge Pump Current



TL/F/10398-13

FIGURE 13. DP8512/13 Phase Detect/Charge Pump Linearity Test Setup

DP 8512/13/14 EVALUATION BOARD CIRCUIT LAYOUT

Following are illustrations of the DP8512/13/14 Evaluation Board printed circuit board (see *Figures 14, 15 and 16*).

Shown also are the PCC Socket diagrams viewed from the bottom side (see *Figures 17 and 18*). A parts list is included in Table IV.

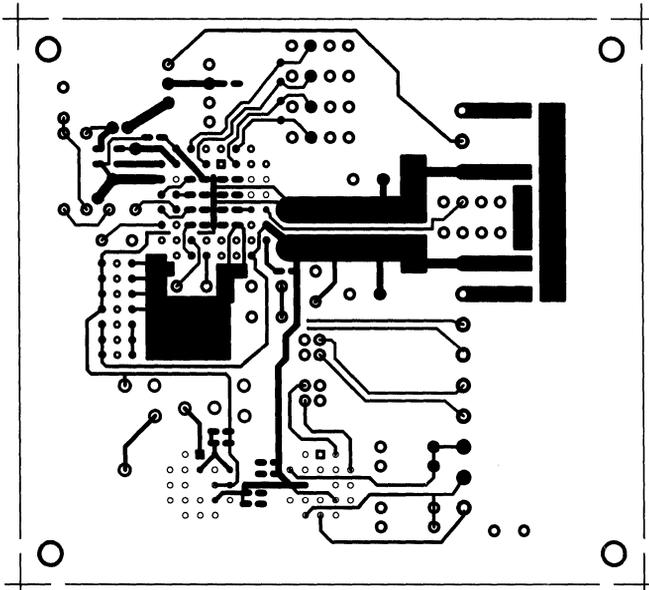


FIGURE 14. Solder Side, DP8512/13/14, Evaluation Board

TL/F/10398-14

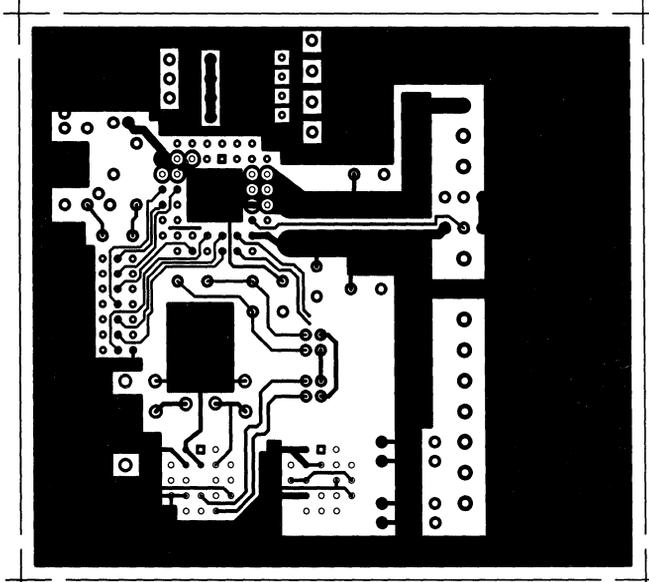


FIGURE 15. Component Side, DP8512/13/14, Evaluation Board

TL/F/10398-15

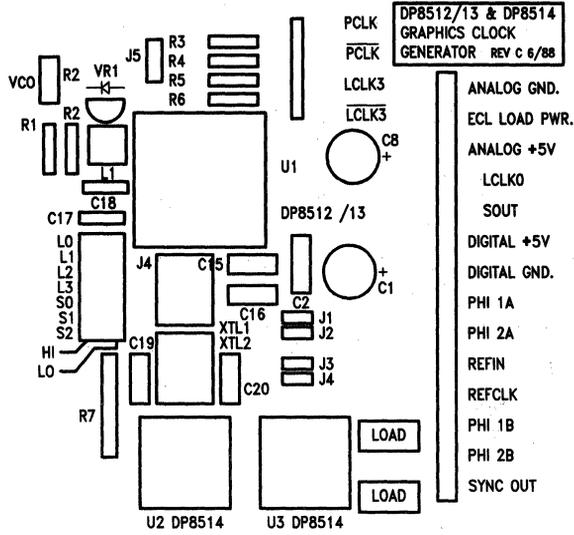
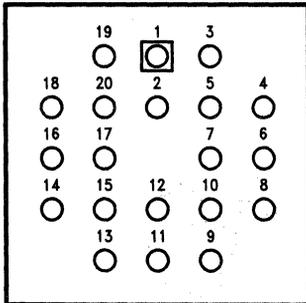


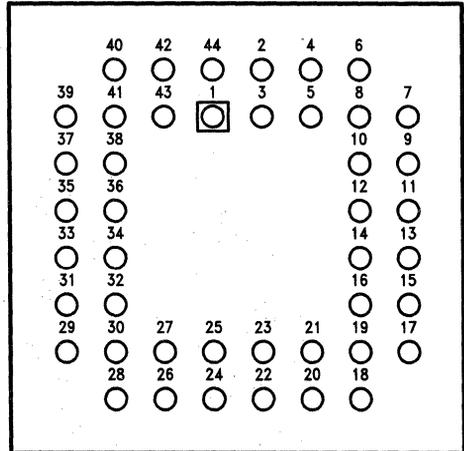
FIGURE 16. Silk Screen, Component Side, DP8512/13/14, Evaluation Board

TL/F/10398-16



TL/F/10398-17

FIGURE 17. Bottom View, Pinout DP8514



TL/F/10398-18

FIGURE 18. Bottom View, Pinout DP8512/13

TABLE IV. Parts List

DES	Description	Part No.	Suggested Vendor
L1	See Table	See Table III	See Table III
U1	DP8512 or DP8513		National Semiconductor
U2	DP8514		National Semiconductor
U3	DP8514		National Semiconductor
XTL1	40 MHz Crystal	NE-18 B	NEL Frequency Control Inc. 357 Belliot St.
XTL2	40 MHz Crystal	NE-18 B	Burlington, WIS. 53105 (414) 763-3591
R1	270 Ω 1/4W 5% Res.		
R2	27 k Ω 1/4W 5% Res.		
R3	51 Ω 1/4W 5% Res.		
R4	51 Ω 1/4W 5% Res.		
R5	51 Ω 1/4W 5% Res.		
R6	51 Ω 1/4W 5% Res.		
R7	4.7k 1/4W 5% Res.		
R9	5.1k Chip Res.		
C1	10 μ F Electrolytic		
C2	0.1 μ F Ceramic Cap.		
C3	0.01 μ F Chip Cap.		
C4	100 pF Chip Cap.		
C5	1000 pF Chip Cap.		
C6	0.01 μ F Chip Cap.		
C7	0.1 μ F Chip Cap.		
C8	10 μ F Electrolytic		
C9	0.01 μ F Chip Cap.		
C10	0.01 μ F Chip Cap.		
C11	0.01 μ F Chip Cap.		
C12	0.01 μ F Chip Cap.		
C13	470 pF Chip Cap.		
C14	30 pF Chip Cap. NPO		
C15	30 pF MICA Cap.		
C16	30 pF MICA Cap.		
C17	0.01 μ F Ceramic Cap.		
C18	1.0 μ F Ceramic Cap.		
C19	30 pF Ceramic Cap.		
C20	30 pF MICA Cap.		
C21	0.1 μ F Chip Cap.		
C22	0.01 μ F Chip Cap.		
C23	0.01 μ F Chip Cap.		
C24	0.1 μ F Chip Cap.		
C25	0.1 μ F Chip Cap.		
C26	0.1 μ F Chip Cap.		
C27	0.01 μ F Chip Cap.		

A Graphics Acceleration Card for the AT Using the Advanced Graphics Chip Set

National Semiconductor
Application Note 609
J. Margeson



1.0 OVERVIEW

Members of the Advanced Graphics Chip Set (AGCS) are used in the design of an AT graphics card. The purpose of the design is to use a high-performance graphics microprocessor and coprocessors on the board to improve the AT's performance for graphics applications.

2.0 ARCHITECTURE

A graphics processor can be added to the AT's architecture to improve graphics performance. It is given the task of manipulating the pixels, freeing the main CPU to spend more processing time on other aspects of applications. In an AT using one of the standard graphics cards, the graphics frame buffer appears as an array in the CPU's memory map. To render graphics primitives, the main CPU must calculate the algorithms for the primitives and write the pixels into the frame buffer, accessing them one at a time. Hardware on the card generates the video timing and collects the data from the array to generate video signals for the monitor.

Figure 1 is the system architecture using the 4-plane board. The DP8500 Raster Graphics Processor (RGP) and DP8511 BITBLT Processing Units (BPU) are used to render text and other objects in the frame buffer and generate the video signals. The RGP is optimized for rendering the graphics primitives; the BPU's allow parallel execution of the algorithms in each bit plane. In a 4-bit per pixel design such as this one, images can be drawn at rates of up to 64 bits (16 pixels) per memory access cycle. The number of pixels per cycle is constant for a RGP/BPU system independent of the number of bit planes. For example, a system with 32 bits per pixel renders at up to 512 bits per memory cycle, a rate that is difficult to attain with a general-purpose CPU architecture.

The 4-plane board's memory is dual-ported to allow the RGP and main CPU to use it as a communication area. Upon power-up, the main processor resets the board and loads the RGP code into its memory. The RGP executes its ROM monitor after a reset has occurred; the main CPU tells it to branch to the start address of the downloaded code. The ROM monitor is also used with the debugger, which runs on the AT. In a typical application, the host CPU gives display lists to the RGP by writing them into its memory. After deposit of the display list the main CPU can resume execution of the application, while the RGP renders the objects in the screen buffer.

Note that the host still has direct access to the screen buffer, as in the standard architecture, allowing use with applications that do not take advantage of the graphics processor. Images can also be downloaded into the screen buffer directly from disk.

3.0 FUNCTIONAL BLOCKS

The following sections explain the design of each of the functional blocks in the architecture. Figures 14a through 14m are the schematics. Figures 15 through 31 are the equations for the GALs. Figure 32 is the parts list.

3.1 Clock Generator

The DP8512 Video Clock Generator (U2), with Y1, C3, C4 and U3, generates the video and bi-phase CPU clocks. The DP8512 data sheet explains this circuit in more detail. The RGP and BPU's used the 20 MHz bi-phase clock (PH1 and PH2). The clocks are buffered by U3 to go to other sections of the board. CK1A is used by the bus arbiter/timing generator and video plane controller. The parallel interface sequencer is clocked by CK1B. CK1C is used as the 20 MHz pixel clock. In other designs, the pixel clock could be much higher; it is not limited by the RGP. CK2 clocks the registered PROMs and is used by U6 to generate the VDRAM row/column select, RSEL. The SOUT output is divided down to 1.25 MHz for use as LCK.

3.2 RGP

The DP8500 Raster Graphics Processor (RGP) executes graphics rendering tasks given to it by the main CPU and generates the video timing. The RGP asserts ALE to request a memory cycle; BS1, BS0, R and W indicate the type of cycle. It fetches instructions, accesses operands, uses the BPU's to execute drawing cycles, and loads the shift registers in the VDRAMs (video dynamic random access memories) with video data for display refresh. Execution of these memory cycles is discussed in the Bus Arbiter/Timing Generator section.

The lower 16 address lines are latched by U7 and U8 on ALE. U11 buffers the upper 8 address lines. U9 and U10 buffer the data lines. These five I.C.'s are in TRI-STATE® when the AT is accessing the RGP's memory.

Video timing is generated from LCKL. Rising edges of LCKL are counted inside the RGP to generate DRREQ (display refresh request), VSYNC, HSYNC and BLANK, according to the software programmable video parameters. LCK, in this design, is 1.25 MHz (800 ns), representing 16 pixels; the video timing can be adjusted in software to this resolution.

The HOLD signal is tied to VCC by jumper W1. This jumper should be in place, unless hardware modifications are made to the board.

The details of the RGP operation are discussed in the DP8500 data sheet.

3.3 Parallel Host Interface

The AT can read and write devices in the 4-plane board's memory map. To execute one of these memory cycles, it asserts the proper address and strobe. The AT interface state machine handles the bus protocol; it holds the AT and requests to the memory arbiter for a remote access memory cycle. Data is transferred when the arbiter grants the cycle, then the AT is released. The AT bus does not have 16M words of space to map the board into, so segment registers are required to hold the upper portion of the on-board address. There are also control registers for reset, interrupts and status.

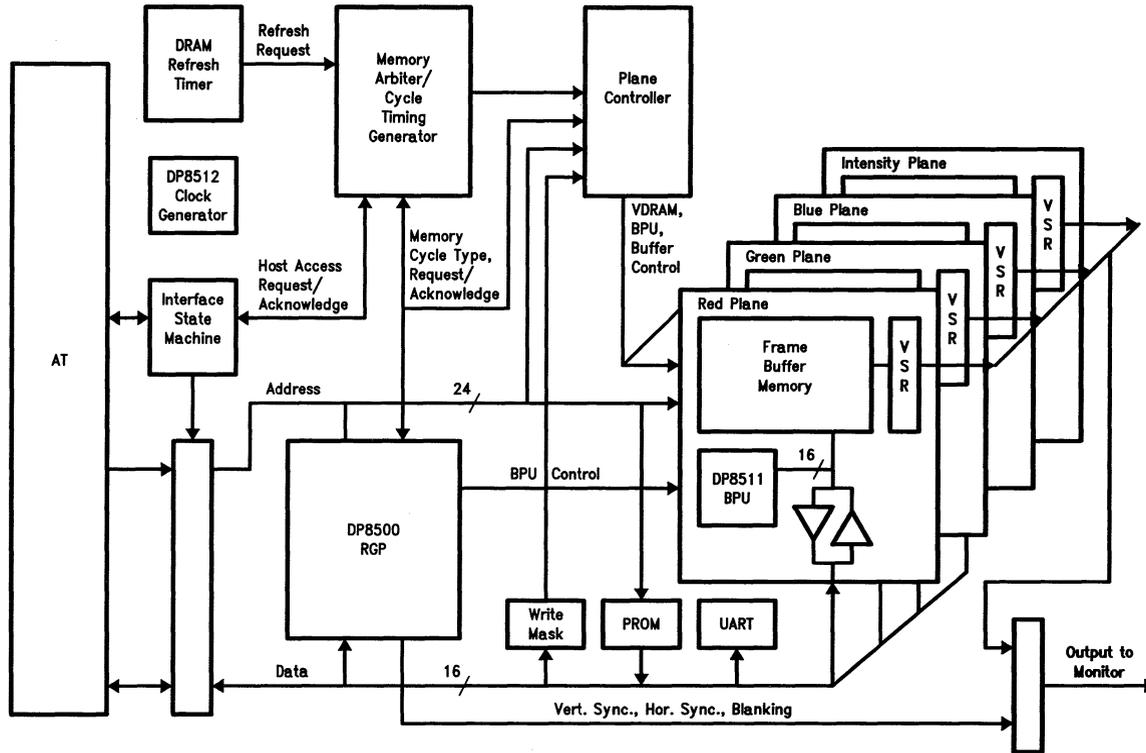


FIGURE 1. System Architecture

TL/F/10429-1

U220 generates $\overline{\text{BDS}}\text{ELL}$ (Board Select Latched, active low) and $\overline{\text{M}}16$ (Memory Select 16-bit, active low) from AEN and the upper eight address lines of the AT bus. $\overline{\text{BDS}}\text{ELL}$ is active when AEN is low and the proper address is asserted and is latched by BALE. The GAL is used as a transparent latch in this case by using BALE as an enable and feeding back $\overline{\text{BDS}}\text{ELL}$. The board resides in a 64k byte area in the AT memory map, the beginning of which is set by jumpers W2, W3 and W4. The base address can be set to 0A0000 (W2), 0D0000 (W3), A00000 (W4) or any combination (there can be multiple images of the board if desired). It is recommended that the board not be used at base address 0A0000 or 0D0000. The board can only accept 16-bit accesses; byte accesses result in errors. AT address line 0 must be "0" for word transfers. If an alternate base address is desired then U220 can be reprogrammed.

The AT's processor, in the mode used by DOS, can only access the first megabyte of the 16 megabytes addressable by the bus. The 4-plane card can reside above 0FFFFFFH if one of three schemes is used to access it: the AT's processor is switched into protected mode (allowing it to assert 24 address lines instead of 20), DMA transfer is used, or a co-processor card is used that is capable of bus mastership. In these cases, the board can be set to reside at A00000H-A1FFFFH (a double image).

The interface is designed to execute 16-bit transfers to maximize the data transfer rate. The $\overline{\text{M}}16$ signal is generated combinatorially from the upper seven address lines. This signal tells the AT that a 16-bit transfer can be executed. If it is asserted too late, the access is split into two eight-bit accesses. Address line 16 can not be used to generate $\overline{\text{M}}16$ because it becomes valid much later than the upper seven address lines. In cases where the board is set to reside in the lowest megabyte of the AT bus space, address line 16 must be ignored, requiring that the neighboring 64k byte area have a sixteen bit device mapped into it or no device at all. If the board resides in the upper 15 megabytes, then a 128k byte space is used (a double image in memory), and address line 16 is ignored.

When accessing the 4-plane board, 24 address lines must be asserted. The AT can not generate all of them during a read or write cycle, so the upper ten lines are held in a segment register. There are two segment registers on the board: one moveable pointer and one fixed pointer. The moveable segment pointer is set before the access is made. AT data line 0 corresponds to RGP address line 14, through AT data line 9 which corresponds to RGP address line 23. At the time of the access, the upper ten lines are driven by the register (U19); the lower 14 lines come from the AT bus, a segment size of 16k words. This register is also readable by the AT (U20). The board does not use RGP address lines 16 and 19 which correspond to AT data lines 2 and 5. When the register is read, data lines 2, 5, and 10 through 15 are undefined and must be masked if the address is to be used in any calculations. However, the register can be restored exactly as it was read because the board ignores these bits. For multiple accesses within the same segment, the segment pointer only needs to be set once. The offset into the area is determined by the lower AT address lines. AT address line 1 corresponds to RGP address line 0, on up to AT address line 14 which corresponds to RGP address line 13. This scheme allows the AT to write to any memory location on the board.

The fixed segment register points to a 8k word contiguous area mapped onto the end of memory plane 3, RGP addresses 46E000-46FFFF. Using the fixed segment area eliminates the overhead of setting the pointer and allows multiple software tasks access the board without needing to save and restore the pointer. The monitor and debugger use this as a communication area. The fixed segment pointer drives the upper eleven address lines; the lower 13 lines come from AT address lines 1 through 13. Address lines 19 and 16 are not used on the board, thus are not driven.

The control and status registers are explained in the AT bus memory map. The AT can reset the board, assert its non-maskable interrupt and maskable interrupt and examine its status. The RGP can deassert the maskable interrupt.

The RGP can interrupt the AT by writing to a register in its memory map. The AT can deassert the interrupt. W10-W21 select the interrupt to be used.

AT Bus Memory Map:

NN0000 is the base address as selected by W2, W3 and W4.

NN0000-NN7FFE: Moveable Data Area. The AT address can be derived in a C program by:

```
AT_addr := ((RGP_addr < 1) &
0x7ffe) + 0x00NN0000;
```

Read and Write.

NN8000-NNBFFE: Fixed Data Area. The AT address can be calculated by:

```
AT_addr := ((RGP_addr < 1) &
0x3ffe) + 0x00NN8000;
```

Read and Write.

NNC000: Segment Pointer. The segment pointer can be calculated by:

```
segment_pointer := (rgp_addr > 14) & 0x03FF;
```

Read and Write.

NNC800: $\overline{\text{NMI}}$ Register. Any write to this location temporarily asserts the RGP's $\overline{\text{NMI}}$ line, which is edge sensitive. Write only.

NND000: AT Interrupt Reset. Any write to this location deasserts the interrupt from the 4-plane board to the AT. It can be used to clear the interrupt when it is serviced. Write only.

NND800: Board Status. The AT can read the present state of four signals on the board:

D3: RGP Reset (RSTI)

D2: RGP Halt (HALT)

D1: RGP Interrupt (INT)

D0: AT Interrupt (ATINT)

Read Only.

NNE000: RGP Reset (RSTI). The RGP can be reset by writing a "1" into this register, followed by a "0". It must be set to "1" for a minimum of 800 ns. Reset initializes the board, causing the RGP to begin execution of the ROM monitor. The default screen will be cleared and the remaining VDRAM will be undefined. Write only.

NNE800:

RGP interrupt (INT). The RGP's level sensitive maskable interrupt can be asserted by writing a "1" into this register. Writing a "0" removes the interrupt request. The RGP also has the ability to reset the request. Write only.

The AT interface sequencer is implemented in U221. Accesses to the control registers are transparent to the interface sequencer; only data transfers requiring a 4-plane memory cycle are handled by the sequencer. A flowchart of the sequencer is shown in Figure 2. The sequencer remains inactive until the host attempts a data transfer to/from the board's memory. In the beginning of the access, the address lines on the AT bus become valid, causing U220 to assert the $M16$ signal. U220 latches \overline{BSEL} on BALE. If the card is being accessed with a DMA cycle or by a coprocessor card, BALE remains high and the upper address lines remain valid throughout the entire cycle.

The read or write strobe follows, causing U223 to asynchronously return $REMRDY$ low (an open collector line). Figure 3 shows the timing of the access. While the AT is being held, the strobe is synchronized to $CK1B$ so that it can be input to the interface sequencer. The strobes are effectively latched twice for metastability: once before the sequencer, and sequencer transitions based on the strobes change only a single bit of the state variable. The first transition from the idle state sends a remote access cycle request to the arbiter ($REMREQ$). The arbiter/timing generator state machine executes a remote access cycle when it is the highest priority cycle requested. It is designed such that the AT is held much less than the maximum $2.1 \mu s$ in the worst case. The time period for the data transfer is delimited by the $REMACK$ signal. In the case of a write, the data is driven onto the 4-plane's data bus by U23 and U24. For a read,

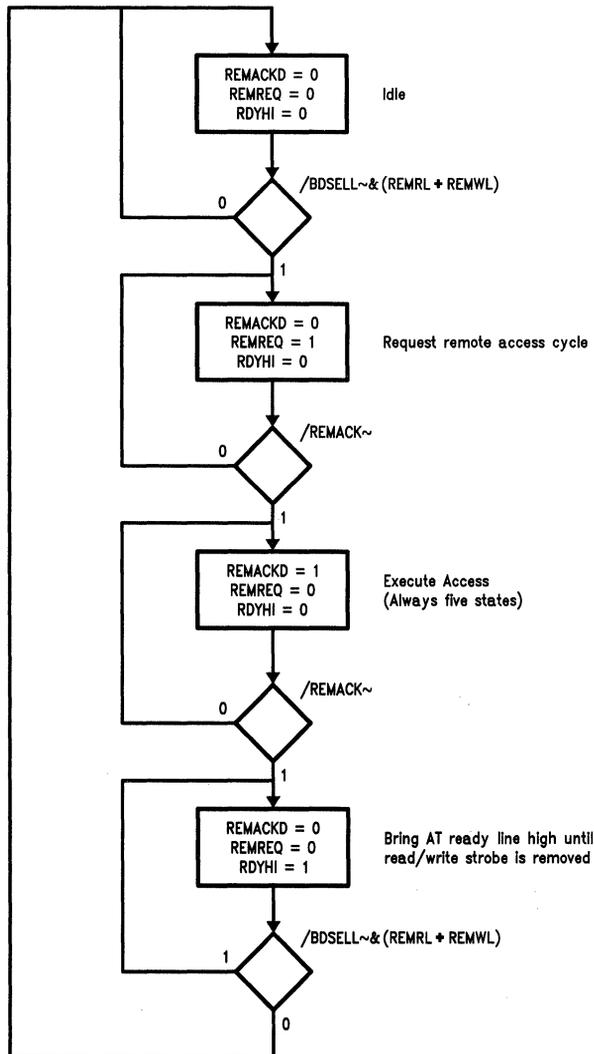


FIGURE 2. AT Interface State Machine Flowchart

TL/F/10429-2

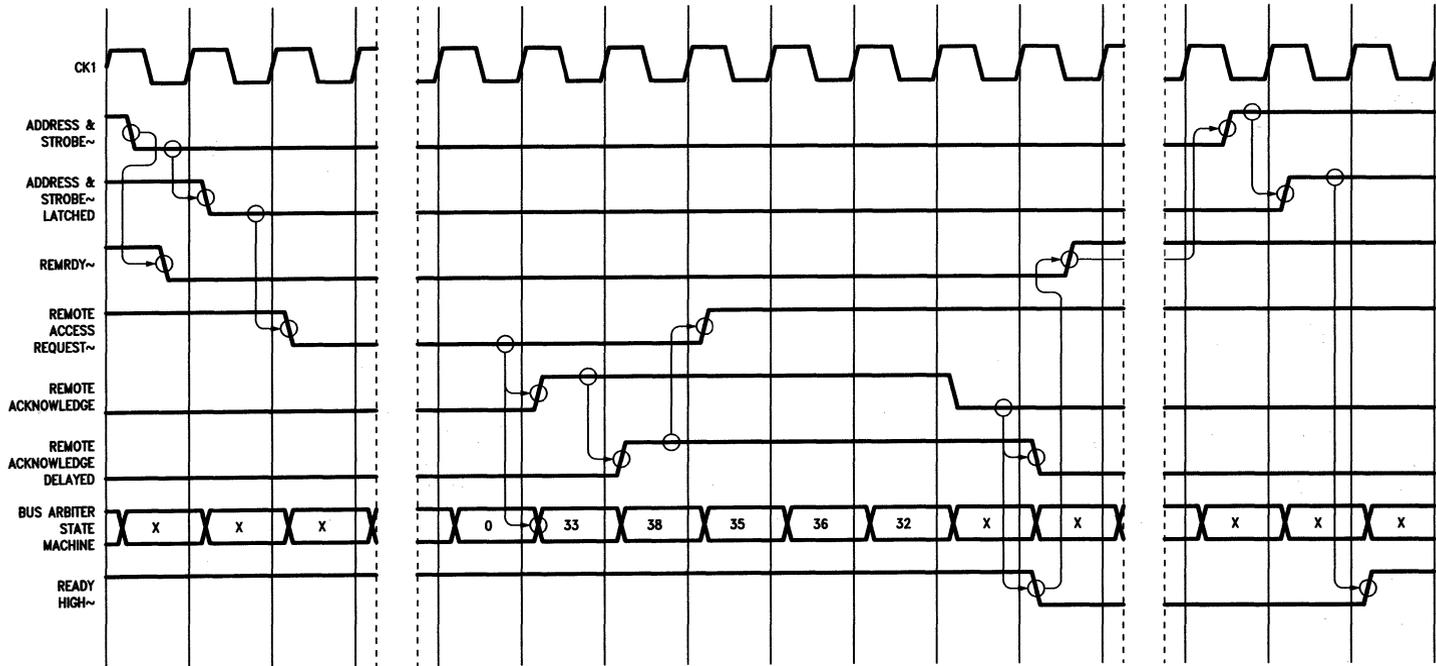


FIGURE 3. Host CPU Access Timing

the data is latched into U23 and U24 on the rising edge of REMACK. After the transfer is made on the 4-plane board, RDYHI is asserted to inhibit the REMRDY signal. The data is driven onto the AT bus until the read strobe is removed. The arbiter/timing generator can continue executing other types of memory cycles once the transfer is completed, before the AT removes its read or write strobe. The interface sequencer returns to the inactive state after the host brings its read or write strobe high.

Accesses to the data area by the AT are longer than most AT bus cycles, so their frequency should be minimized. These accesses also consume 4-plane board memory bandwidth, but will usually represent a small portion of it. They should not significantly reduce the RGP's rendering speed in most applications, because the number of memory cycles required to render objects is typically much greater than the number of accesses needed to transfer display lists. Control register accesses (addresses NNC000–NNFFFE) do not add wait states to the AT bus cycle.

3.4 VDRAM Refresh Generator

VDRAM refresh requests are generated by a counter in U208. LCKL is divided by 20 to request cycles at 16 μ s intervals. U6 holds the request until it is reset during refresh cycle by the timing generator. The request is synchronized in U18 to make setup time for U200.

3.5 Memory Arbiter/Timing Generator

Several actions requiring memory cycles have been discussed. Requests for these memory cycles come from various sources; this circuit services the requests on a priority basis and generates the timing for them.

The arbitration and timing generation is done by a sequencer implemented in U200. Figure 4 is the flowchart of the sequencer. Six types of memory cycles are executed, listed from highest to lowest priority:

- A Display Refresh
- B DRAM Refresh
- C Host Processor Access
- D Instruction/Operand Read or Write
- E1 Drawing Read or Write
- E2 Drawing Read/Modify/Write

Highest priority is given to branch A, the display refresh cycle; it is the most latency-sensitive of the six cycles. This cycle is used to load the shift registers inside of the VDRAMs for video signal generation. The RGP generates the request for this cycle and the address for the load.

The RGP internally arbitrates between display refresh and its processing and drawing cycles. A display refresh cycle request can be queued behind another RGP cycle that is in progress. The worst-case delay for entering the display refresh cycle occurs when the RGP requests a drawing read/modify/write cycle just before it internally generates the display refresh request, preceded by a host processor access request, followed by a VDRAM refresh and a second host processor access request. The AT access is serviced first because the request arrives before the VDRAM refresh. The VDRAM refresh is serviced second, followed by the second AT access, the drawing cycle and then the display refresh. Two VDRAM refresh cycles will not occur in close proximity, which is also true of the display refresh requests.

Once the cycle is entered, the sequencer can be held by the MIDSCAN signal to insure proper timing of the shift register load. If the load is done during video blanking then the cycle

is not held; the shift registers are not clocked during this period and timing of the load is not critical. If the load occurs in the middle of a scan line, then the sequencer is held until after the last shift of the previously loaded data.

The VDRAM refresh cycle, branch B, is second in priority. The refresh requests are generated every 320 clock cycles. The request must be serviced and cleared before another is generated. The worst-case latency occurs when this request is made just after the RGP has started a drawing read/modify/write and it generates a display refresh request following the drawing cycle. This guarantees that RSTRRQ is asserted and deasserted before the next request is generated.

The remote access cycle, C, is third in priority. The AT must be held until the remote access is completed, but the AT bus cycle must not be stretched longer than 2.1 μ s (42 clock cycles). If the bus is held too long then the AT's DMA controller may miss refresh cycles for its DRAMs. The worst case occurs when the AT initiates a remote access just after a read/modify/write cycle has been started, followed by a VDRAM refresh request and a display refresh request. The remote access request will not be serviced until these three cycles are completed; the read/modify/write cycle can not be interrupted after it is started and the other two cycles are higher in priority.

Branches, D, E1 and E2 execute instruction/operand memory accesses, drawing read/writes, and drawing read/modify/writes respectively. These three cycles are used by the RGP for processing and drawing and are effectively equal in priority as it will never request more than one of them at once. They are lowest in priority because they have no particular maximum latency; delaying them only slows RGP code execution.

The VDRAM access time is reduced when consecutive drawing accesses have the same row address. The drawing algorithms tend to operate on memory locations in close groups. The memory page size is 256 words, which is equivalent to 4k pixels or 6.4 scan lines. The RGP generates a page break signal to indicate if a row address boundary has been crossed. At the end of a drawing cycle, the sequencer can loop back into another drawing cycle of either type, keeping $\overline{\text{RAS}}$ low and reducing the cycle time by 100 ns. This will occur if: a drawing cycle is requested, it is in the same page and there are no requests for higher priority cycles. A display refresh cycle request will cause the RGP to break up the page of drawing accesses, because the arbitration in this case is done inside the RGP. Request of a VDRAM refresh or host access causes the arbiter to force a page break.

The AT interface sequencer requests a host access cycle by bringing REMREQ low. REMACK is returned to indicate when the transfer takes place. While REMACK = low, the buffers driving the address and data lines from the RGP are disabled so that the AT interface circuitry can drive them. REMACK also controls the multiplexing between the RGP's and AT's read and write lines.

The REMREQ (Remote Request, active low) signal so that the host bus interface sequencer can indicate to the bus state machine that a remote access cycle is needed. REMACK (Remote Acknowledge, active low) indicates when the bus state machine is executing a remote access cycle. The host bus interface sequencer uses this acknowledge signal to determine when the host data is to be transferred to/from RGP memory.

The sequencer equations in *Figure 5* were compiled from the sequencer flowchart in *Figure 4*. *Figure 6* shows the equations after minimization. The states were renumbered so that the equations minimized into sets of product terms that fit into the GAL20V8. Each output has eight available product terms. ST1 requires ten product terms, so the result of three of them was fed back from another output. The setup time is critical on some of the inputs, so product terms dependent only on present state and RESET were fed back.

Figure 7 shows how the memory control signals are generated from the state variables; the minimized equations are shown in *Figure 8*. This logic is implemented in U201 and U202.

3.6 Address Decoder/Memory Control

The frame buffer is organized as four memory planes, as shown in *Figure 9*. Each pixel is represented by one bit in each plane. Each word in the array contains one bit of sixteen pixels and is mapped to a unique address in the global address space. Multiple memory planes can be accessed at once during drawing cycles, so a separate drawing address space is used. This space ranges from 000000 to 00FFFF; each location corresponds to four words, one in each plane. In cases where one pixel is accessed such as line drawing, B(3:0) specify which of the sixteen pixels is selected. When INTPL = 0, the drawing source read comes from within each plane, as shown in *Figure 10*. *Figure 11* shows the data path for a source read when the interplane bit is set. In this case, data is read from one plane into the four BPUs through the global data bus. The interplane BITBLT allows images to be copied from one plane to another and is convenient for rendering characters in all planes from fonts stored in a single plane. The data path for a destination write is shown in *Figure 12*.

Some of the memory control signals are held in U16. This register is set before a drawing operation is executed. The write mask, WMASK (3:0), is used to enable/disable planes for drawing operations. The interplane bit, INTPL, is used to select the BITBLT data source. Three of the bits in the register are used as flags. The flags are connected to J7; FLAG0 drives an LED.

The BPU control register enable, BPU function register select enables, and device selects for the write mask, UART and PROMs are generated by U204. PDLE and POE are used to access the pixel port and are generated by U206.

The column address strobes, CAS(3:0), are generated by U203. The control lines BE(3:0) and DIR(3:0) for the buffers between the main data bus and the data busses on each plane (U102, U103, U112, U113, U122, U123, U132 and U133) are generated by U205 and U206. The row and column addresses are multiplexed by U12 and U13. U6 controls the multiplexer by delaying RAS a half clock cycle.

The BPUs are run one clock cycle behind the RGP so that some of the status signals generated by the RGP are available earlier. The BPU control signals are delayed by U4. The details of the BPU operation are discussed in the DP8511 data sheet.

Software written for the DP850EB Evaluation Board can run on this board if the video parameters are altered; the old memory map is a subset of the new one.

RGP Memory Map:

000000–0007FF:	Monitor PROM (2k Words, Word = 16 Bits)
400000–40FFFF:	Bitplane 0 (64k Words)
420000–42FFFF:	Bitplane 1 (64k Words)
440000–44FFFF:	Bitplane 2 (64k Words)
460000–46FFFF:	Bitplane 3 (64k Words)
800001:	BPU Control Register Enable ($\overline{\text{CRE}}$)
800002:	BPU Pixel Data Latch Enable (PDLE)
800004:	BPU Pixel Output Enable ($\overline{\text{POE}}$)
A00001:	BPU Function Select Register Enable (FSE0 Bitplane 0)
A00002:	BPU Function Select Register Enable (FSE1 Bitplane 1)
A00004:	BPU Function Select Register Enable (FSE2 Bitplane 2)
A00008:	BPU Function Select Register Enable (FSE3 Bitplane 3)
C00000–C00007:	UART. The UART should not be accessed by the AT.
C00000:	RGP Maskable Interrupt ($\overline{\text{INT}}$) Reset. The RGP can reset its $\overline{\text{INT}}$ line to inactive, after the host has asserted it, by writing any data. Write only.
E00007:	Video Plane Write Mask Register. The bits in the mask register are the same as before. An LED is driven by FLAG0; it is on when the flag is set to zero. Write only.
F00000:	AT Interrupt Register. The RGP can interrupt the AT by writing with D0 = "1". The interrupt line asserted is determined by the jumper arrangement on the board. Writing with D0 = "0" resets the AT interrupt. The AT can also remove the interrupt. Write only.

3.7 Video Generation

The video timing is generated by the RGP. The rising edge of LCKL increments the software programmable counters in the RGP to generate the video shift register loads, BLANK, VSYNC and HSYNC. These signals can be adjusted in software to the resolution of LCKL (800 ns).

The display refresh memory cycles transfer data into the video shift registers inside the VDRAMs. The VDRAM shift registers hold 1024 pixels in this system, 256 bits per device. The video data is shifted every 800 ns during active video, 16 pixels per shift. The clock is derived by masking LCKL with $\overline{\text{BLANKD}}$. Data from the shift register outputs of the VDRAMs is multiplexed by using the output enables SE(3:0) into the shift registers U100, U110, U120 and U130. These four shift registers can be 25 ns GALs or 74LS195As. The data is blanked in U211 during horizontal and vertical blanking; the data and synchronization signals are latched and buffered by U18.

U207 and U5 are used to control the timing of the video shift register loads. These memory cycles can occur during blanking of the video data or while data is being shifted out to the monitor. During blanking, the timing of the load is not critical because the data is not being shifted. In this case, the MIDSCAN signal is kept low and the sequencer is not held. When data must be loaded in the middle of a scan line, the window in which the load must be done is much smaller. It must be loaded after the last word of the previous scan line is shifted (more than 25 ns after the rising edge of GSCL) but at least 20 ns before the next shift clock edge. The arbiter/timing generator enters the display refresh cycle and waits at state 18 until the MIDSCAN signal is low. The load occurs on the rising edge of $\overline{D}TO\overline{E}$ in the first half of state 0 after state 21. Early entry into the cycle ensures that the load can be made before the end of the window. The MIDSCAN signal holds the sequencer so that the load is not too early.

The mid-scan line load feature allows contiguous mapping of the screen buffer as shown in *Figure 9*, resulting in more efficient use of screen buffer memory. A screen size of 480 x 640 uses 307,200 pixels, 29.3% of the 1,048,576 pixels available in the VDRAM. Two buffers of this size can be used with 27,136 words of free memory remaining in each of the four planes for programs, data, character fonts and graphics images. In a system without mid-scan line load, using 640 pixels per horizontal scan line, 384 pixels (37.5%) of each shift register load would be unused. This scheme would also consume more memory bandwidth for screen refresh because only 62.5% of the pixels in each load are used, requiring the shift register to be loaded more frequently.

Figure 13 shows a code sequence for initializing the video parameters; these parameters are different than those of the DP850EB board. The blanking signal must be generated two counts earlier in this design for the mid-scan line shift register load circuitry.

3.8 PROM

The board has two PROMs, U14 and U15, which reside at locations 000000H–0007FFH. CK2 is used to clock their internal registers. They hold the monitor which is executed by the RGP upon reset of the board. The monitor communicates with the host-resident RGP debugger through the communication area in VDRAM when using the parallel interface, or through the UART when using the serial interface. The monitor also uses a small area in VDRAM for variable storage, which must remain undisturbed if the moni-

tor is to be reentered. To run programs on the RGP, the host loads the code into VDRAM through the parallel interface, then issues a command to the monitor to branch to it. The details are discussed in the debugger/monitor documentation.

3.9 UART

The UART (U26) is used for serial communication with the monitor program. U28 generates the clock for the UART. The line transceiver is U27. W5 allows pins 2 and 3 of the RS-232 connector to be switched. W6 allows pins 6 and 20 to be switched. A serial connection is optional; the monitor can be accessed through the AT interface by using the communication area in memory.

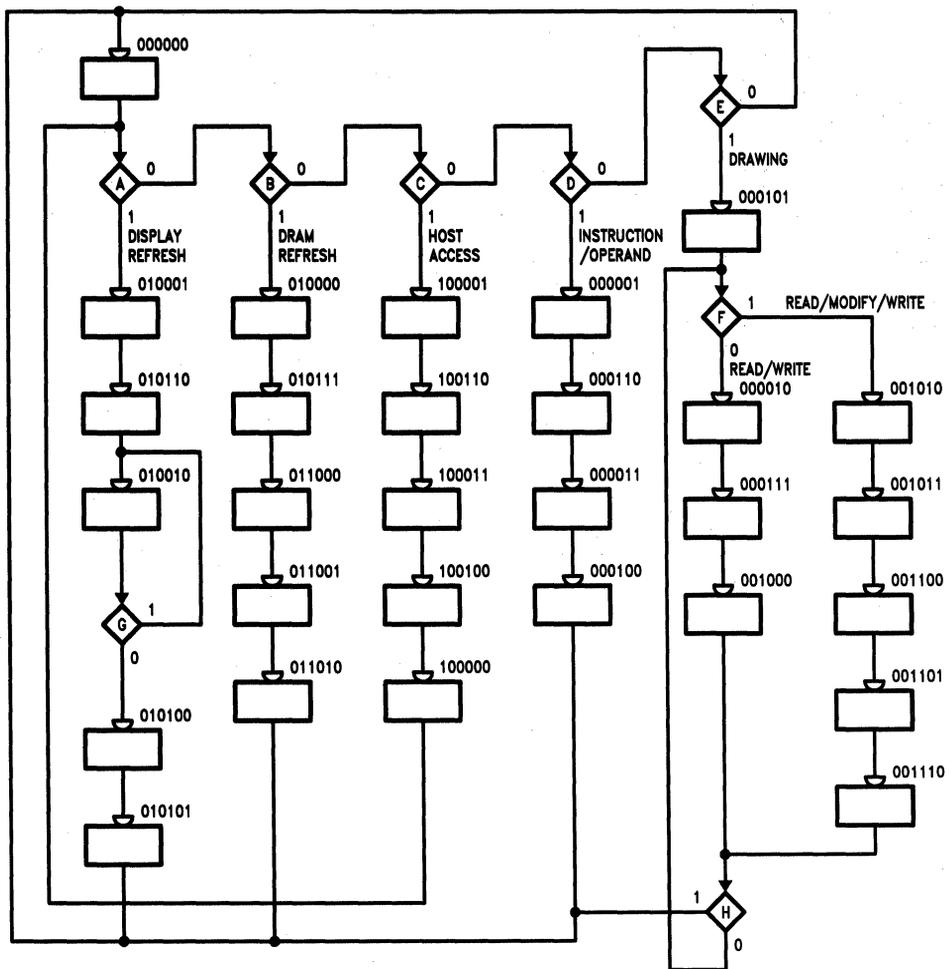
4.0 OTHER DESIGN ISSUES

The pixel clock frequency could be increased to display higher screen resolutions. Higher resolutions may require more memory; the set of VDRAMs on the board will hold a single buffer of 1k x 1k. The RGP can directly support screen buffer sizes of up to 16k x 16k, though schemes can be used to access larger screen buffers.

Memory planes can be added to increase the number of bits per pixel. The number of planes that can be added is not limited by the RGP; the rendering algorithms execute at the same speed, independent of the number of planes. A palette/DAC could be added to generate analog video signals and allow selection of various color sets.

If a larger PLD was used for the main sequencer in U200, then ALE could be latched inside it instead of U5. This scheme would give more set-up time for ALE, allowing a slower PLD to be used. The larger PLD also could accommodate additional features such as allowing the AT to initiate drawing cycles so that single pixel values could be written and read.

In the AT interface circuitry, the data write latches and lower address latches are used as buffers. If the circuitry was altered, they could be used as latches to pipeline writes from the host. This would allow the host to deposit data and the address of the data to be written without being held during the entire write cycle. The host could then be obtaining the next word to be written while the previous word is propagating into the RGP's memory. The AT would be held if it attempted to write a second word before transfer of the first word was completed. This technique would result in a slightly more complicated host bus interface machine, but would reduce the host bus bandwidth spent writing to the board.



TL/F/10429-4

- A $ALE1 * BS1 * BS0$
- B RRQ
- C \overline{REMREQ}
- D $ALE1 * \overline{BS1}$
- E $ALE1$
- F $\overline{MUXW} * \overline{MUXR}$
- G $MIDSCAN$
- H $RRQ + \overline{REMREQ} + \overline{PB} + \overline{BS1} + BS0 + \overline{ALE1}$

FIGURE 4. Flowchart for the Arbiter/Timing Generator State Machine

INPUTS		OUTPUTS	INPUTS		OUTPUTS
ABMMMMRR	PPPPPP	NNNNNN	101X00110	001000	001010
LSSIUUBER	RSSSSS	RSSSSS	101X01110	001000	000010
E01DXX~MQ	ETTTTT	ETTTTT	101X1X110	001000	000010
1 SRW R	M43210	M43210	XXXXXXXXXX	001010	001011
C~~ E	A	A	XXXXXXXXXX	001011	001100
A Q	C	C	XXXXXXXXXX	001100	001101
N ~	K	K	XXXXXXXXXX	001101	001110
			XXXXXXXXX1	001110	000000
			XXXXXXXXX0X	001110	000000
111XXXXXX	000000	010001	XXXXXXXX0XX	001110	000000
110XXXXX1	000000	010000	XXXXXXXX0XX	001110	000000
110XXXXX0	000000	100001	XX0XXXXXX	001110	000000
110XXXX10	000000	000001	X1XXXXXXX	001110	000000
10XXXXXX1	000000	010000	0XXXXXXX	001110	000000
10XXXXX00	000000	100001	101X00110	001110	001010
100XXXX10	000000	000001	101X01110	001110	000010
101XXXX10	000000	000101	101X1X110	001110	000010
0XXXXXX1	000000	010000	111XXXXXX	100000	010001
0XXXXXX00	000000	100001	110XXXXX1	100000	010000
0XXXXXX10	000000	000000	110XXXXX00	100000	100001
XXXXXXXXX	010001	010110	110XXXXX10	100000	000001
XXXXXXXXXX	010110	010010	10XXXXXX1	100000	010000
XXX1XXXXX	010010	010010	10XXXXX00	100000	100001
XXX0XXXXX	010010	010100	100XXXX10	100000	000001
XXXXXXXXXX	010100	010101	101XXXX10	100000	000101
XXXXXXXXXX	000010	000111	0XXXXXX1	100000	010000
XXXXXXXXXX	100110	100011	0XXXXXX00	100000	100001
XXXXXXXXXX	000110	000011	0XXXXXX10	100000	000000
XXXXXXXXXX	010000	010111			
XXXXXXXXXX	100001	100110			
XXXXXXXXXX	000001	000110			
XXXX00XXX	000101	001010			
XXXX01XXX	000101	000010			
XXXX1XXXX	000101	000010			
XXXXXXXXXX	010111	011000			
XXXXXXXXXX	011000	011001			
XXXXXXXXXX	011001	011010			
XXXXXXXXXX	010101	000000			
XXXXXXXXXX	011010	000000			
XXXXXXXXXX	100011	100100			
XXXXXXXXXX	100100	100000			
XXXXXXXXXX	000011	000100			
XXXXXXXXXX	000100	000000			
XXXXXXXXXX	000111	001000			
XXXXXXXXX1	001000	000000			
XXXXXXXX0X	001000	000000			
XXXXXX0XX	001000	000000			
XX0XXXXXX	001000	000000			
X1XXXXXXXX	001000	000000			
0XXXXXXXX	001000	000000			

TL/F/10429-8

TL/F/10429-5

FIGURE 5. Equations for the Arbiter/Timing Generator, before Minimization

INPUTS		OUTPUTS
ABBMMMPRR	PPPPPP	NNNNNN
LSSIUUBER	RSSSSS	RSSSSS
E01DXX~MQ	ETTTTT	ETTTTT
1 SRW R	M43210	M43210
C~ E A		A
A Q C		C
N ~ K		K
X0XXXXX00	X00000	100000
0XXXXXX00	X00000	100000
XXXXXXXXXX	1000X1	100000
XXXXXXXXXX	1001X0	100000
XX0XXXXX00	X00000	100000
XXXXXXXXXX	01X00X	010000
XXXXXXXXXX	01011X	010000
XXXXXXXXXX	010XX0	010000
XXXXXXXXXX1	X00000	010000
111XXXXXX	X00000	010000
XXXXXXXXXX	00110X	001000
XXXXXXXXXX	00101X	001000
XXXXXXXXXX	01100X	001000
XXXXXXXXXX	0X0111	001000
XXXXX00XXX	00X101	001000
101X00110	001XX0	001000
XXXXXXXXXX	010X00	000100
XXXXXXXXXX	01000X	000100
XXX0XXXXX	0X0010	000100
XXXXXXXXXX	00001X	000100
XXXXXXXXXX	X000X1	000100
XXXXXXXXXX	00X011	000100
XXXXXXXXXX	00110X	000100
101XXXX10	X0000X	000100
XXXXXXXXXX	01000X	000010
XXXXXXXXXX	01X001	000010
XXXXXXXXXX	0X0110	000010
XXX1XXXXX	0X0X10	000010
XXXXXXXXXX	X00001	000010
XXXXXXXXXX	00X010	000010
XXXXXXXXXX	X00110	000010
XXXXXXXXXX	00X101	000010
101XXX110	0010X0	000010
101XXX110	00XX10	000010
XXXXXXXXXX	01X000	000001
XXXXXXXXXX	010X00	000001
1XXXXXXXX0	X00000	000001
XXXXXXXXX00	X00000	000001
XXXXXXXXXX	001100	000001
XXXXXXXXXX	00X010	000001
XXXXXXXXXX	X00110	000001
111XXXXXX	X00000	000001

TL/F/10429-7

FIGURE 6. Equations for the Arbiter/Timing Generator, after Minimization

INPUTS		OUTPUTS				INPUTS		OUTPUTS				
RWRW	RSSSS	UWDDD	DDRR	RRIRC	SS	XXXX	X10011	00110	0000	01111	11	
1111	ETTTTT	ARTLO	DBDS	SFOAA	TT	XXXX	X10010	00001	0010	01111	11	
~~DD	M43210	R~OEE	IEYT	TISS	86	XXXX	X10001	11001	0010	01110	11	
~~	A~~~~	T E~~	N~	A	RPNII	3						
C		W ~		L R	NN	4	XXXX	X01110	00100	0011	00100	11
K		R		E Q	~~		XXXX	X01001	00100	0010	00111	11
~		~		~	~		XXXX	X01101	00100	0010	00111	11
							XXXX	X01010	00000	0010	00110	11
XXXX	X11111	00000	0010	01100	11							
XXXX	111110	00000	0011	01100	10	XXXX	X01111	00000	0010	10101	11	
1XXX	111001	00000	0110	01011	10	XXXX	X01000	00000	0010	00101	11	
0XXX	111001	00000	1110	01011	10	XXXX	X00111	00000	0010	00111	11	
1XXX	111100	00000	0100	01011	10	XXXX	X00110	00000	0010	00110	11	
0XXX	111100	00100	1100	01011	10	XXXX	X00101	00000	0010	00100	11	
0XXX	111011	00100	1110	01010	11	XXXX	X10110	XXXXX	XXXX	XXXXX	XX	
10XX	111011	11000	0110	01010	11	XXXX	X10000	XXXXX	XXXX	XXXXX	XX	
11XX	111011	00000	0110	01010	11	XXXX	X01100	XXXXX	XXXX	XXXXX	XX	
XXXX	011110	00000	X010	01100	10	XXXX	X00100	XXXXX	XXXX	XXXXX	XX	
XXXX	011001	00000	X010	01011	10	XXXX	X00011	XXXXX	XXXX	XXXXX	XX	
1XXX	011100	00000	X010	01011	10	XXXX	X00010	XXXXX	XXXX	XXXXX	XX	
0XXX	011100	00100	X010	01011	10	XXXX	X00001	XXXXX	XXXX	XXXXX	XX	
0XXX	011011	00100	X010	01010	11	XXXX	X00000	XXXXX	XXXX	XXXXX	XX	
10XX	011011	11000	X010	01010	11							
11XX	011011	00000	X010	01010	11							
XXXX	X11010	00000	0010	01100	11							
X1XX	X11101	00000	0001	01111	11							
X0XX	X11101	00001	0001	01111	11							
0XXX	X11000	00100	0010	01111	01							
1XXX	X11000	00001	0010	01111	01							
XX00	X10111	11001	0010	01110	11							
XX01	X10111	00110	0010	01110	11							
XX10	X10111	11001	0010	01110	11							
XX11	X10111	00100	0010	01110	11							
XXXX	X10101	00000	0011	01111	11							
XXXX	X10100	00100	0010	01111	11							

TL/F/10429-10

TL/F/10429-9

FIGURE 7. Equations for Memory Control Signal Generation, before Minimization

INPUTS						OUTPUTS						INPUTS						OUTPUTS											
RWRW	RSSSSS	UWDDD	DDRR	RRIRC	SS	XXXX	X011X0	00000	0001	00000	00	XXXX	X011X0	00000	0001	00000	00	XXXX	X01111	00000	0000	10000	00	XXXX	X1XXXX	00000	0000	01000	00
1111	ETTTT	ARTLO	DBDS	SFOAA	TT	XXXX	X0XXXX	00000	0000	00100	00	XXXX	XX0XXX	00000	0000	00100	00	XXXX	XX0XXX	00000	0000	00100	00	XXXX	XX0XXX	00000	0000	00100	00
~DD	M43210	R~OEE	IEYT	TIESS	86	XXXX	XXX0X0	00000	0000	00100	00	XXXX	XXX0X0	00000	0000	00100	00	XXXX	XXX1X0	00000	0000	00100	00	XXXX	XXX1X1	00000	0000	00100	00
~A	~	T E	~ N	~ A	RPNII 3	XXXX	XX001X	00000	0000	00010	00	XXXX	XXX0X1	00000	0000	00010	00	XXXX	X1XX0X	00000	0000	00010	00	XXXX	XX1X01	00000	0000	00010	00
C	~	W	~	L R	NN 4	XXXX	XX0X1X	00000	0000	00010	00	XXXX	XX0X1X	00000	0000	00010	00	XXXX	XX1X0X	00000	0000	00001	00	XXXX	X1X10X	00000	0000	00001	00
K	~	R	~	E Q	~	XXXX	XX001X	00000	0000	00001	00	XXXX	XX001X	00000	0000	00001	00	XXXX	X0XX11	00000	0000	00001	00	XXXX	X0XX11	00000	0000	00001	00
~	~	~	~	~	~	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00
XXXX	X1011X	10000	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX000X	10000	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
10XX	X11011	10000	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X1011X	01000	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX000X	01000	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
10XX	X11011	01000	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XXX011	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX0011	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X011X0	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X0110X	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXX1	X10X11	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
0XXX	X1XX00	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX0X00	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X0X0X1	00100	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX0011	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XX01	X10X11	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
X0XX	X11101	00001	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X1011X	00001	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX00X0	00001	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX000X	00001	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
1XXX	X1X000	00001	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
0XXX	X110X1	00000	1000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
0XXX	XX1100	00000	1000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	1110X1	00000	0100	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	1X1100	00000	0100	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XX0X0X	00000	0010	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X0XX0X	00000	0010	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XXX11X	00000	0010	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	XXXX10	00000	0010	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	OXXXX0	00000	0010	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	X110XX	00000	0010	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						
XXXX	1X1110	00000	0001	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00	XXXX	XX001X	00010	0000	00000	00						

TL/F/10429-12

TL/F/10429-11

FIGURE 8. Equations for Memory Control Signal Generation, after Minimization

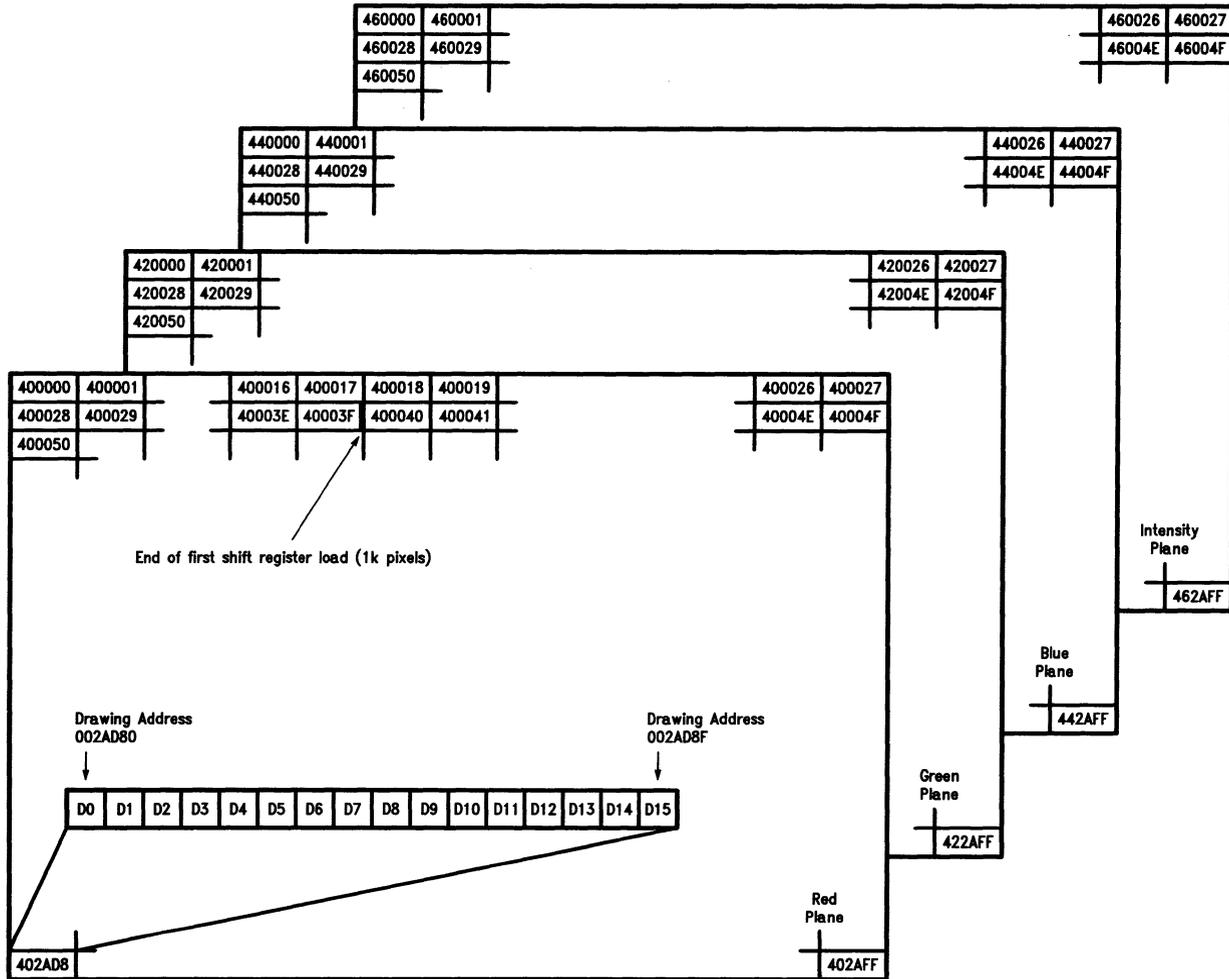


FIGURE 9. Frame Buffer Organization

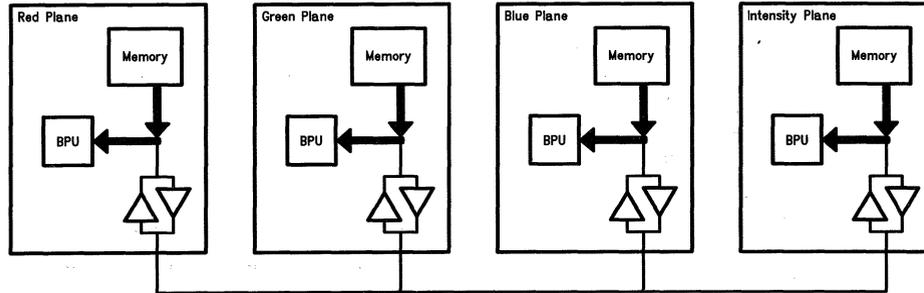


FIGURE 10. Local Drawing Source Read

TL/F/10429-14

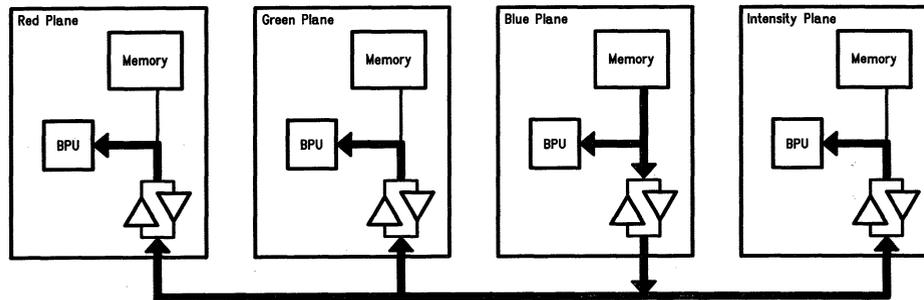


FIGURE 11. Interplane Drawing Source Read

TL/F/10429-15

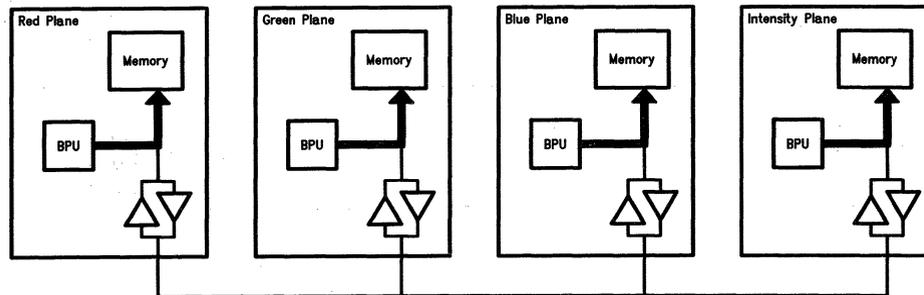


FIGURE 12. Local Drawing Destination Write

TL/F/10429-16

```
hslt_val:      .equ 47
hse_val:      .equ 2
hbe_val:      .equ 2
hbs_val:      .equ 42
vft_val:      .equ 492
vse_val:      .equ 2
vbe_val:      .equ 11
vbs_val:      .equ 491
screen_warp:  .equ 40
turnonvideo:  mov  #stop_vcr, vcr
              mov  #hslt_val, video
              mov  #hse_val, video
              mov  #hbe_val, video
              mov  #hbs_val, video
              mov  #vft_val, video
              mov  #vse_val, video
              mov  #vbe_val, video
              mov  #vbs_val, video
              mov  #screen_base, dbb
              mov  #screen_warp, dbwrp
              mov  #run_vcr, vcr
              ret
```

FIGURE 13. Instructions for Setting Video Parameters

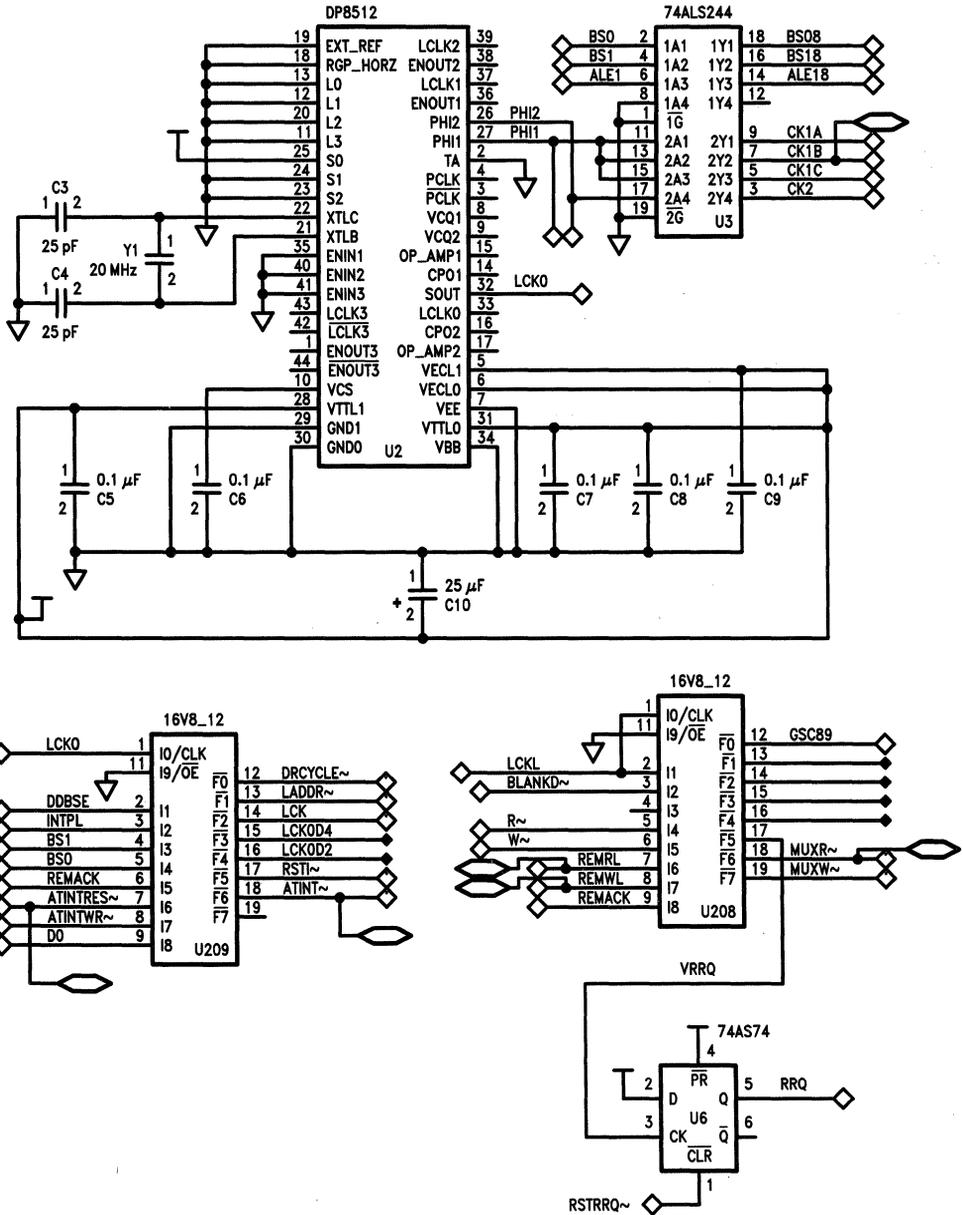


FIGURE 14a

TL/F/10429-17

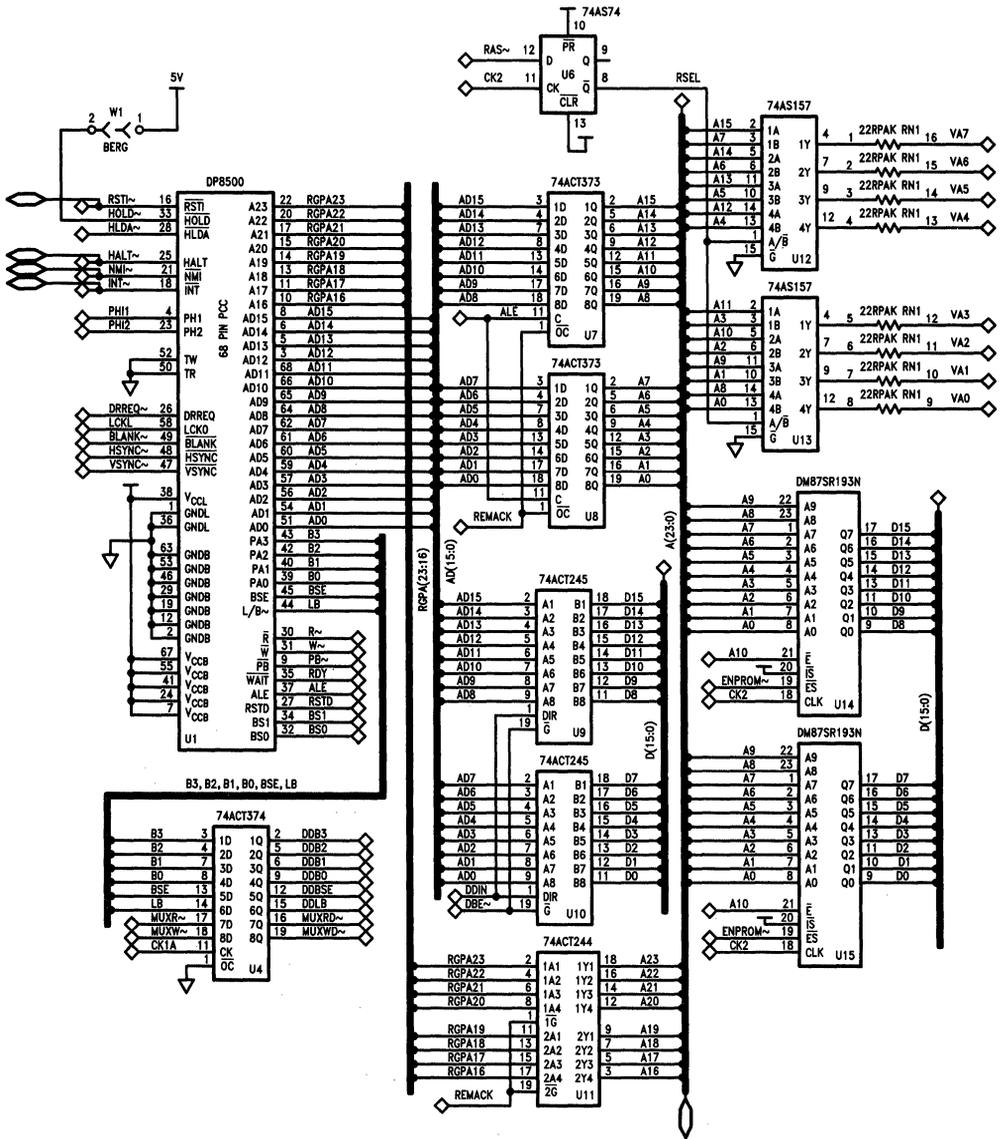


FIGURE 14b

TL/F/10429-18

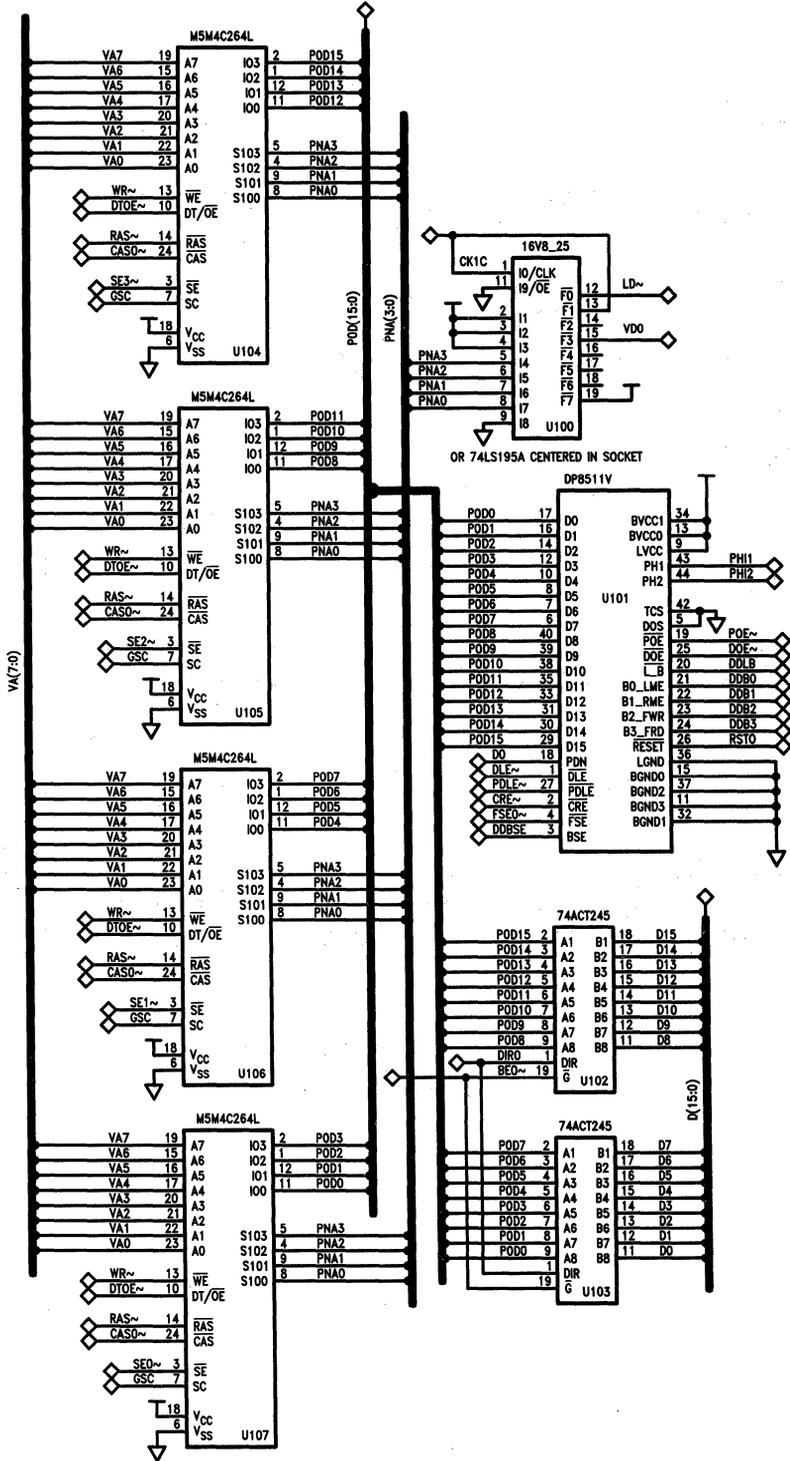


FIGURE 14c

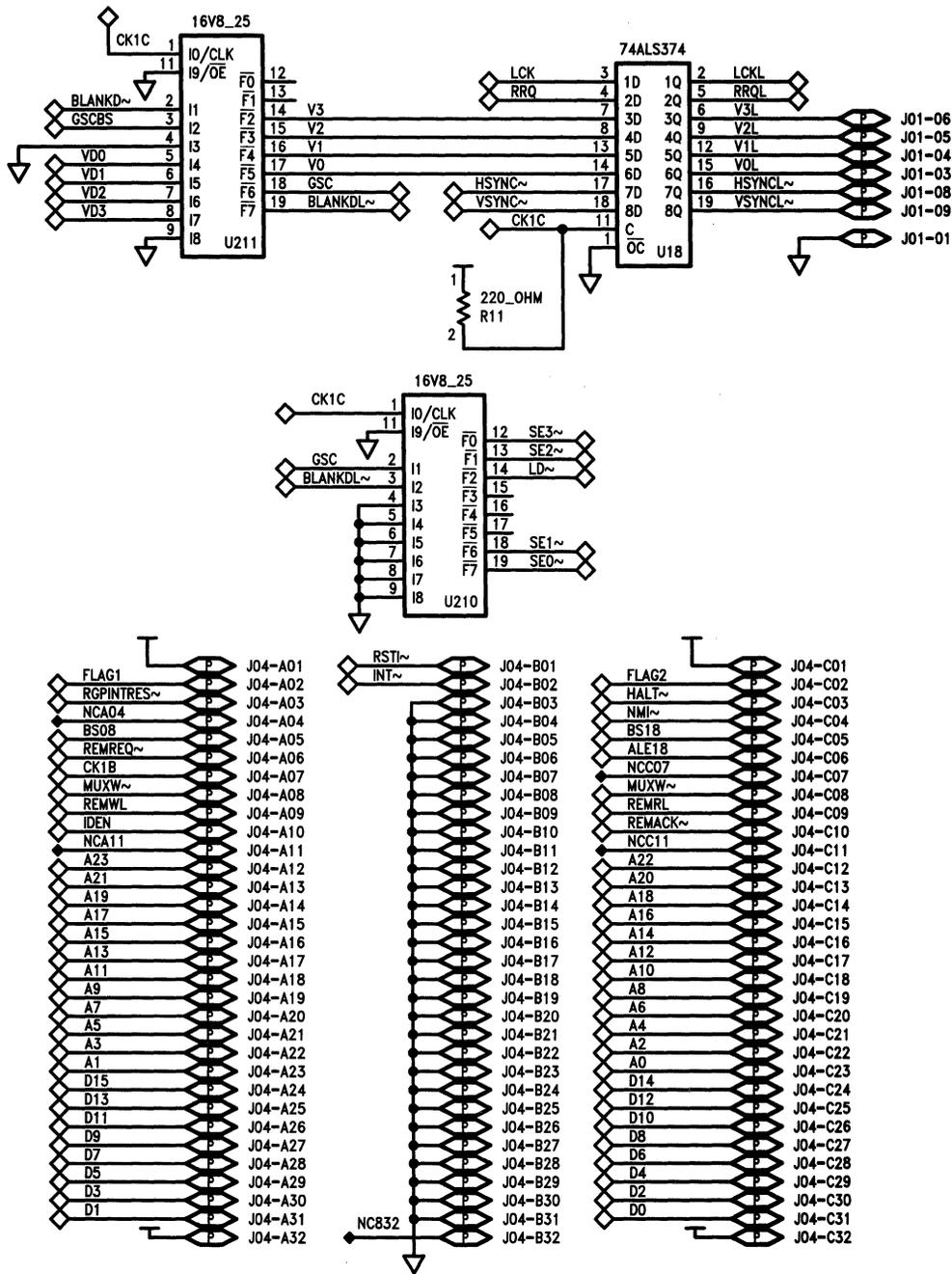


FIGURE 14d

TL/F/10429-20

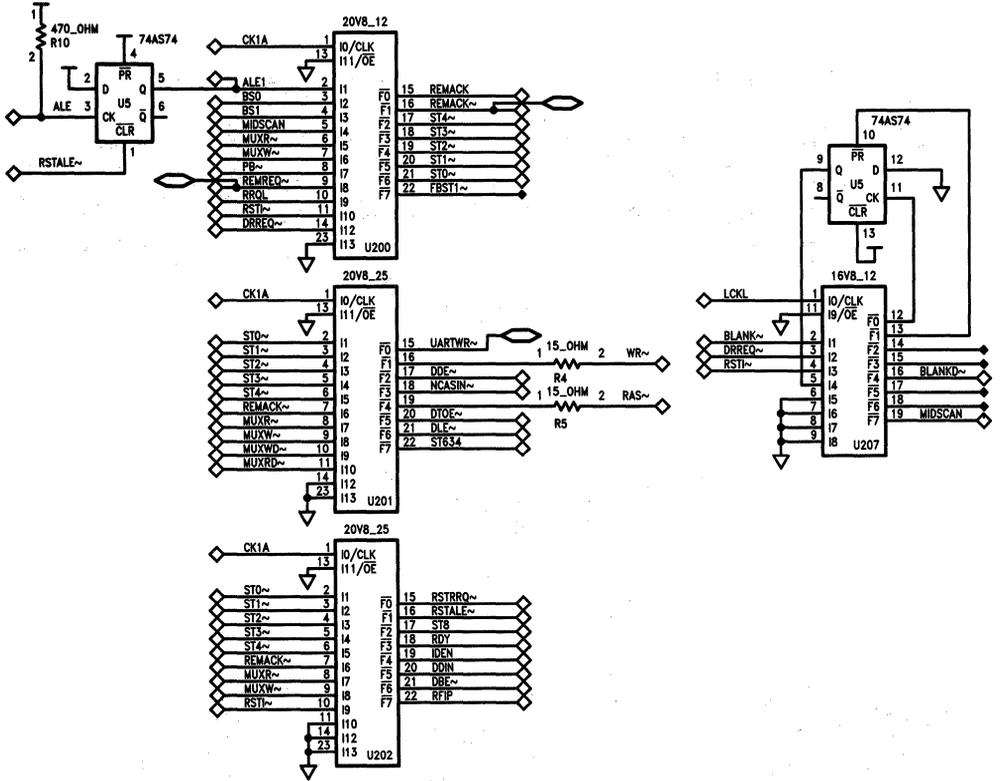


FIGURE 14e

TL/F/10429-21

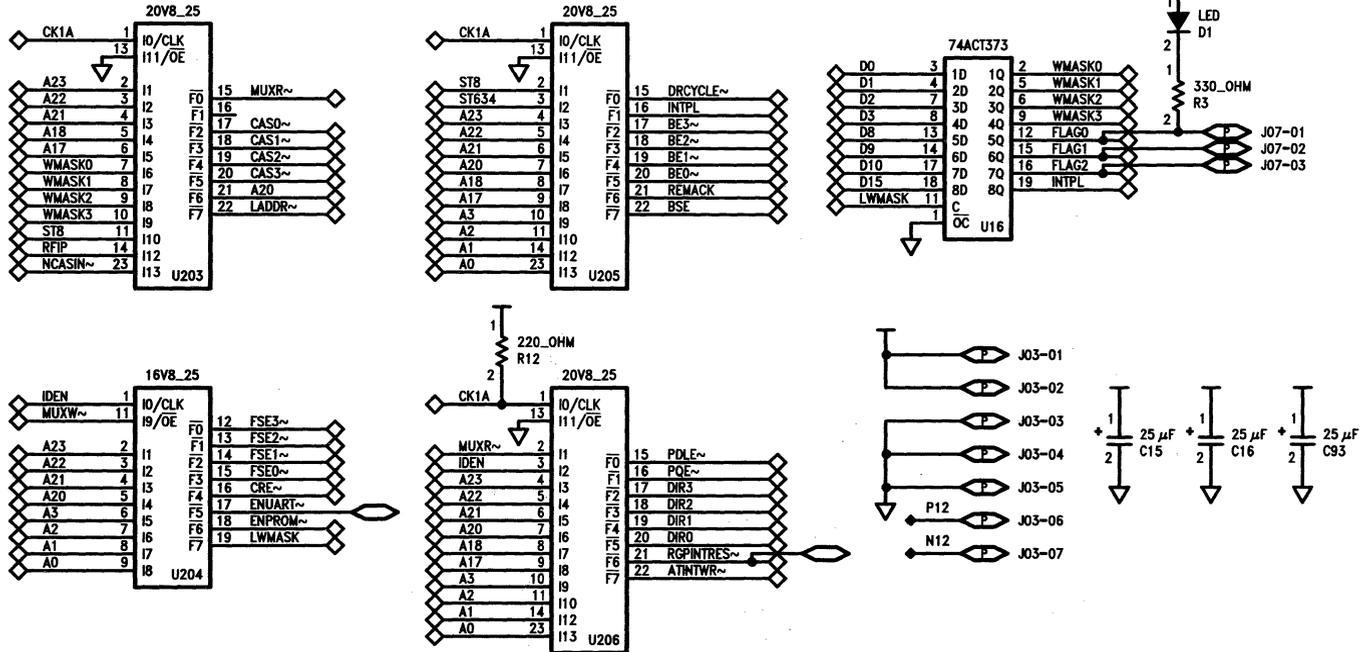


FIGURE 14f

TL/F/10429-22

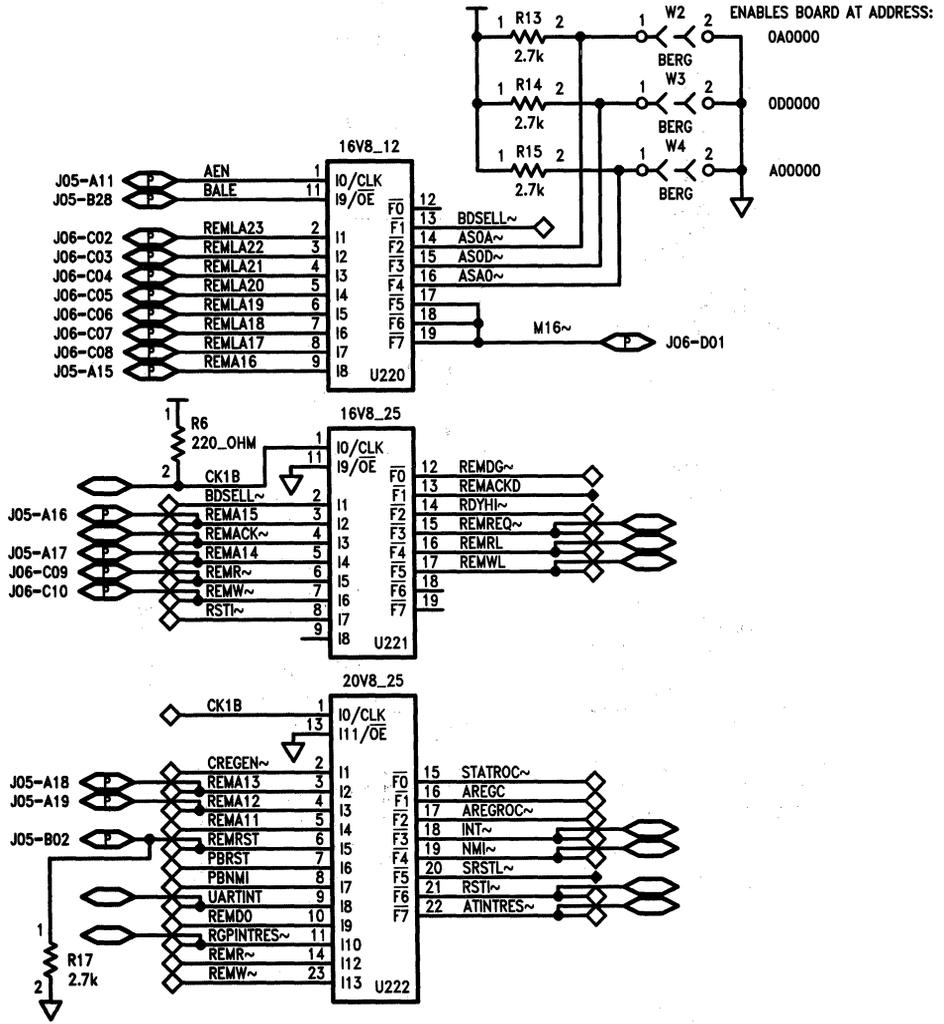
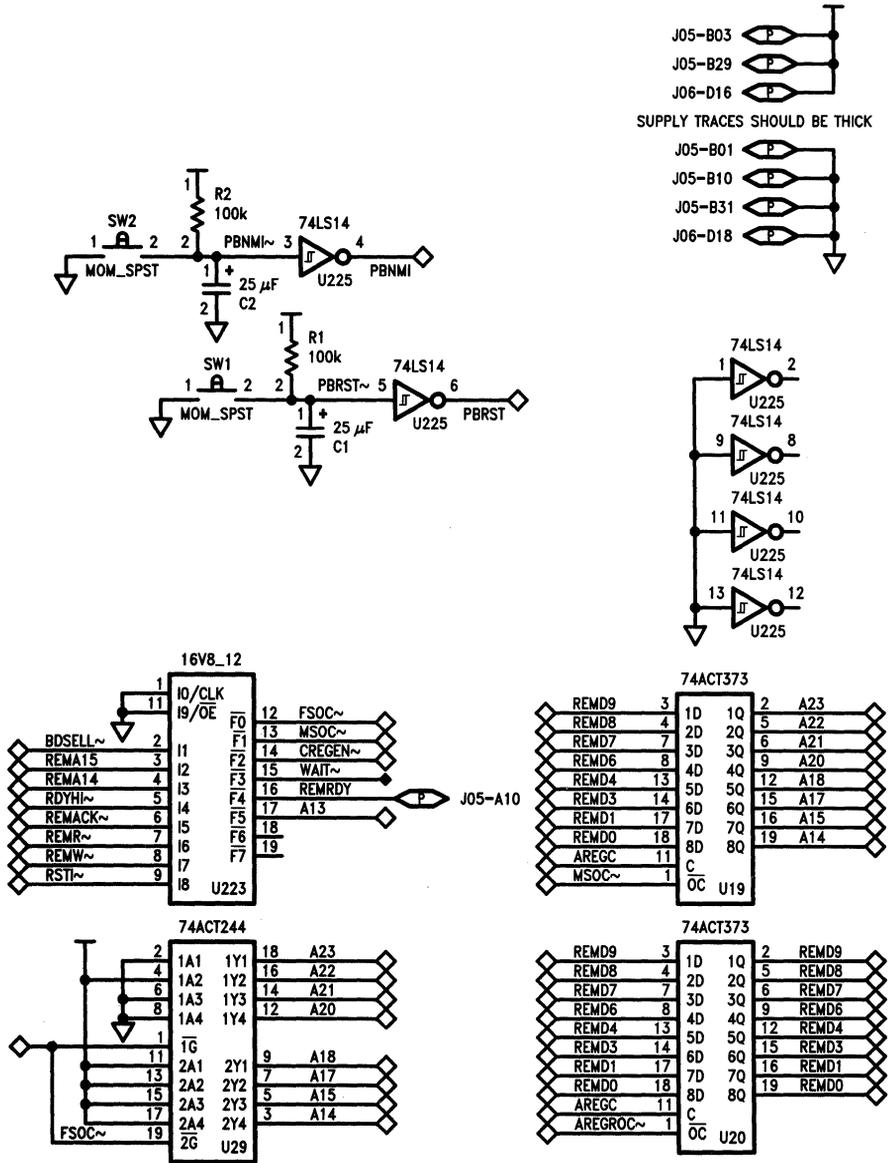


FIGURE 14g

TL/F/10429-45



SUPPLY TRACES SHOULD BE THICK

FIGURE 14h

TL/F/10429-46

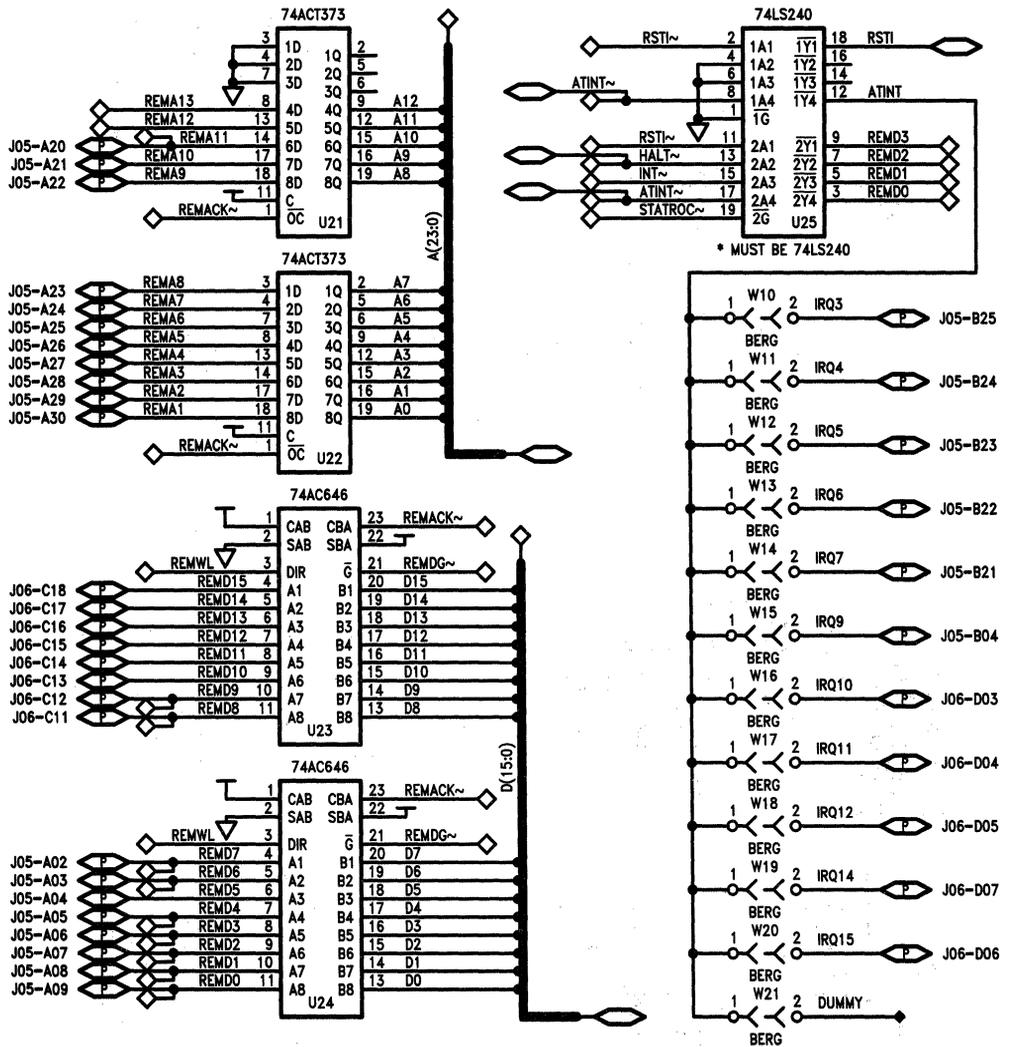


FIGURE 141

TL/F/10429-47

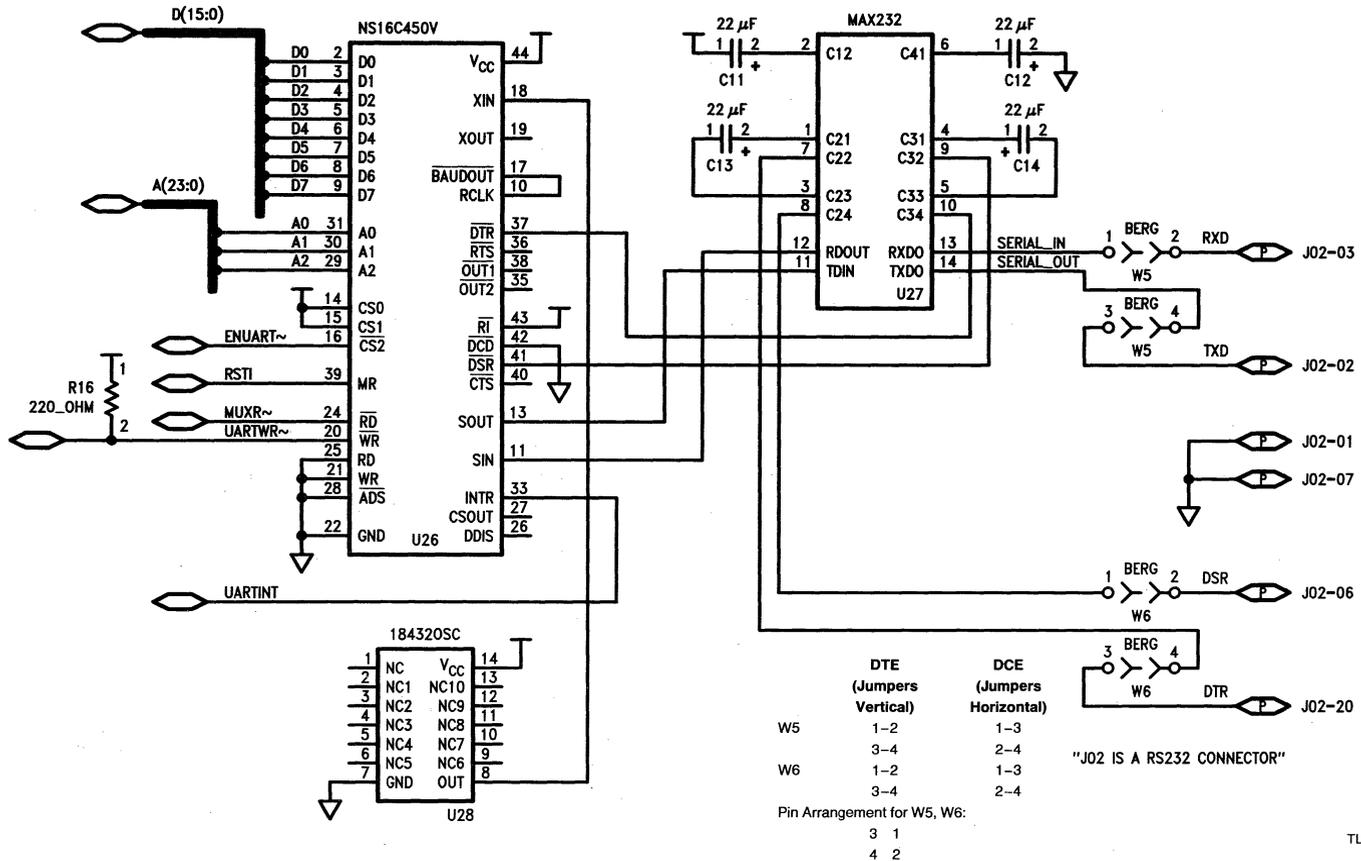


FIGURE 14j

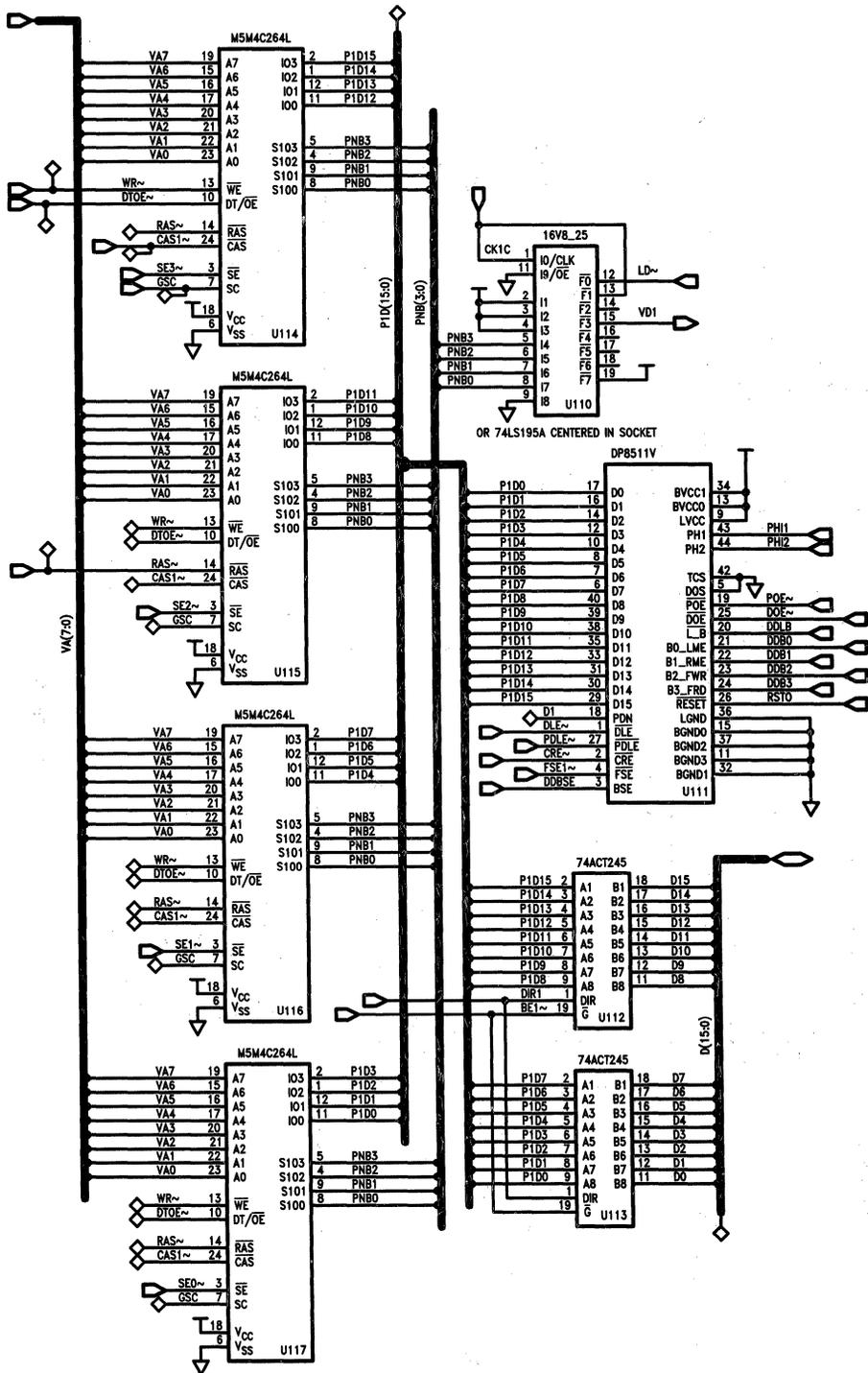


FIGURE 14k

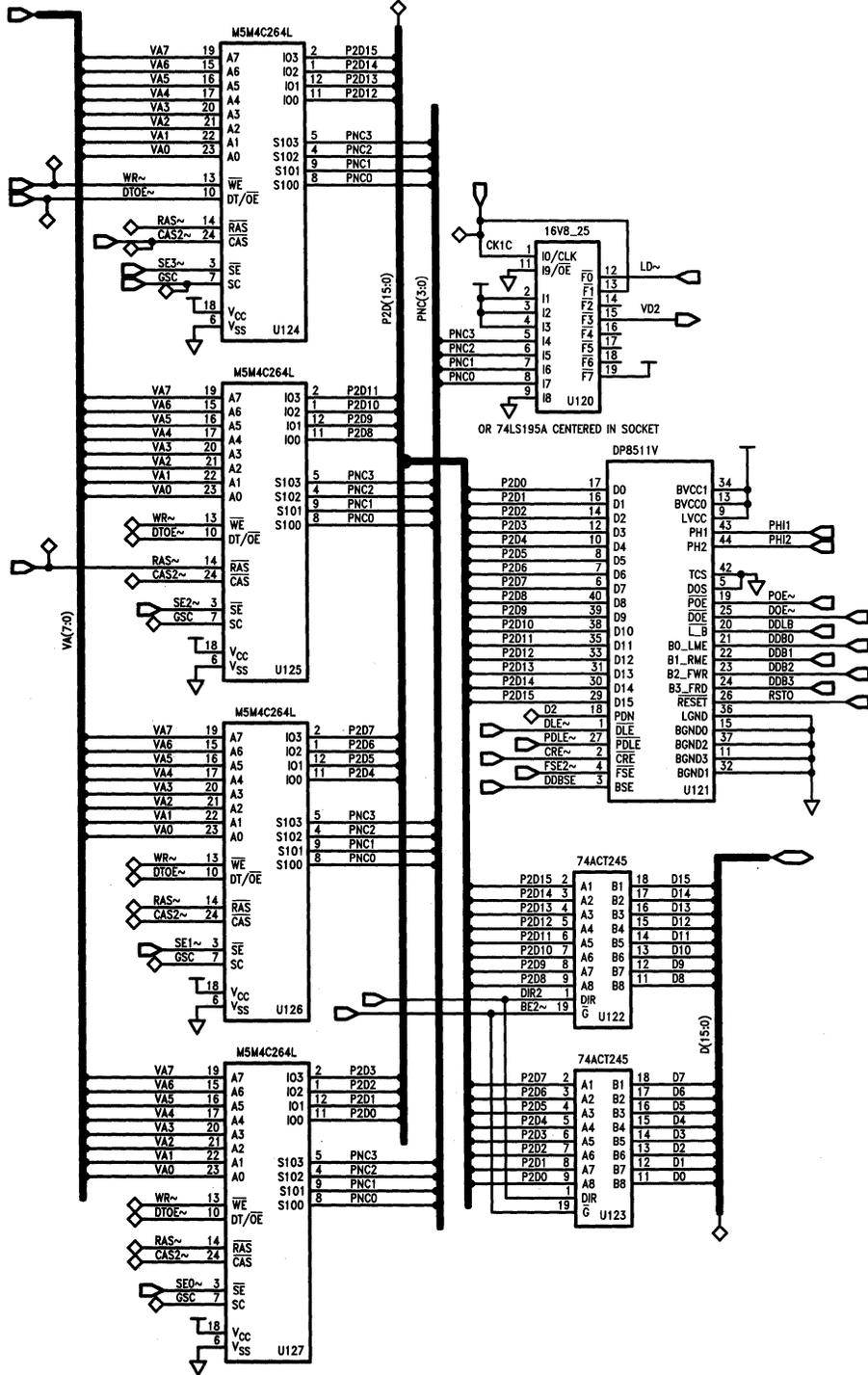


FIGURE 141

TL/F/10429-23

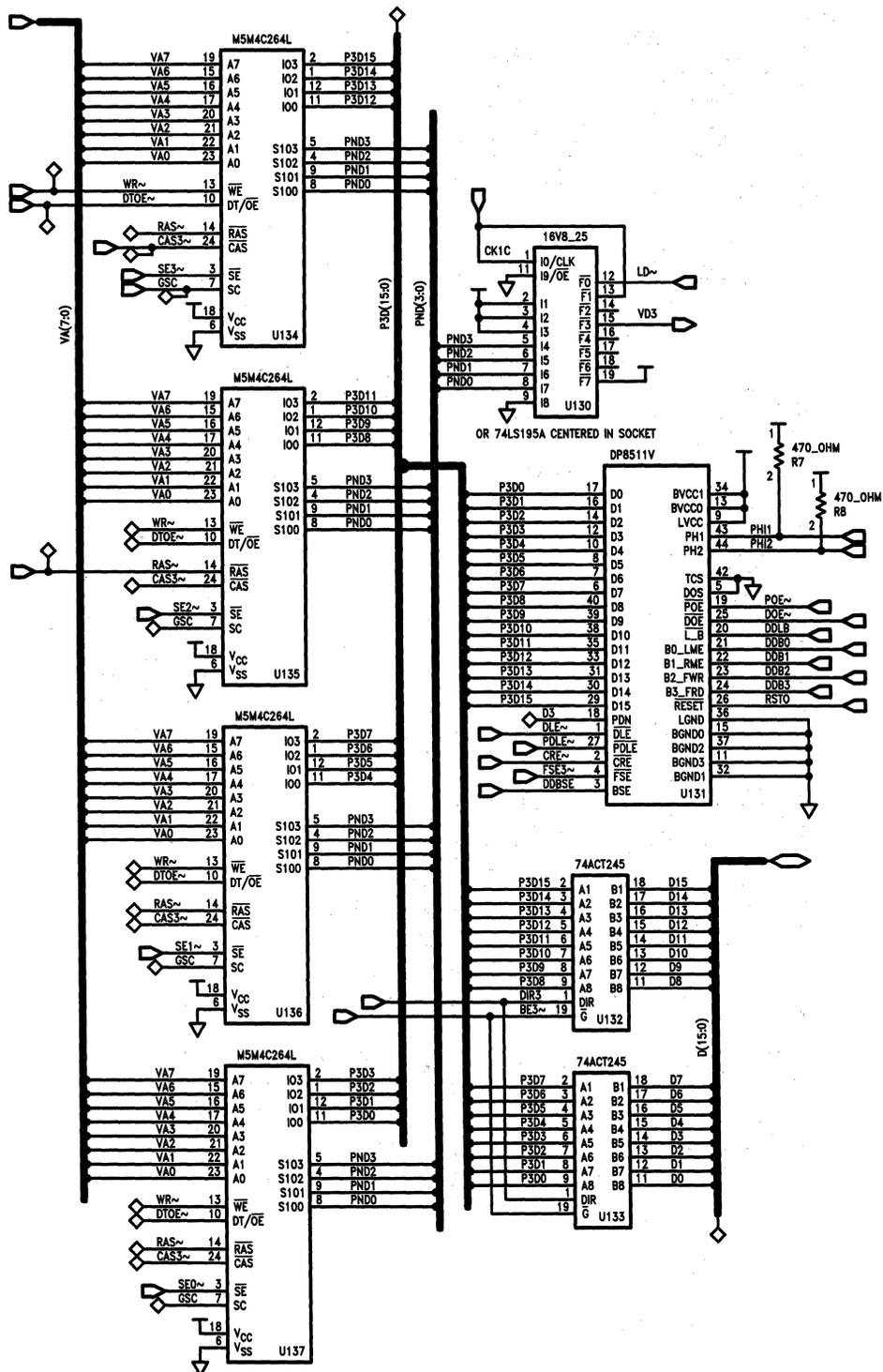


FIGURE 14m

```

MODULE U100R
TITLE
'Video Shift Register. U110, U120, U130 are same. 74LS195A
equiv. Device:U100  Filename:U100R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U100RJ device 'p16v8r';
```

```

CK1C      pin    1;
NC2       pin    2;
NC3       pin    3;
NC4       pin    4;
PNA3      pin    5;
PNA2      pin    6;
PNA1      pin    7;
PNA0      pin    8;
NC9       pin    9;
GND       pin   10;
OE~       pin   11;
LD~       pin   12;
NC13      pin   13;
O5        pin   14;
VD0       pin   15;
O4        pin   16;
O3        pin   17;
NC18      pin   18;
NC19      pin   19;
VCC       pin   20;

```

Equations

```

!VD0 := !LD~ & !PNA0 # LD~ & !O5;
!O5  := !LD~ & !PNA1 # LD~ & !O4;
!O4  := !LD~ & !PNA2 # LD~ & !O3;
!O3  := !LD~ & !PNA3;

```

```
End;
```

FIGURE 15. Equations for U100

TL/F/10429-25

MODULE U200R
 TITLE
 'Main Bus Sequencer (from B5)
 Device:U200 Filename:U200R
 Jim Margeson
 Date: 12/15/88
 Revision: 2.0'

U200RJ device 'p20v8r';

CK1A	pin	1;
ALE1	pin	2;
BS0	pin	3;
BS1	pin	4;
MIDSCAN	pin	5;
MUXR~	pin	6;
MUXW~	pin	7;
PB~	pin	8;
REMREQ~	pin	9;
RRQ	pin	10;
RSTI~	pin	11;
GND	pin	12;
OE~	pin	13;
DRREQ~	pin	14;
REMACK	pin	15;
REMACK~	pin	16;
ST4~	pin	17;
ST3~	pin	18;
ST2~	pin	19;
ST1~	pin	20;
ST0~	pin	21;
FBST1~	pin	22;
NC23	pin	23;
VCC	pin	24;

Equations

```
REMACK := RSTI~&!BS0&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&!ALE1&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&!REMACK~&ST4~&ST3~&ST2~&!ST0~
# RSTI~&!REMACK~&ST4~&ST3~&!ST2~&ST0~
# RSTI~&!BS1&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~;
```

```
!REMACK~ := RSTI~&!BS0&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&!ALE1&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&!REMACK~&ST4~&ST3~&ST2~&!ST0~
# RSTI~&!REMACK~&ST4~&ST3~&!ST2~&ST0~
# RSTI~&!BS1&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~;
```

```
!ST4~ := RSTI~&REMACK~&!ST4~&ST2~&ST1~
# RSTI~&REMACK~&!ST4~&ST3~&!ST2~&!ST1~
# RSTI~&REMACK~&!ST4~&ST3~&ST0~
# RSTI~&RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&ALE1&BS0&BS1&ST4~&ST3~&ST2~&ST1~&ST0~;
```

```
!ST3~ := RSTI~&REMACK~&ST4~&!ST3~&!ST2~&ST1~
```

FIGURE 16. Equations for U200

```

# RSTI~&REMACK~&ST4~&!ST3~&ST2~&!ST1~
# RSTI~&REMACK~&!ST4~&!ST3~&ST2~&ST1~
# RSTI~&REMACK~&ST3~&!ST2~&!ST1~&!ST0~
# RSTI~&!MUXR~&!MUXW~&REMACK~&ST4~&!ST2~&ST1~&!ST0~
# RSTI~&ALE1&!BS0&BS1&!MUXR~&!MUXW~&PB~&REMREQ~&!RRQ&REMACK~
&ST4~&!ST3~&ST0~;

!ST2~ := RSTI~&REMACK~&!ST4~&ST3~&ST1~&ST0~
# RSTI~&REMACK~&!ST4~&ST3~&ST2~&ST1~
# RSTI~&!MIDSCAN&REMACK~&ST3~&ST2~&!ST1~&ST0~
# RSTI~&REMACK~&ST4~&ST3~&ST2~&!ST1~
# RSTI~&ST4~&ST3~&ST2~&!ST0~
# RSTI~&REMACK~&ST4~&ST2~&!ST1~&!ST0~
# RSTI~&REMACK~&ST4~&!ST3~&!ST2~&ST1~
# RSTI~&ALE1&!BS0&BS1&REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~;

!ST1~ := RSTI~&ALE1&!BS0&BS1&PB~&REMREQ~&!RRQ&REMACK~
&ST4~&!ST3~&ST2~&ST0~
# RSTI~&ALE1&!BS0&BS1&PB~&REMREQ~&!RRQ&REMACK~&ST4~&!ST1~&ST0~
# RSTI~&MIDSCAN&REMACK~&ST3~&!ST1~&ST0~
# RSTI~&REMACK~&!ST4~&ST3~&ST2~&ST1~
# RSTI~&REMACK~&!ST4~&ST2~&ST1~&!ST0~
# RSTI~&REMACK~&ST3~&!ST2~&!ST1~&ST0~
# RSTI~&ST4~&ST3~&ST2~&ST1~&!ST0~
# RSTI~&!FBST1~;

!FBST1~ = RSTI~&REMACK~&ST4~&ST2~&!ST1~&ST0~
# RSTI~&ST4~&ST3~&!ST2~&!ST1~&ST0~
# RSTI~&REMACK~&ST4~&!ST2~&ST1~&!ST0~;

!ST0~ := RSTI~&REMACK~&!ST4~&ST2~&ST1~&ST0~
# RSTI~&REMACK~&!ST4~&ST3~&ST1~&ST0~
# RSTI~&ALE1&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&!REMREQ~&!RRQ&ST4~&ST3~&ST2~&ST1~&ST0~
# RSTI~&REMACK~&ST4~&!ST3~&!ST2~&ST1~&ST0~
# RSTI~&REMACK~&ST4~&ST2~&!ST1~&ST0~
# RSTI~&ST4~&ST3~&!ST2~&!ST1~&ST0~
# RSTI~&ALE1&BS0&BS1&ST4~&ST3~&ST2~&ST1~&ST0~;

```

End;

FIGURE 16. Equations for U200 (Continued)

TL/F/10429-27

MODULE U201R
 TITLE
 'Control Signal Generator #1 (from B5-3)
 Device:U201 Filename:U201R
 J. Margeson
 Date: 12/15/88
 Revision: 2.0'

U201RJ device 'p20v8r';

CK1A	pin	1;
ST0~	pin	2;
ST1~	pin	3;
ST2~	pin	4;
ST3~	pin	5;
ST4~	pin	6;
REMAC~	pin	7;
MUXR~	pin	8;
MUXW~	pin	9;
MUXWD~	pin	10;
MUXRD~	pin	11;
GND	pin	12;
OE~	pin	13;
NC14	pin	14;
UARTWR~	pin	15;
WR~	pin	16;
DOE~	pin	17;
NCASIN~	pin	18;
RAS~	pin	19;
DTOE~	pin	20;
DLE~	pin	21;
ST634	pin	22;
NC23	pin	23;
VCC	pin	24;

Equations

```
!UARTWR~ := MUXR~&!MUXW~& ST4~& ST3~&!ST2~&!ST1~& ST0~
# MUXR~&!MUXW~& ST4~& ST3~& ST2~&!ST1~&!ST0~;
```

```
!WR~ := !MUXWD~&ST4~&!ST3~&ST2~&ST1~
# !ST3~&!ST2~&!ST1~
# MUXR~&!MUXW~&ST4~&ST3~&!ST2~&ST1~&ST0~;
```

```
!DTOE~ := !MUXR~&!ST2~&ST1~&ST0~
# !ST3~&!ST2~&ST1~&ST0~
# !ST4~&ST3~&ST2~&!ST0~
# !ST4~&ST3~&ST2~&!ST1~
# MUXWD~&ST4~&!ST3~&ST1~&ST0~
# !MUXR~&ST4~&!ST1~&!ST0~
# !ST3~&!ST1~&!ST0~
# !ST4~&!ST2~&ST0~;
```

```
!DLE~ := !ST3~&!ST2~&ST1~&ST0~
# !MUXRD~&MUXWD~&ST4~&!ST3~&ST1~&ST0~;
```

FIGURE 17. Equations for U201

TL/F/10429-28

```

!DOE~ := !MUXW~&ST4~&ST3~&ST2~&!ST1~&ST0~
# !MUXWD~&ST4~&!ST3~&ST2~&ST1~
# !ST3~&!ST2~&!ST0~
# !ST3~&!ST2~&!ST1~
# MUXR~&ST4~&!ST2~&!ST1~&!ST0~;

!RAS~ := !ST4~&!ST2~&ST1~
# !ST2~&ST0~
# ST4~&!ST1~
# ST3~&!ST1~&ST0~
# !ST3~&ST1~;

!NCASIN~ := ST3~&!ST1~
# ST4~&ST2~&!ST1~
# !ST3~&!ST2~&ST1~
# !ST4~&ST1~&ST0~;

!ST634 := !ST2~&!ST0~
# !ST2~&ST1~
# !ST3~
# !ST4~
# ST2~&ST0~;

```

End;

TL/F/10428-29

FIGURE 17. Equations for U201 (Continued)

```

MODULE U202R
TITLE
'Control Signal Generator #2 (from B5-3)
Device:U202  Filename:U202R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U202RJ device 'p20v8r';
```

```

CK1A      pin    1;
ST0~     pin    2;
ST1~     pin    3;
ST2~     pin    4;
ST3~     pin    5;
ST4~     pin    6;
REMACK~  pin    7;
MUXR~    pin    8;
MUXW~    pin    9;
RSTI~    pin   10;
NC11     pin   11;
GND      pin   12;
OE~      pin   13;
NC14     pin   14;
RSTRRQ~  pin   15;
RSTALE~  pin   16;
ST8      pin   17;
RDY      pin   18;
IOEN     pin   19;
DDIN     pin   20;
DBE~     pin   21;
RFIP     pin   22;
NC23     pin   23;
VCC      pin   24;

```

Equations

```

!DDIN := !MUXR~&ST4~&ST3~&!ST2~&ST0~
#      !MUXR~&ST3~&ST2~&!ST1~&!ST0~;

!DBE~ := REMACK~&ST4~&ST3~&!ST2~&ST0~
#      REMACK~&ST3~&ST2~&!ST1~&!ST0~;

!RDY := !ST3~&!ST1~
#      !ST4~&!ST1~
#      ST2~&ST1~
#      ST1~&!ST0~
#      !REMACK~&!ST0~
#      ST4~&ST3~&!ST2~;

!RSTALE~ := REMACK~&ST3~&ST2~&ST1~&!ST0~
#      !ST4~&ST3~&ST2~&!ST0~
#      ST4~&ST2~&!ST1~&ST0~
#      !RSTI~;

!RSTRRQ~ := !ST4~&ST3~&ST2~&ST1~&ST0~

```

FIGURE 18. Equations for U202

TL/F/10429-30

```
# !RSTI~;  
!RFIP := ST4~;  
!IOEN := !ST4~  
# !ST3~  
# !ST2~&!ST0~  
# ST1~&!ST0~  
# ST2~&ST0~;  
  
!ST8 := ST0~  
# ST1~  
# ST2~  
# !ST4~;
```

End;

TL/F/10429-31

FIGURE 18. Equations for U202 (Continued)

```

MODULE U203R
TITLE
'CAS
Device:U203   Filename:U203R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U203RJ device 'p20v8r';
```

```

CK1D      pin      1;
A23       pin      2;
A22       pin      3;
A21       pin      4;
A18       pin      5;
A17       pin      6;
WMASK0    pin      7;
WMASK1    pin      8;
WMASK2    pin      9;
WMASK3    pin     10;
ST8       pin     11;
GND       pin     12;
OE~       pin     13;
RFIP      pin     14;
MUXR~     pin     15;
NC16      pin     16;
CAS0~     pin     17;
CAS1~     pin     18;
CAS2~     pin     19;
CAS3~     pin     20;
A20       pin     21;
LADDR~    pin     22;
NCASIN~   pin     23;
VCC       pin     24;

```

Equations

```

!CAS3~ := !LADDR~ & !A23 & A22 & A18 & A17 & !NCASIN~ & !ST8
# LADDR~ & WMASK3 & !NCASIN~ & !ST8
# RFIP & !NCASIN~ & !ST8
# ST8 & !CAS3~;

```

```

!CAS2~ := !LADDR~ & !A23 & A22 & A18 & !A17 & !NCASIN~ & !ST8
# LADDR~ & WMASK2 & !NCASIN~ & !ST8
# RFIP & !NCASIN~ & !ST8
# ST8 & !CAS2~;

```

```

!CAS1~ := !LADDR~ & !A23 & A22 & !A18 & A17 & !NCASIN~ & !ST8
# LADDR~ & WMASK1 & !NCASIN~ & !ST8
# RFIP & !NCASIN~ & !ST8
# ST8 & !CAS1~;

```

```

!CAS0~ := !LADDR~ & !A23 & A22 & !A18 & !A17 & !NCASIN~ & !ST8
# LADDR~ & WMASK0 & !NCASIN~ & !ST8
# RFIP & !NCASIN~ & !ST8
# ST8 & !CAS0~;

```

```
End;
```

FIGURE 19. Equations for U203

```

MODULE U204R
TITLE
'Device Enables
Device:U204   Filename:U204R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U204RJ device 'p16v8c';
```

```

IOEN      pin    1;
A23       pin    2;
A22       pin    3;
A21       pin    4;
A20       pin    5;
A3        pin    6;
A2        pin    7;
A1        pin    8;
A0        pin    9;
GND       pin   10;
MUXW~     pin   11;
FSE3~     pin   12;
FSE2~     pin   13;
FSE1~     pin   14;
FSE0~     pin   15;
CRE~      pin   16;
ENUART~   pin   17;
ENPROM~   pin   18;
LWMASK    pin   19;
VCC       pin   20;

```

Equations

```

!FSE3~ = A23 & !A22 & A21 & !A20 & A3 & IOEN;
!FSE2~ = A23 & !A22 & A21 & !A20 & A2 & IOEN;
!FSE1~ = A23 & !A22 & A21 & !A20 & A1 & IOEN;
!FSE0~ = A23 & !A22 & A21 & !A20 & A0 & IOEN;
!CRE~  = A23 & !A22 & !A21 & !A20 & !A3 & !A2 & !A1 & A0 & IOEN;
!ENUART~ = A23 & A22 & !A21 & !A20 & IOEN;
!ENPROM~ = !A23 & !A22 & !A21 & !A20 & IOEN & MUXW~;
!LWMASK = !A23 # !A22 # !A21 # A20 # !IOEN # MUXW~;

```

```
End;
```

FIGURE 20. Equations for U204

TL/F/10429-33

```

MODULE U205R
TITLE
'BE
Device:U205  Filename:U205R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U205RJ device 'p20v8r';
```

```

CK1A      pin      1;
ST8       pin      2;
ST634     pin      3;
A23       pin      4;
A22       pin      5;
A21       pin      6;
A20       pin      7;
A18       pin      8;
A17       pin      9;
A3        pin     10;
A2        pin     11;
GND       pin     12;
OE~       pin     13;
A1        pin     14;
DRCYCLE~  pin     15;
INTPL     pin     16;
BE3~      pin     17;
BE2~      pin     18;
BE1~      pin     19;
BE0~      pin     20;
NC21     pin     21;
BSE       pin     22;
A0        pin     23;
VCC       pin     24;

```

Equations

```

!BE3~ := ST634 & !A23 & A22 & A18 & A17
# ST634 & A23 & !A22 & !A21 & !A3 & !A2 & !A1 & A0
# ST634 & A23 & !A22 & A21 & A3
# INTPL & BSE & !DRCYCLE~ & !ST8
# ST8 & !BE3~;

```

```

!BE2~ := ST634 & !A23 & A22 & A18 & !A17
# ST634 & A23 & !A22 & !A21 & !A3 & !A2 & !A1 & A0
# ST634 & A23 & !A22 & A21 & A2
# INTPL & BSE & !DRCYCLE~ & !ST8
# ST8 & !BE2~;

```

```

!BE1~ := ST634 & !A23 & A22 & !A18 & A17
# ST634 & A23 & !A22 & !A21 & !A3 & !A2 & !A1 & A0
# ST634 & A23 & !A22 & A21 & A1
# INTPL & BSE & !DRCYCLE~ & !ST8
# ST8 & !BE1~;

```

```

!BE0~ := ST634 & !A23 & A22 & !A18 & !A17
# ST634 & A23 & !A22 & !A21 & !A3 & !A2 & !A1 & A0
# ST634 & A23 & !A22 & A21 & A0
# INTPL & BSE & !DRCYCLE~ & !ST8
# ST8 & !BE0~;

```

```
End;
```

FIGURE 21. Equations for U205

```

MODULE U206R
TITLE
'
Device:U206   Filename:U206R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U206RJ device 'p20v8r';
```

```

CK1D      pin      1;
MUXR~     pin      2;
IOEN      pin      3;
A23       pin      4;
A22       pin      5;
A21       pin      6;
A20       pin      7;
A18       pin      8;
A17       pin      9;
A3        pin     10;
A2        pin     11;
GND       pin     12;
OE~       pin     13;
A1        pin     14;
PDLE~     pin     15;
POE~      pin     16;
DIR3      pin     17;
DIR2      pin     18;
DIR1      pin     19;
DIR0      pin     20;
RGPINTRES~ pin    21;
ATINTWR~  pin    22;
A0        pin    23;
VCC       pin    24;

```

Equations

```

!DIR3 := MUXR~ # A23 & !A22 # !A18 # !A17;
!DIR2 := MUXR~ # A23 & !A22 # !A18 # A17;
!DIR1 := MUXR~ # A23 & !A22 # A18 # !A17;
!DIR0 := MUXR~ # A23 & !A22 # A18 # A17;

!PDLE~ = A23 & !A22 & !A21 & !A20 & !A3 & !A2 & A1 & !A0 & IOEN;
!POE~  = A23 & !A22 & !A21 & !A20 & !A3 & A2 & !A1 & !A0 & IOEN;

!RGPINTRES~ = A23 & A22 & !A21 & A20 & IOEN & MUXR~;
!ATINTWR~   = A23 & A22 & A21 & A20 & IOEN & MUXR~;

End;

```

FIGURE 22. Equations for U206

TL/F/10429-35

```
MODULE U207R
TITLE
'MIDSCAN, BLANK~ GENERATOR
Device:U207  Filename:U207R
Jim Margeson
Date: 12/15/88
Revision: 2.0'
```

```
U207RJ device 'p16v8r';
```

```
LCK      pin    1;
BLANK~   pin    2;
DRREQ~   pin    3;
RST~     pin    4;
L_DR_REQ~ pin    5;
I5       pin    6;
I6       pin    7;
I7       pin    8;
I8       pin    9;
GND      pin   10;
I9       pin   11;
DR_REQ   pin   12;
RST_REQ~ pin   13;
DD       pin   14;
F3       pin   15;
D_BLANK~ pin   16;
F5       pin   17;
F6       pin   18;
MISCAN   pin   19;
VCC      pin   20;
```

Equations

```
DR_REQ = !DRREQ~;
```

```
F3 := BLANK~;
```

```
DD = D_BLANK~;
```

```
D_BLANK~ := F3;
```

```
RST_REQ~ = RST~ & !F6;
```

```
F5 := !L_DR_REQ~ & !F5;
```

```
F6 := !L_DR_REQ~ & F5 & !F6;
```

```
MISCAN = !F6 & D_BLANK~;
```

```
End;
```

FIGURE 23. Equations for U207

TL/F/10429-36

```

MODULE U208R
TITLE
'DRAM REFRESH REQUEST, GSC, READ + WRITE MUX
Device:U208   Filename:U208R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U208RJ device 'p16v8r';
```

```

LCKL1   pin    1;
LCKL    pin    2;
BLANKD~ pin    3;
NC4     pin    4;
R~      pin    5;
W~      pin    6;
REMRL   pin    7;
REMWL   pin    8;
REMACK  pin    9;
GND     pin   10;
OE~     pin   11;
GSC     pin   12;
O6      pin   13;
O5      pin   14;
O4      pin   15;
O3      pin   16;
VRRQ    pin   17;
MUXR~   pin   18;
MUXW~   pin   19;
VCC     pin   20;

```

Equations

```

!VRRQ := !VRRQ & O3 & O4 & O5
# !VRRQ & O3 & O4 & O6
# VRRQ & !O3 & !O4 & !O5 & !O6;

```

```

!O3 := VRRQ & !O3 & O5
# VRRQ & !O3 & O4
# VRRQ & !O3 & O6
# VRRQ & O3 & !O4 & !O5 & !O6;

```

```

!O4 := VRRQ & !O4 & O6
# VRRQ & !O4 & O5
# VRRQ & O4 & !O5 & !O6;

```

```

!O5 := VRRQ & O5 & !O6
# VRRQ & !O5 & O6
# O3 & O4 & !O5 & O6
# O3 & O4 & O5 & !O6;

```

```

!O6 := O3 & O4 & O6
# VRRQ & O6;

```

```
!GSC = !BLANKD~ # !LCKL;
```

```

!MUXW~ = !W~ & !REMACK
# REMWL & REMACK;

```

```

!MUXR~ = !R~ & !REMACK
# REMRL & REMACK;

```

```
End;
```

FIGURE 24. Equations for U208

TL/F/10429-37

MODULE U209R

TITLE

'

Device:U209 Filename:U209R

Jim Margeson

Date: 12/15/88

Revision: 2.0'

U209R device 'p16v8r';

```

LCK0      pin    1;
DDBSE     pin    2;
INTPL     pin    3;
BS1       pin    4;
BS0       pin    5;
REMACK    pin    6;
ATINTRES~ pin    7;
ATINTWR~  pin    8;
D0        pin    9;
GND       pin   10;
OE~       pin   11;
DRCYCLE~  pin   12;
LADDR~    pin   13;
LCK       pin   14;
LCK0D4    pin   15;
LCK0D2    pin   16;
RSTI~     pin   17;
ATINT~    pin   18;
NC19      pin   19;
VCC       pin   20;

```

Equations

```
!LCK0D2 := LCK0D2;
```

```
!LCK0D4 := !LCK0D4 & LCK0D2
# LCK0D4 & !LCK0D2;
```

```
!LCK := !LCK & LCK0D4
# !LCK & LCK0D2
# LCK & !LCK0D4 & !LCK0D2;
```

```
!DRCYCLE~ = BS1 & !BS0 & !REMACK;
```

```
!LADDR~ = DDBSE & INTPL & BS1 & !BS0
# !BS1
# REMACK;
```

```
!ATINT~ = D0 & !ATINTWR~ & RSTI~
# D0 & !ATINT~ & ATINTRES~ & RSTI~
# ATINTWR~ & !ATINT~ & ATINTRES~ & RSTI~;
```

End;

FIGURE 25. Equations for U209

TL/F/10428-38

```

MODULE U210R
TITLE
'SE~, LD~
Device:U210   Filename:U210R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U210RJ device 'pl6v8r';
```

```

CK1C      pin    1;
GSC       pin    2;
BLANKD~   pin    3;
NC4       pin    4;
NC5       pin    5;
NC6       pin    6;
NC7       pin    7;
NC8       pin    8;
NC9       pin    9;
GND       pin   10;
OE~       pin   11;
SE3~     pin   12;
SE2~     pin   13;
LD~       pin   14;
O5        pin   15;
O4        pin   16;
O3        pin   17;
SE1~     pin   18;
SE0~     pin   19;
VCC       pin   20;

```

Equations

```

!SE0~ = O3 & GSC;
!SE1~ = !O3 & GSC;
!SE2~ = O3 & !GSC;
!SE3~ = !O3 & !GSC;

!O3 := !O3 & O5 & BLANKD~
# !O3 & O4 & BLANKD~
# O3 & !O4 & !O5 & BLANKD~;

!O4 := O5 & !O4 & BLANKD~
# !O5 & O4 & BLANKD~;

!O5 := O5 & BLANKD~;

!LD~ := O4 & O5;

End;

```

TL/F/10429-39

FIGURE 26. Equations for U210

```

MODULE U211R
TITLE
'latch and blank video data
Device:U211  Filename:U211R
Jim Margeson
Date: 12/15/88
Revision: 2.0'

```

```
U211RJ device 'p16v8r';
```

```

CK1C      pin    1;
BLANKD~   pin    2;
GSC       pin    3;
NC4       pin    4;
VD0       pin    5;
VD1       pin    6;
VD2       pin    7;
VD3       pin    8;
NC9       pin    9;
GND       pin   10;
OE~       pin   11;
BLNKDLL   pin   12;
BLNKDLLL  pin   13;
V3        pin   14;
V2        pin   15;
V1        pin   16;
V0        pin   17;
GSCL      pin   18;
BLANKDL~  pin   19;
VCC       pin   20;

```

Equations

```

!GSCL      := !GSC;
!BLANKDL~  := !BLANKD~;
!BLNKDLL   := !BLANKDL~;
!BLNKDLLL  := !BLNKDLL;

!V0        := !VD0 # BLNKDLLL;
!V1        := !VD1 # BLNKDLLL;
!V2        := !VD2 # BLNKDLLL;
!V3        := !VD3 # BLNKDLLL;

```

```
End;
```

TL/F/10429-40

FIGURE 27. Equations for U211

MODULE U220R1
 TITLE
 'Host Address Decode and 16-bit Transfer Request
 Device:U220 Filename:U220R1
 Jim Margeson
 Date: 1/23/88
 Revision: 2.01'

U220RJ1 device 'p16v8c';

AEN	pin	1;
REMLA23	pin	2;
REMLA22	pin	3;
REMLA21	pin	4;
REMLA20	pin	5;
REMLA19	pin	6;
REMLA18	pin	7;
REMLA17	pin	8;
REMA16	pin	9;
GND	pin	10;
BALE	pin	11;
NC12	pin	12;
BDSELL~	pin	13;
AS0A~	pin	14;
AS0D~	pin	15;
ASA0~	pin	16;
M160A~	pin	17;
M160D~	pin	18;
M16A0~	pin	19;
VCC	pin	20;

Equations

!M160A~ = !AS0A~ # AS0A~ # !AS0D~ # AS0D~ # !ASA0~ # ASA0~;
 ENABLE M160A~ = !REMLA23 & !REMLA22 & !REMLA21 & !REMLA20 & REMLA19
 & !REMLA18 & REMLA17 & !AS0A~;

!M160D~ = !AS0A~ # AS0A~ # !AS0D~ # AS0D~ # !ASA0~ # ASA0~;
 ENABLE M160D~ = !REMLA23 & !REMLA22 & !REMLA21 & !REMLA20 & REMLA19
 & REMLA18 & !REMLA17 & !AS0D~;

!M16A0~ = !AS0A~ # AS0A~ # !AS0D~ # AS0D~ # !ASA0~ # ASA0~;
 ENABLE M16A0~ = REMLA23 & !REMLA22 & REMLA21 & !REMLA20 & !REMLA19
 & !REMLA18 & !REMLA17 & !ASA0~;

!BDSELL~ = !AEN & !REMLA23 & !REMLA22 & !REMLA21 & !REMLA20 & REMLA19
 & !REMLA18 & REMLA17 & !REMA16 & !AS0A~ & BALE
 # !AEN & !REMLA23 & !REMLA22 & !REMLA21 & !REMLA20 & REMLA19
 & !REMLA18 & REMLA17 & !REMA16 & !AS0A~ & !BDSELL~
 # !AEN & !REMLA23 & !REMLA22 & !REMLA21 & !REMLA20 & REMLA19
 & !REMLA18 & !REMLA17 & REMA16 & !AS0D~ & BALE
 # !AEN & !REMLA23 & !REMLA22 & !REMLA21 & !REMLA20 & REMLA19
 & REMLA18 & !REMLA17 & REMA16 & !AS0D~ & !BDSELL~
 # !AEN & REMLA23 & !REMLA22 & REMLA21 & !REMLA20 & !REMLA19
 & !REMLA18 & !REMLA17 & !ASA0~ & BALE
 # !AEN & REMLA23 & !REMLA22 & REMLA21 & !REMLA20 & !REMLA19
 & !REMLA18 & !REMLA17 & !ASA0~ & !BDSELL~
 # !BALE & !BDSELL~;

End;

FIGURE 28. Equations for U220

TL/F/10429-41

```

MODULE U221R1
TITLE
'Host Interface Sequencer
Device:U221   Filename:U221R1
Jim Margeson
Date: 2/4/88
Revision: 2.01'

```

```
U221RJ1 device 'p16v8r';
```

```

CK1B      pin    1;
BDSELL~   pin    2;
REMA15    pin    3;
REMACK~   pin    4;
REMA14    pin    5;
REMR~     pin    6;
REMW~     pin    7;
RSTI~     pin    8;
NC9       pin    9;
GND       pin   10;
OE~       pin   11;
REMDG~    pin   12;
REMACKD   pin   13;
RDYHI~    pin   14;
REMREQ~   pin   15;
REMRL     pin   16;
REMWL     pin   17;
NC18     pin   18;
NC19     pin   19;
VCC      pin   20;

```

Equations

```

!REMRL := REMR~ # BDSELL~ # !RSTI~;
!REMWL := REMW~ # BDSELL~ # !RSTI~;

!REMREQ~ := !BDSELL~ & !REMA15      & REMWL & !REMACKD & RDYHI~ & RSTI~
          # !BDSELL~ & REMA15 & !REMA14 & REMWL & !REMACKD & RDYHI~ & RSTI~
          # !BDSELL~ & !REMA15      & REMRL & !REMACKD & RDYHI~ & RSTI~
          # !BDSELL~ & REMA15 & !REMA14 & REMRL & !REMACKD & RDYHI~ & RSTI~;

!REMACKD := REMACK~ # BDSELL~ # !RSTI~;

!RDYHI~ := !BDSELL~ & !REMA15 & REMREQ~ & REMACK~ & REMACKD & RDYHI~
          & RSTI~
          # !BDSELL~ & REMA15 & !REMA14 & REMREQ~ & REMACK~ & REMACKD
          & RDYHI~ & RSTI~
          # REMWL & !RDYHI~ & RSTI~
          # REMRL & !RDYHI~ & RSTI~;

!REMDG~ = !REMACK~ & REMWL & RSTI~
          # !BDSELL~ & !REMA15 &          !REMR~ & RSTI~
          # !BDSELL~ & REMA15 & !REMA14 & !REMR~ & RSTI~;

```

```
End;
```

FIGURE 29. Equations for U221

MODULE U222R

TITLE

'

Device:U222 Filename:U222R

Jim Margeson

Date: 12/15/88

Revision: 2.0'

U222RJ device 'p20v8r';

```

CK1B      pin    1;
CREGEN~   pin    2;
REMA13    pin    3;
REMA12    pin    4;
REMA11    pin    5;
REMRST    pin    6;
PBRST     pin    7;
PBNMI     pin    8;
UARTINT   pin    9;
REMD0     pin   10;
RGPINTRES~ pin 11;
GND       pin   12;
OE~       pin   13;
REMR~     pin   14;
STATROC~  pin   15;
AREGC     pin   16;
AREGROC~  pin   17;
INT~      pin   18;
NMI~      pin   19;
SRSTL~    pin   20;
RSTI~     pin   21;
ATINTRES~ pin   22;
REMW~     pin   23;
VCC       pin   24;

```

Equations

```

!SRSTL~ = REMD0 & !CREGEN~ & REMA13 & !REMA12 & !REMA11 & !REMW~
        & !REMRST & !PBRST
# REMD0   & !SRSTL~ & !REMRST & !PBRST
# CREGEN~ & !SRSTL~ & !REMRST & !PBRST
# !REMA13 & !SRSTL~ & !REMRST & !PBRST
# REMA12 & !SRSTL~ & !REMRST & !PBRST
# REMA11 & !SRSTL~ & !REMRST & !PBRST
# REMW~  & !SRSTL~ & !REMRST & !PBRST;
!RSTI~  := !SRSTL~ # REMRST # PBRST;

```

```

!INT~ = REMD0 & !CREGEN~ & REMA13 & !REMA12 & REMA11 & !REMW~
      & RGPINTRES~ & RSTI~
# REMD0   & !INT~ & RGPINTRES~ & RSTI~
# CREGEN~ & !INT~ & RGPINTRES~ & RSTI~
# !REMA13 & !INT~ & RGPINTRES~ & RSTI~
# REMA12 & !INT~ & RGPINTRES~ & RSTI~
# !REMA11 & !INT~ & RGPINTRES~ & RSTI~
# REMW~  & !INT~ & RGPINTRES~ & RSTI~;

```

```

!NMI~ = PBNMI # !CREGEN~ & !REMA13 & !REMA12 & REMA11 & !REMW~;
!AREGC = CREGEN~ # REMA13 # REMA12 # REMA11 # REMW~;
!AREGROC~ = !CREGEN~ & !REMA13 & !REMA12 & !REMA11 & !REMR~;
!STATROC~ = !CREGEN~ & !REMA13 & REMA12 & REMA11 & !REMR~;
!ATINTRES~ = !CREGEN~ & !REMA13 & REMA12 & !REMA11 & !REMW~ # !RSTI~;

```

End;

FIGURE 30. Equations for U222

TL/F/10429-43

MODULE U223R

TITLE

'

Device:U223 Filename:U223R

Jim Margeson

Date: 12/15/88

Revision: 2.0'

U223RJ device 'pl6v8c';

```

NC1      pin 1;
BDSELL~  pin 2;
REMA15   pin 3;
REMA14   pin 4;
RDYHI~   pin 5;
REMACK~  pin 6;
REMR~    pin 7;
REMW~    pin 8;
RSTI~    pin 9;
GND      pin 10;
NC11     pin 11;
FSOC~    pin 12;
MSOC~    pin 13;
CREGEN~  pin 14;
WAIT~    pin 15;
REMRDY   pin 16;
A13      pin 17;
NC18     pin 18;
NC19     pin 19;
VCC      pin 20;

```

Equations

```

!WAIT~ = !BDSELL~ & !REMA15 &          RDYHI~ & !REMW~ & RSTI~
#       !BDSELL~ & !REMA15 &          RDYHI~ & !REMR~ & RSTI~
#       !BDSELL~ & REMA15 & !REMA14 & RDYHI~ & !REMW~ & RSTI~
#       !BDSELL~ & REMA15 & !REMA14 & RDYHI~ & !REMR~ & RSTI~;

```

```

!REMRDY = !BDSELL~ # BDSELL~;

```

```

ENABLE REMRDY = !WAIT~;

```

```

!MSOC~ = !REMACK~ & !REMA15;

```

```

!FSOC~ = !REMACK~ & REMA15 & !REMA14;

```

```

!CREGEN~ = !BDSELL~ & REMA15 & REMA14;

```

```

!A13 = !REMA14 & !REMA15;

```

```

ENABLE A13 = !REMACK~;

```

```

End;

```

FIGURE 31. Equations for U223

TL/F/10429-44

QUANTITIES ARE FOR 1 BOARD

PARTS FOR WAVE SOLDER:

1: 14-PIN SOCKETS 4

U5

U6

U28

U225

2: 16-PIN SOCKETS 4

U12

U13

U27

RN1

3: 20-PIN SOCKETS 36

U25

U3

U11

U29

U9

U10

U102

U103

U112

U113

U122

U123

U132

U133

U7

U8

U16

U19

U20

U21

U22

U4

U18

U100

U110

U120

U130

U204

U207

U208

U209

U210

U211

U221

U220

U223

4: 24-PIN DIP SOCKETS 11

U14

U15

U23

U24

U200

U201

U202

U203

U205

U206

U222

5: 68-PIN PLCC SOCKET 1

U1

6: 44-PIN PLCC SOCKET 6

U2

U26

U101

U111

U121

U131

7: 24-PIN SIP SOCKETS 16

U104

U105

U106

U107

U114

U115

U116

U117

U124

U125

U126

U127

U134

U135

U136

U137

FIGURE 32. Parts List

QUANTITIES ARE FOR 1 BOARD

PARTS FOR WAVE SOLDER:

8:	25 pF CAPACITOR	2	14:	15 OHM RESISTOR	2
C3			R4		
C4			R5		
9:	0.1 μ F CAPACITOR	81	15:	220 OHM RESISTOR	4
C5			R6		
C6			R11		
C7			R12		
C8			R16		
C9			16:	470 OHM RESISTOR	3
C17-C92			R7		
10:	25 μ F or 22 μ F CAPACITOR, 15V OR GREATER	10	R8		
C1	(NOTE 2)		R10		
C2	(NOTE 2)		17:	2.7k RESISTOR	4
C10	(NOTE 2)		R13		
C11	(NOTE 2)		14		
C12	(NOTE 2)		15		
C13	(NOTE 2)		17	(NOTE 4)	
C14	(NOTE 2)		18:	PUSH BUTTON SWITCH	2
C15	(NOTE 2)		SW1		
C16	(NOTE 2)		SW2		
C93	(NOTE 2)		19:	20 MHz CRYSTAL	1
11:	LED	1	Y1		
D1	(NOTE 3)		20:	9-PIN CONNECTOR	1
12:	12k RESISTOR	2	J1		
R1			21:	25-PIN CONNECTOR	
R2			J2		
13:	330 OHM RESISTOR	1	22:	7-PIN POWER CONNECTOR	1
R3			J3	(NOTE 5)	
			23:	3 JUMPER PINS	1
			J7		

FIGURE 32. Parts List (Continued)

QUANTITIES ARE FOR 1 BOARD

PARTS FOR WAVE SOLDER:

24: PAIR OF JUMPER PINS 20

W1

W2

W3

W4

W5 (4 PIN SQUARE)

W5

W6 (4 PIN SQUARE)

W6

W10

W11

W12

W13

W14

W15

W16

W17

W18

W19

W20

W21

NOTE 1: DO NOT USE SOCKET FOR SPARE U224

NOTE 2: POLARITY SHOWN IN THE ASSEMBLY DRAWING

NOTE 3: ANODE AND CATHODE MARKED "A" and "C" ON SILK SCREEN

NOTE 4: MOUNT AS SHOWN IN ASSEMBLY DRAWING

NOTE 5: PLASTIC FLANGE TOWARD C75

FIGURE 32. Parts List (Continued)

PARTS FOR INSERTION:

U225	74LS14	1	U211	16V8_25	
U5	74AS74	2	U221	16V8_25	
U6	74AS74		U208	16V8_12	4
U28	184320SC	1	U209	16V8_12	
U12	74AS157	2	U220	16V8_12	
U13	74AS157		U223	16V8_12	
U27	MAX232	1	U14	DM87SR193N	2
RN1	22RPAK	1	U15	DM87SR193N	
U25	74ALS240	1	U23	74ALS646 (NOTE 7)	2
U3	74ALS244	1	U24	74ALS646	
U11	74ACT244	2	U200	20V8_12	1
U29	74ACT244		U201	20V8_25	
U9	74ACT245	10	U202	20V8_25	
U10	74ACT245		U203	20V8_25	
U102	74ACT245		U205	20V8_25	
U103	74ACT245		U206	20V8_25	
U112	74ACT245		U222	20V8_25	
U113	74ACT245		U1	DF8500	1
U122	74ACT245		U2	DF8512	1
U123	74ACT245		U26	NS18C450V	1
U132	74ACT245		U101	DF8511V	4
U133	74ACT245		U111	DF8511V	
U7	74ACT373	7	U121	DF8511V	
U8	74ACT373		U131	DF8511V	
U16	74ACT373		U104	M5M4C264L 120 ns	16
U19	74ACT373		U105	M5M4C264L	
U20	74ACT373		U106	M5M4C264L	
U21	74ACT373		U107	M5M4C264L	
U22	74ACT373		U114	M5M4C264L	
U4	74ACT374	1	U115	M5M4C264L	
U18	74ALS374	1	U116	M5M4C264L	
U100	74LS195A (NOTE 6)	4	U117	M5M4C264L	
U110	74LS195A		U124	M5M4C264L	
U120	74LS195A		U125	M5M4C264L	
U130	74LS195A		U126	M5M4C264L	
U204	16V8_25	5	U127	M5M4C264L	
U207	16V8_25		U134	M5M4C264L	
U210	16V8_25		U135	M5M4C264L	
			U136	M5M4C264L	
			U137	M5M4C264L	

NOTE 6: CENTERED IN 20-PIN SOCKET

NOTE 7: OR 74AC646

FIGURE 32. Parts List (Continued)



Section 4
**Physical Dimensions/
Appendices**



Section 4 Contents

Appendix A: Related Datasheets	4-3
Physical Dimensions	4-4
Bookshelf	
Distributors	

APPENDIX A: Related Datasheets

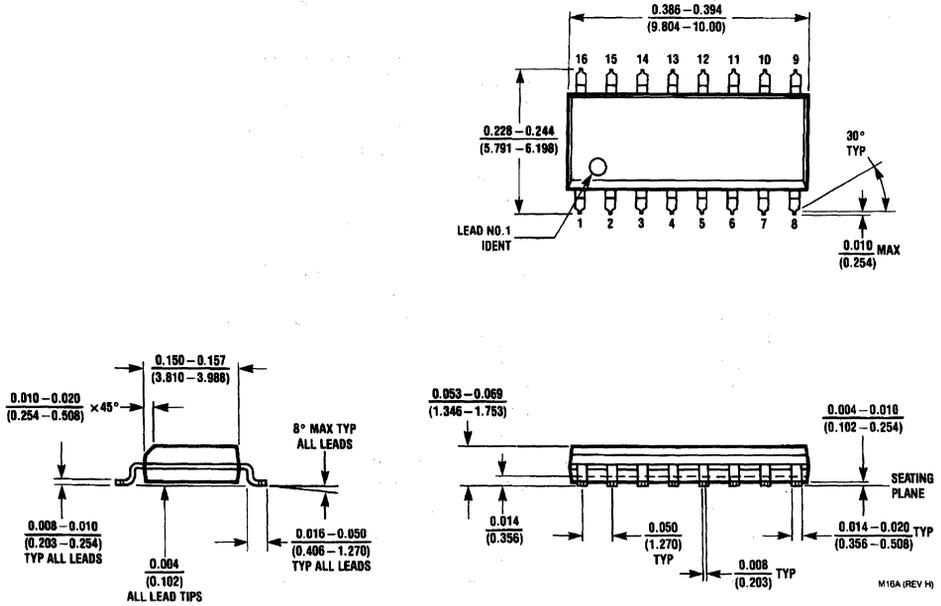
The table below lists a series of other products/functions that may have interest and which are available as datasheets from National Semiconductor.

Product Number	Product Description
LH4002	Wideband Video Buffer
LH4003	Precision RF Closed Loop Buffer
LH4004	Wideband FET Input Buffer Amplifier
LH4101	Wideband High Current Operational Amplifier
LM592	Differential Video Amplifier
LM675	Power Operational Amplifier
LM1203	RGB Video Amplifier System
LM1211	Broadband Demodulator SSC129
LM1578	Switching Regulator
LM1823	Video IF Amplifier/PLL Detection System
LM1875	20W Power Audio Amplifier
LM1881	Video Sync Separator
LM1884	TV Stereo Decoder
LM1886	TV Video Matrix D to A
LM1889	TV Video Modulator
LM2889	TV Video Modulator
LM6161	High Speed Operational Amplifiers
LM6164	High Speed Operational Amplifiers
LM6165	High Speed Operational Amplifiers
LMC555	CMOS Timer
NS32CG16	High-Performance Printer/Display Processor

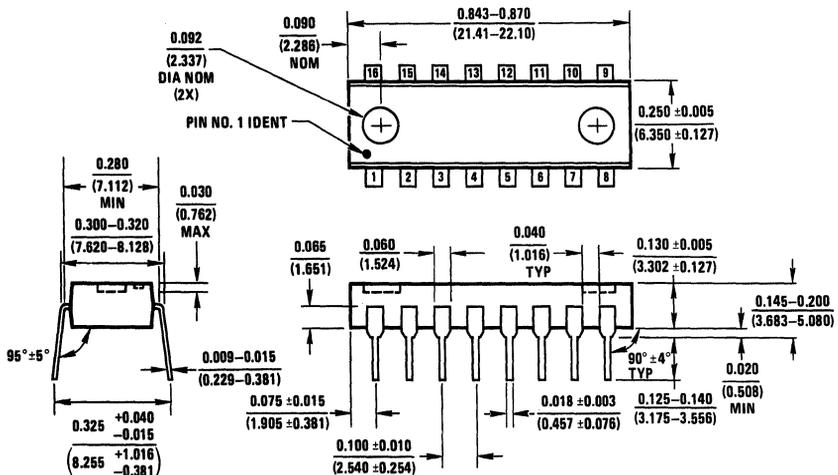
APPLICATION NOTES

Note No.	Description
AN-391	The LM1823—A High Quality TV Video IF Amplifier and Synchronous Detector for Cable Receivers
AN-402	LM2889 R.F. Modulator

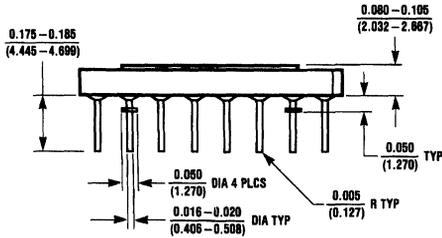
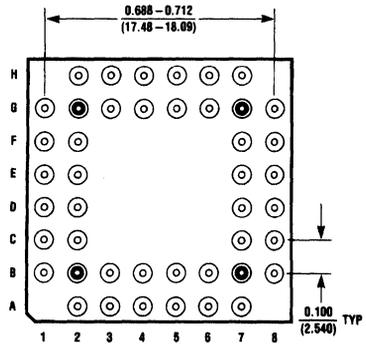
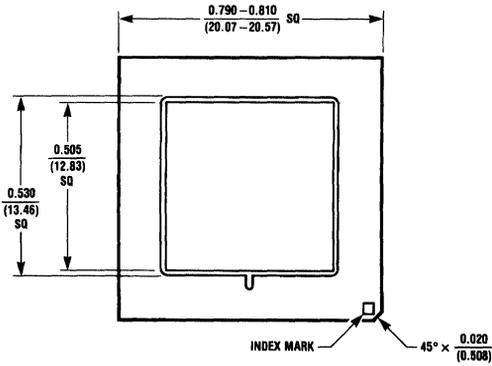
16 Lead (0.150" Wide) Small Outline Molded Package (M) NS Package Number M16A



16 Lead Molded Dual In-Line Package (N) NS Package Number N16A

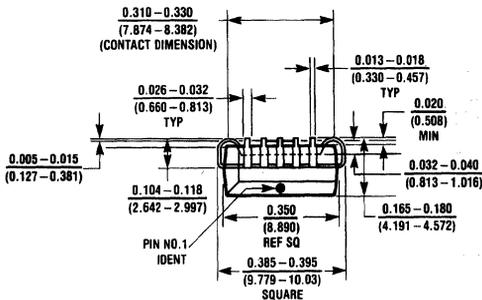
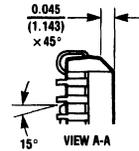
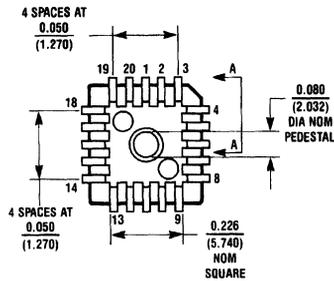


44 Pin Ceramic Pin Grid Array, Cavity Up NS Package Number U44A



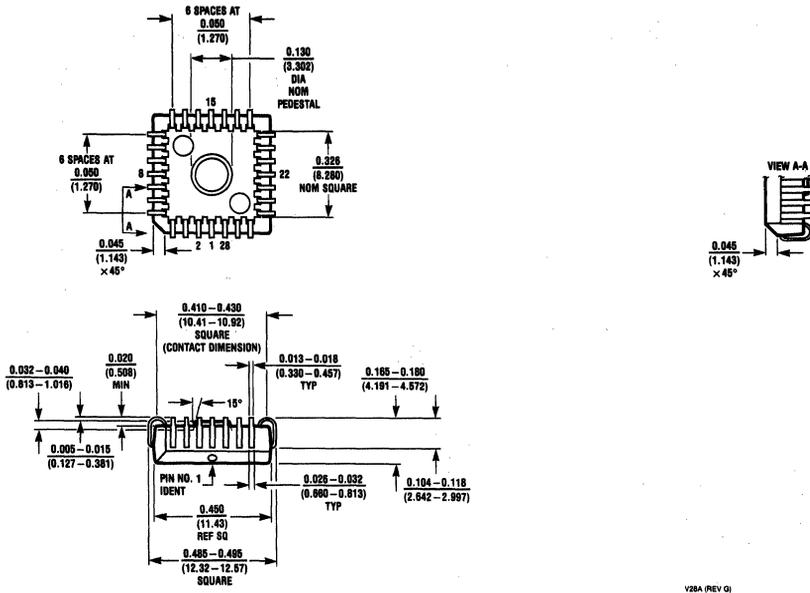
U44A (REV A)

20 Lead Plastic Chip Carrier (PCC) NS Package Number V20A

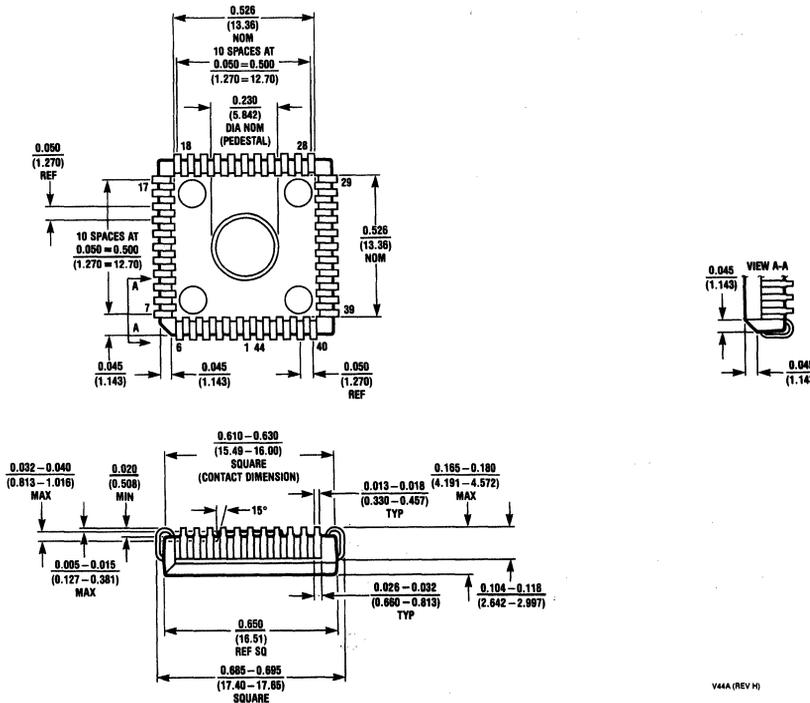


V20A (REV J)

28 Lead Plastic Chip Carrier (PCC) NS Package Number V28A



44 Lead Plastic Chip Carrier (PCC) NS Package Number V44A





National Semiconductor

Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you informed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement.

Please send them to:

Technical Communications Dept. M/S 16300
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

ALS/AS LOGIC DATABOOK—1987

Introduction to Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky

ASIC DESIGN MANUAL/GATE ARRAYS & STANDARD CELLS—1987

SSI/MSI Functions • Peripheral Functions • LSI/VLSI Functions • Design Guidelines • Packaging

CMOS LOGIC DATABOOK—1988

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC
MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • Surface Mount

DATA COMMUNICATION/LAN/UART DATABOOK—Rev. 1—1988

LAN IEEE 802.3 • High Speed Serial/IBM Data Communications • ISDN Components • UARTs
Modems • Transmission Line Drivers/Receivers

DISCRETE SEMICONDUCTOR PRODUCTS DATABOOK—1989

Selection Guide and Cross Reference Guides • Diodes • Bipolar NPN Transistors
Bipolar PNP Transistors • JFET Transistors • Surface Mount Products • Pro-Electron Series
Consumer Series • Power Components • Transistor Datasheets • Process Characteristics

DRAM MANAGEMENT HANDBOOK—1988

Dynamic Memory Control • Error Detection and Correction • Microprocessor Applications for the
DP8408A/09A/17/18/19/28/29 • Microprocessor Applications for the DP8420A/21A/22A

F100K DATABOOK—1989

Family Overview • F100K Datasheets • 11C Datasheets • 10K and 100K Memory Datasheets
Design Guide • Circuit Basics • Logic Design • Transmission Line Concepts • System Considerations
Power Distribution and Thermal Considerations • Testing Techniques • Quality Assurance and Reliability

FACT™ ADVANCED CMOS LOGIC DATABOOK—1989

Description and Family Characteristics • Ratings, Specifications and Waveforms
Design Considerations • 54AC/74ACXXX • 54ACT/74ACTXXX

FAST® ADVANCED SCHOTTKY TTL LOGIC DATABOOK—1988

Circuit Characteristics • Ratings, Specifications and Waveforms • Design Considerations • 54F/74FXXX

FAST® APPLICATIONS HANDBOOK—REPRINT

Reprint of 1987 Fairchild FAST Applications Handbook

Contains application information on the FAST family: Introduction • Multiplexers • Decoders • Encoders
Operators • FIFOs • Counters • TTL Small Scale Integration • Line Driving and System Design
FAST Characteristics and Testing • Packaging Characteristics • Index

INTERFACE DATABOOK—1988

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral Power Drivers • Display Drivers
Memory Support • Microprocessor Support • Level Translators and Buffers • Frequency Synthesis • Hi-Rel Interface

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LINEAR 1 DATABOOK—1988

Voltage Regulators • Operational Amplifiers • Buffers • Voltage Comparators • Instrumentation Amplifiers • Surface Mount

LINEAR 2 DATABOOK—1988

Active Filters • Analog Switches/Multiplexers • Analog-to-Digital • Digital-to-Analog • Sample and Hold
Sensors • Voltage References • Surface Mount

LINEAR 3 DATABOOK—1988

Audio Circuits • Radio Circuits • Video Circuits • Motion Control • Special Functions • Surface Mount

LS/S/TTL DATABOOK—1987

Introduction to Bipolar Logic • Low Power Schottky • Schottky • TTL • Low Power

MASS STORAGE HANDBOOK—1988

Winchester Disk Preamplifiers • Winchester Disk Servo Control • Winchester Disk Pulse Detectors
Winchester Disk Data Separators/Synchronizers and ENDECs • Winchester Disk Data Controller
SCSI Bus Interface Circuits • Floppy Disk Controllers

MEMORY DATABOOK—1988

PROMs, EPROMs, EEPROMs • Flash EPROMs and EEPROMs • TTL I/O SRAMs
ECL I/O SRAMs • ECL I/O Memory Modules

MICROCONTROLLER DATABOOK—1988

COP400 Family • COP800 Family • COPS Applications • HPC Family • HPC Applications
MICROWIRE and MICROWIRE/PLUS Peripherals • Display/Terminal Management Processor (TMP)
Microcontroller Development Tools

PROGRAMMABLE LOGIC DATABOOK & DESIGN MANUAL—1989

Product Line Overview • Datasheets • Designing with PLDs • PLD Design Methodology • PLD Design Development Tools
Fabrication of Programmable Logic • Application Examples

REAL TIME CLOCK HANDBOOK—1989

Real Time Clocks and Timer Clock Peripherals • Application Notes

RELIABILITY HANDBOOK—1986

Reliability and the Die • Internal Construction • Finished Package • MIL-STD-883 • MIL-M-38510
The Specification Development Process • Reliability and the Hybrid Device • VLSI/VHSIC Devices
Radiation Environment • Electrostatic Discharge • Discrete Device • Standardization
Quality Assurance and Reliability Engineering • Reliability and Documentation • Commercial Grade Device
European Reliability Programs • Reliability and the Cost of Semiconductor Ownership
Reliability Testing at National Semiconductor • The Total Military/Aerospace Standardization Program
883B/RETS™ Products • MILS/RETSTM Products • 883/RETS™ Hybrids • MIL-M-38510 Class B Products
Radiation Hardened Technology • Wafer Fabrication • Semiconductor Assembly and Packaging
Semiconductor Packages • Glossary of Terms • Key Government Agencies • AN/ Numbers and Acronyms
Bibliography • MIL-M-38510 and DESC Drawing Cross Listing

SERIES 32000 MICROPROCESSORS DATABOOK—1988

Series 32000 Overview • Central Processing Units • Slave Processors • Peripherals • Board Level Products
Development Systems and Tools • Software Support • Application Notes • NSC800 Family

TELECOMMUNICATIONS—1987

Line Card Components • Integrated Services Digital Network Components • Modems
Analog Telephone Components • Application Notes



National Semiconductor
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

SALES OFFICES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA

Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F., Mexico
Tel: 52-5-524-9402

National Semicondutores Do Brasil Ltda.

Av. Bng. Faria Lima, 1383
6.0 Andor-Conj. 62
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Fax: (55/11) 211-1181 NSBR BR

National Semiconductor GmbH

Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: (0-81-41) 103-0
Telex: 527-649

National Semiconductor (UK) Ltd.

The Maple, Kembrey Park
Swindon, Wiltshire SN2 6UT
United Kingdom
Tel: (07-93) 61-41-41
Telex: 444-674
Fax: (07-93) 69-75-22

National Semiconductor Benelux

Vorstlaan 100
B-1170 Brussels
Belgium
Tel: (02) 6-61-06-80
Telex: 61007

National Semiconductor (UK) Ltd.

Ringager 4A, 3
DK-2605 Brøndby
Denmark
Tel: (02) 43-32-11
Telex: 15-179
Fax: (02) 43-31-11

National Semiconductor S.A.

Centre d'Affaires-La Boursidiere
Bâtiment Champagne, B.P. 90
Route Nationale 186
F-92357 Le Plessis Robinson
France
Tel: (1) 40-94-88-88
Telex: 631065
Fax: (1) 40-94-88-11

National Semiconductor (UK) Ltd.

Unit 2A
Clonskeagh Square
Clonskeagh Road
Dublin 14
Tel: (01) 69-55-89
Telex: 91047
Fax: (01) 69-55-89

National Semiconductor S.p.A.

Strada 7, Palazzo R/3
20089 Rozzano
Milanofiori
Italy
Tel: (02) 8242046/7/8/9

National Semiconductor (UK) Ltd.

P.O. Box 29
N-1321 Stabekk
Norway
Tel: (2) 12-53-70
Fax: (2) 12-53-75

National Semiconductor AB

Box 2016
Stensatrvagen 13
S-12702 Skarholmen
Sweden
Tel: (08) 970190
Telex: 10731

National Semiconductor

Calle Agustin de Foxa, 27
28036 Madrid
Spain
Tel: (01) 733-2958
Telex: 46133

National Semiconductor Switzerland

Alte Winterthurerstrasse 53
Postfach 567
Ch-8304 Wallisellen-Zurich
Switzerland
Tel: (01) 830-2727
Telex: 828-444

National Semiconductor

Kauppakartanonkatu 7 A22
SF-00930 Helsinki
Finland
Tel: (90) 33-80-33
Telex: 126116

National Semiconductor

Postbus 90
1380 AB Weesp
The Netherlands
Tel: (0-29-40) 3-04-48
Telex: 10-956
Fax: (0-29-40) 3-04-30

National Semiconductor Japan Ltd.

Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-ku
Tokyo 160 Japan
Tel: 3-299-7001
Fax: 3-299-7000

National Semiconductor

Hong Kong Ltd.
Suite 513, 5th Floor,
Chinachem Golden Plaza,
77 Mody Road, Tsimshatsui East,
Kowloon, Hong Kong
Tel: 3-7231290
Telex: 52996 NSSEA HX
Fax: 3-3112536

National Semiconductor (Australia) PTY, Ltd.

1st Floor, 441 St. Kilda Rd.
Melbourne, 3004
Victoria, Australia
Tel: (03) 267-5000
Fax: 61-3-2677458

National Semiconductor (PTE), Ltd.

200 Cantonment Road 13-01
Southpoint
Singapore 0208
Tel: 2252226
Telex: RS 33877

National Semiconductor (Far East) Ltd.

Taiwan Branch
P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (86) 02-501-7227
Telex: 22837 NSTW
Cable: NSTW TAIPEI

National Semiconductor (Far East) Ltd.

Korea Branch
13th Floor, Dai Han Life Insurance
63 Building,
60, Yoido-dong, Youngdeungpo-ku,
Seoul, Korea 150-763
Tel: (02) 784-8051/3, 785-0696/8
Telex: 24942 NSPKLO
Fax: (02) 784-8054