NSC800 MICROPROCESSOR FAMILY HANDBOOK

NATIONAL SEMICONDUCTOR

NSC800 MICROPROCESSOR FAMILY HANDBOOK

NATIONAL SEMICONDUCTOR

Preface

This manual contains detailed design-related information pertaining to the National Semiconductor Corporation 800-Family Series of high-performance/low-power microprocessor components. Detailed information is provided for all components in the family, hardware functions, software operations and system design considerations.

The information contained in this manual is accurate at the time of publication, but is subject to change without notice.

Additional copies of this manual, or other National Semiconductor publications, may be obtained from the National Semiconductor sales office or distributor serving your locality.

2900 Semiconductor Drive
Santa Clara, CA. 95051
(408) 737-5000/ TWX (910) 339-9240

Table of Contents

Page

Chapter 1 INTRODUCTION

1.1 PURPOSE OF THE MANUAL 1-1
1.2 NOTES ON THIS EDITION 1-1
1.3 ADDITIONAL INFORMATION 1-1
1.4 MANUAL OUTLINE 1-1

Chapter 2 MICROPROCESSOR SYSTEMS

2.1 GENERAL CAPABILITIES 2-1
2.2 PERFORMANCE SPECTRUM 2-1

Chapter 3 NSC800 FAMILY

3.1 HIGH-PERFORMANCE/LOW-POWER CONCEPT 3-1
3.1.1 CMOS Development 3-1
3.1.2 NMOS Development 3-1
3.1.3 P2CMOS Development 3-1
3.2 APPLICATIONS 3-2
3.2.1 Power Consumption 3-2
3.2.2 Power Supply 3-2
3.2.3 Electrical Noise 3-2
3.2.4 Temperature 3-2
3.3 FAMILY MEMBERS 3-2
3.3.1 NSC800 CPU 3-3
3.3.2 NSC810 RAM-I/O-Timer 3-3
3.3.3 NSC830 ROM-I/O 3-3
3.4 RELATED COMPONENTS 3-3
3.4.1 Memory 3-4
3.4.2 Logic 3-4
3.4.3 Data Acquisition 3-4

Chapter 4 NSC800 HARDWARE DESCRIPTION

4.1 INTRODUCTION 4-1
4.2 ARCHITECTURE 4-1
4.3 CPU WORKING AND ALTERNATE REGISTER SETS 4-1
4.3.1 CPU Working Registers 4-1
4.3.2 Alternate Registers 4-1
4.4 REGISTER FUNCTIONS 4-1
4.4.1 Accumulator (A Register) 4-1
4.4.2 F Register - Flags 4-1
4.4.2.1 Carry (C) 4-3
4.4.2.2 Adds/Subtract (N) 4-3
4.4.2.3 Parity/Overflow (P/V) 4-3
4.4.2.4 Half Carry (H) 4-4
4.4.2.5 Zero Flag (Z) 4-4
4.4.2.6 Sign Flag (S) 4-4
4.4.3 Additional General-Purpose Registers 4-4
4.4.4 Alternate Configurations 4-4
4.5 DEDICATED REGISTERS 4-5
4.5.1 Index Register (IX and IY) 4-5
4.5.2 Interrupt Register (I) 4-5
4.5.3 Refresh Register (R) 4-5
4.5.4 Stack Pointer (SP) 4-5
4.5.5 Program Counter (PC) 4-5
4.6 CPU FUNCTIONS 4-6
4.6.1 Reset 4-6
4.6.2 Timing 4-7

Table of Contents (Cont'd.)

Page

4.6.3 Address/Data Bus 4-8
4.6.4 Bus Control 4-10
4.6.5 Wait 4-10
4.6.6 Interrupts 4-10
4.6.6.1 Non-Maskable Interrupt ($\overline{NMI}$) 4-10
4.6.6.2 Maskable Interrupts ($\overline{RSTA}$, $\overline{RSTB}$, $\overline{RSTC}$) 4-16
4.6.6.3 Multi-Mode Interrupt ($\overline{INTR}$) 4-16
4.6.6.4 INS8080A Compatible - Mode 0 4-17
4.6.6.5 Restart Input - Mode 1 4-17
4.6.6.6 Indirect Call - Mode 2 4-17
4.6.6.7 Interrupt Enable, Interrupt Disable 4-17
4.6.6.8 Interrupt Control Register 4-18
4.6.7 Advanced Cycle Status 4-21
4.6.8 Refresh 4-21
4.6.9 Power-Save 4-22
4.7 POWER CONSUMPTION 4-23

Chapter 5 NSC800 SOFTWARE/PROGRAMMING DESCRIPTION

5.1 INTRODUCTION 5-1
5.2 ADDRESSING MODES 5-1
5.2.1 Register 5-1
5.2.2 Implied 5-1
5.2.3 Immediate 5-1
5.2.4 Immediate Extended 5-2
5.2.5 Direct Addressing 5-2
5.2.6 Register Indirect 5-2
5.2.7 Indexed 5-2
5.2.8 Relative 5-3
5.2.9 Modified Page Zero 5-3
5.2.10 Bit 5-3
5.3 INSTRUCTION SET 5-3
5.3.1 8-Bit Loads 5-5
5.3.1.1 Register To Register 5-5
5.3.1.2 Register To Memory 5-6
5.3.1.3 Memory To Register 5-6
5.3.2 16-Bit Loads 5-7
5.3.2.1 Register To Register 5-7
5.3.2.2 Register To Memory 5-8
5.3.2.3 Memory To Register 5-8
5.3.3 8-Bit Arithmetic 5-9
5.3.3.1 Register Addressing Arithmetic 5-9
5.3.3.2 Immediately Addressed Arithmetic 5-12
5.3.3.3 Memory Addressed Arithmetic 5-14
5.3.4 16-Bit Arithmetic 5-16
5.3.5 Bit Set, Reset, and Test 5-18
5.3.5.1 Register 5-18
5.3.5.2 Memory 5-18
5.3.6 Rotate and Shift 5-19
5.3.6.1 Register 5-19
5.3.6.2 Memory 5-21
5.3.6.3 Register/Memory 5-23
5.3.7 Exchanges 5-24
5.3.7.1 Register/Register 5-24
5.3.7.2 Register/Memory 5-24
5.3.8 Memory Block Moves and Searches 5-24
5.3.8.1 Single Operations 5-24
5.3.8.2 Repeat Operations 5-25
5.3.9 Input/Output 5-26

Table of Contents (Cont'd.)

Page

5.3.10 CPU Control 5-29
5.3.11 Program Control Group 5-30
5.3.11.1 Jumps 5-30
5.3.11.2 Calls 5-31
5.3.11.3 Returns 5-31
5.3.11.4 Restarts 5-32
5.4 COMPARISON TO INS8080A 5-32
5.4.1 Data Transfer Group 5-32
5.4.2 Arithmetic Group 5-33
5.4.3 Logical Group 5-33
5.4.4 Branch Group 5-34
5.4.5 Stack, I/O, and Machine Control Group 5-35
5.4.6 Flag Manipulation 5-35

Chapter 6 NSC810 RAM-I/O-TIMER

6.1 INTRODUCTION 6-1
6.1.1 Architecture 6-1
6.1.2 RAM-I/O-Timer Functions 6-1
6-2 NSC810 INTERFACE 6-1
6.2.1 Reset 6-2
6.2.2 Chip Enable (CE) 6-2
6.2.3 Address Latch Enable (ALE) 6-3
6.2.4 Input/Output Timer or RAM Select (IOT/M) 6-3
6.2.5 Read Signal (RD) 6-3
6.2.6 Write Signal (WR) 6-3
6.3 INPUT/OUTPUT (I/O) Functions 6-3
6.3.1 Standard I/O Operation 6-3
6.3.1.1 Port Addressing 6-3
6.3.1.2 Data Direction 6-3
6.3.1.3 Bit Set/Bit Clear 6-3
6.3.1.4 Mode Definition Register (MDR) 6-4
6.3.1.5 Port Functions - Basic I/O 6-4
6.3.2 Strobed Mode of Operation 6-4
6.3.3 Handshake Signals 6-5
6.3.3.1 Interrupt/PC0 6-6
6.3.3.2 Buffer Full/PC1 6-6
6.3.3.3 Strobed/PC2 6-6
6.3.4 Strobed Input 6-6
6.3.5 Strobed Output 6-6
6.3.6 Strobed Output TRI-STATE™ Mode (Mode 3) 6-7
6.4 TIMER 6-7
6.4.1 Timer Control Signals 6-7
6.4.1.1 Timer Input (TIN) 6-7
6.4.1.2 Timer Gate (TG) 6-7
6.4.2 Timer Modes 6-8
6.4.2.1 Event Counter (Mode 1, TMR Bits = 001) 6-8
6.4.2.2 Accumulative Timer (Mode 2, TMR Bits = 010) 6-8
6.4.2.3 Restartable Timer (Mode 3, TMR Bits = 011) 6-9
6.4.2.4 One Shot (Mode 4, TMR Bits = 100) 6-9
6.4.2.5 Square Wave (Mode 5, TMR Bits = 101) 6-9
6.4.2.6 Pulse Generator (Mode 6, TMR Bits = 110) 6-9
6.4.3 Timer Prescaler 6-10
6.4.4 Timer Read/Write Mode 6-10
6.4.5 Gate Input Polarity 6-10
6.4.6 Output Polarity 6-10
6.5 PROGRAMMING 6-10
6.6 NSC810 RAM MEMORY 6-11

Table of Contents (Cont'd.)

Page

Chapter 7 NSC830 ROM-I/O

7.1 INTRODUCTION 7-1
7.1.1 Architecture 7-1
7.1.2 ROM-I/O Functions 7-2
7.2 INTERFACE 7-2
7.2.1 Reset Signal 7-2
7.2.2 Chip Enable (CE_0, $\overline{IOR}$, $\overline{CE_1}$) 7-3
7.2.3 Address Latch Enable (ALE) 7-3
7.2.4 Input/Output or ROM Select (IO/$\overline{M}$) 7-3
7.2.5 Read Signal ($\overline{RD}$) 7-3
7.2.6 Write Signal ($\overline{WR}$) 7-3
7.3 MEMORY 7-3
7.4 INPUT/OUTPUT (I/O) FUNCTIONS 7-3
7.4.1 Standard I/O Operation 7-3
7.4.1.1 Port Addressing 7-3
7.4.1.2 Data Direction 7-3
7.4.1.3 Bit Set/Bit Clear 7-4
7.4.1.4 Mode Definition Register (MDR) 7-4
7.4.1.5 Port Functions - Basic I/O 7-4
7.4.2 Strobed Mode of Operation 7-5
7.4.2.1 Handshake Signals 7-5
7.4.2.2 Interrupt/PC0 7-5
7.4.2.3 Buffer Full/PC1 7-5
7.4.2.4 Strobed/PC2 7-6
7.4.3 Strobed Input 7-6
7.4.4 Strobed Output 7-6
7.4.5 Strobed Output TRI-STATE Mode (Mode 3) 7-6
7.5 PROGRAMMING 7-6

Chapter 8 HARDWARE SUPPORT AND SYSTEM DESIGN

8.1 INTRODUCTION 8-1
8.2 NSC800 ARCHITECTURE - OVERVIEW 8-1
8.3 SYSTEM TIMING 8-1
8.3.1 Clock Generation 8-1
8.3.2 Control Strobes 8-1
8.3.3 System Initialization 8-3
8.4 POWER SUPPLY CONSIDERATIONS 8-3
8.5 SYSTEM BUS STRUCTURE 8-4
8.6 BUFFERING 8-4
8.7 INTERFACING 8-6
8.8 ADDRESS DECODING 8-6
8.9 MEMORY INTERFACING 8-7
8.10 I/O INTERFACING 8-8
8.10.1 Standard I/O Mapped Interface 8-8
8.10.2 Memory Mapped I/O 8-8
8.10.3 Memory and Peripheral I/O Cycle Extend 8-8
8.11 INTERFACING TO OTHER FAMILIES 8-11
8.12 APPLICATIONS OF THE NSC800 FAMILY 8-11
8.12.1 Expanded NSC800 System 8-11
8.12.2 Serial I/O Using Port Bits of NSC830 8-11
8.12.3 Emulating The NSC830 8-12
8.12.4 Data Acquisition System 8-13

Chapter 9 DEVELOPMENT SUPPORT

9.1 INTRODUCTION 9-1
9.2 PUBLICATIONS 9-1

Table of Contents (Cont'd.)

Page

9.3 TRAINING 9-1
9.4 TECHNICAL SUPPORT PROGRAMS 9-1
9.5 STARPLEX™ DEVELOPMENT SYSTEM 9-1
9.6 IN-SYSTEM EMULATOR 9-1

Chapter 10 GENERAL INFORMATION

10.1 INTRODUCTION 10-1
10.2 NSC830 ORDERING INFORMATION 10-1
10.3 PACKAGING AND PROTECTION 10-1
10.4 PHYSICAL DIMENSIONS 10-1

Appendix A

Page

NSC800 High-Performance Low-Power Microprocessor A-3
NSC810 RAM-I/O-Timer A-25
NSC830 ROM-I/O; NSC831 I/O Only A-40
MM74PC00 Quad 2-Input NAND Gate A-52
MM74PC02 Quad 2-Input NOR Gate
MM74PC04 Hex Inverter
MM74PC08 Quad 2-Input AND Gate
MM74PC32 Quad 2-Input OR Gate
MM74PC74 Dual D Flip-Flop A-57
MM74PC138 3-Line To 8-Line Decoder/Demultiplexer A-63
MM82PC08 8-Bit Bidirectional Transceiver A-70
MM82PC12 8-Bit Input/Output Port A-77
μB3 Comparison Study NSC800 vs. 8085, Z80® A-87
μB4 Software Comparison NSC800 vs. 8085, Z80® A-91

List Of Illustrations

Figure	Title	Page
2-1	National Semiconductor Microprocessors Performance Spectrum	2-1
3-1	P²CMOS Cross-Section	3-1
3-2	Minimum NSC800 System	3-2
4-1	NSC800 CPU Functional Block Diagram	4-2
4-2	CPU Working and Alternate Registers	4-1
4-3	Flag Register	4-3
4-4	Dedicated Registers	4-5
4-5	Pin Configuration	4-7
4-6	NSC800 RESET Function	4-7
4-7	Power-On RESET Timing	4-7
4-8	NSC800 Timing Function	4-8
4-9	Timing Control Configurations	
	A. External Clock Input	4-8
	B. RC Network Timing	4-8
	C. Crystal Timing	4-8
	D. Clock Timing Waveforms	4-8
4-10	Typical Frequency vs. Operating Voltage	4-9
4-11	Basic Timing Cycles	4-9

List of Illustrations (Cont'd.)

Figure	Title	Page
4-12	NSC800 Bus Structure	4-9
4-13	Demultiplexed Address Data Bus Configuration	4-10
4-14A	Opcode Fetch Cycle	4-11
4-14B	Memory Read and Write Cycle	4-11
4-15	NSC800 Bus Control Signals	4-12
4-16	Bus Acknowledge Cycle	4-12
4-17	NSC800 $\overline{WAIT}$ Function	4'12
4-18	Effect of $\overline{WAIT}$ on Machine Cycles	4-13
	A. Opcode Fetch Cycles Without $\overline{WAIT}$ States	4-13
	B. Opcode Fetch Cycles With $\overline{WAIT}$ States	4-13
	C. Memory Read/Write Cycles Without $\overline{WAIT}$ States	4-14
	D. Memory Read/Write Cycles With $\overline{WAIT}$ States	4-14
	E. Input/Output Cycles Without $\overline{WAIT}$ States	4-15
	F. Input/Output Cycles With $\overline{WAIT}$ States	4-15
4-19	NSC800 Interrupt Functions	4-16
4-20	Mode 0 $\overline{INTR}/\overline{INTA}$ Timing	4-18
4-21	Mode 2 Interrupt Timing	4-19
4-22	Mode 2 Interrupt Handling Addressing	4-20
4-23	IFF_1 and IFF_2 Operation	4-20
4-24	Interrupt Control Register	4-21
4-25	NSC800 Advanced Cycle Status Function	4-21
4-26	Advanced Cycle State Codes	4-21
4-27	NSC800 Refresh Function	4-21
4-28	Refresh Timing	4-22
4-29	Power-Save Function	4-22
4-30	Power-Save Timing	4-23
4-31	Power-Save Implementation of Single Instruction	4-23
6-1	NSC810 Detailed Block Diagram	6-1
6-2	NSC810 Pin Configuration	6-2
6-3	Block Diagram for Port Bit (i)	6-3
6-4	Read from RAM, Port or Timer	6-4
6-5	Write to RAM, Port or Timer	6-5
6-6	NSC810 Port A Block Diagram	6-5
6-7	Strobed Mode Input	6-6
6-8	Strobed Mode Output	6-7
6-9	Timer Internal Block Diagram (One of Two)	6-8
6-10	Mode 1: Counter	6-8
6-11	Mode 2: Timer	6-9
6-12	Mode 3: Timer	6-9
6-13	Mode 4: One Shot	6-9
6-14	Mode 5: Square Wave	6-10
6-15	Mode 6: Pulse Generator	6-10
7-1	NSC830 Block Diagram	7-1
7-2	Pin Configuration	7-2
7-3	Block Diagram for Port Bit (i)	7-4
7-4	Read Cycle Timing	7-4
7-5	Write Cycle Timing	7-5
7-6	NSC830 Port A Block Diagram	7-6
7-7	Strobed Mode Input	7-7
7-8	Strobed Mode Output	7-7
8-1	NSC800 CPU Functional Block Diagram	8-2
8-2	External Clock Circuits	8-3
8-3	NSC800 Initialization	8-3
8-4	Battery Back-Up In Power-Save	8-4
8-5	System Bus Separation	8-5
8-6	Buffering	8-5

List of Illustrations (Cont'd.)

Figure	Title	Page
8-7	Buffering/Demultiplexing	8-6
8-8	Minimum NSC800 System	8-7
8-9	Address Decoding	8-7
8-10	Expanded R/W Memory System	8-9
8-11	Minimum NSC800 System (Memory-Mapped I/O)	8-10
8-12	Single WAIT State Generation	8-10
8-13	Multiple WAIT State Generation	8-10
8-14	P²CMOS at High Voltage to Standard TTL	8-11
8-15	Standard TTL to P²CMOS at Higher Voltage	8-11
8-16	Expanded NSC800 System	8-12
8-17	Serial I/O Via Port I/O Pins	8-13
8-18	NSC830 Emulator	8-14
8-19	Remote Data Acquisition	8-15
9-1	STARPLEX™ Development System	9-2

List of Tables

Table	Title	Page
4-1	Functional Pin Descriptions	4-6
4-2	Access Time Requirements	4-16
5-1	Mnemonic Notation Instruction Set	5-3
5-2	Assembled Object Code Notation	5-4
5-3	Register Addressing Arithmetic	5-12
6-1	NSC810 Pin Descriptions	6-2
6-2	Mode Definition Register Bit Assignments	6-4
6-3	I/O Port and Timer Addressing	6-11
6-4	Timer Mode Selection	6-11
6-5	Timer Programming Selection Examples	6-12
7-1	NSC830 Pin Descriptions	7-2
7-2	Mode Definition Register Bit Assignments	7-4
7-3	I/O Port Addressing	7-8
8-1	Typical Timer Output Frequencies	8-16
10-1	Required Order Information	10-2

Chapter 1

Introduction

1.1 PURPOSE OF THE MANUAL

This manual furnishes the reader with comprehensive information relative to National Semiconductor's new NSC800 Family of high-performance/low-power microprocessor components.

Detailed information is provided for all components in the family. These include hardware functions, software operations, and system design considerations. It is assumed that the reader has a background in electronics and is familiar, in general, with microprocessor systems and terminology.

1.2 NOTES ON THIS EDITION

Effort has been made to publish this manual as early as possible so that information is available to designers doing early implementations of NSC800 based systems. Therefore, this edition contains preliminary and/or partial information in some areas.

1.3 ADDITIONAL INFORMATION

As with all National Semiconductor products, additional information may be obtained from your nearest National Sales Office. In addition, specific application problems can be addressed to local National Field Application Engineers. These field engineers are backed-up by Microprocessor Application Engineers located at National's Santa Clara Offices.

1.4 MANUAL OUTLINE

The manual discusses a number of separate issues relative to the NSC800 Family. Following the initial introduction, general information pertinent to microprocessor systems is discussed in Chapter 2. Particular focus is placed on where the NSC800 Family fits into the entire microprocessor spectrum.

General information relating to the NSC800 Family is found in Chapter 3. Particular family members are described as well as the concept behind family development. Chapters 4 through 7 describe the low-power microprocessor family components in terms both of their hardware functions and programmed operations. Chapter 8 discusses the system aspects of the NSC800 Family devices and other devices that are appropriate for NSC800 system design.

This information is followed by data regarding development systems support, in terms of both software development and hardware prototyping support. The prototype support focuses on emulation techinques and equipment.

Finally, ordering and packaging instructions are supplied in Chapter 10. In addition, specifically related literature is included in the appendices.

Chapter 2

Microprocessor Systems

2.1 GENERAL CAPABILITIES

The NSC800 combines the features of the Intel 8085 and the Zilog Z80 in a new high performance CMOS technology, P²CMOS. Incorporation of a multiplexed address and data bus provides increased functionality on the CPU and support circuits for reduced chip count. The family is available in commercial (0 to 70° C) and industrial (-40 to +85° C) temperature ranges with military temperature components (-55 to +125° C) to be announced.

A minimum system may be configured using the NSC800 with the NSC810 (RAM-I/O-Timer) and NSC830 (ROM-I/O). This system provides the 128 bytes of RAM, 2K bytes of ROM, 42 I/O lines, five interrupt inputs, two timers, and integral clock generation. Typical power dissipation for this three-chip system at 5V and 4MHz is 100mW.

2.2 PERFORMANCE SPECTRUM

National Semiconductor divides microprocessors and associated support devices into three product groups that are defined as: Low-End, Mid-Range and High-End. These divisions are initially defined by areas of application and overall system size (in terms of usable memory, number of peripherals, etc.). Secondly, and harder to quantify, aspects, such as sophistication of instruction and expansion capabilities that could also determine the division. *Figure 2-1* illustrates the division and performance spectrum.

The low-end microprocessor applications are characterized by dedicated general tasks with few, if any, peripheral devices. Performance is usually limited in terms of instruction execution time and the variety of instruction types. These types of microprocessor solutions are extremely effective in applications where low cost and low chip count are of vital importance. If more information is required on these types of microprocessors, the reader is referred to the 70-Series Microprocessor and the 48-Series Microcomputers Handbooks.

In mid-range applications, the task to be performed is expanded considerably both in size and complexity. In addition, the need for more specialized dedicated peripherial circuits is greater because mid-range systems are required to communicate with a variety of I/O interfaces. The NSC800 Family provides full mid-range capability. Due to the high degree of integration, it has the added capability of furnishing highly sophisticated performances to low-end systems as well.

Another mid-range microprocessor family available from National Semiconductor is the Series 8000 Microprocessor Family, which is based on National's INS8080A microprocessor.

The high-end division is typified by a 16-bit microprocessor that is comparable to minicomputers in terms of computational and system performance. These processors can be found in larger systems that require the greatest intelligence capability and flexibility. Information on National's High-End processors can be found in the NS16000 Family of 16-Bit Microprocessors.

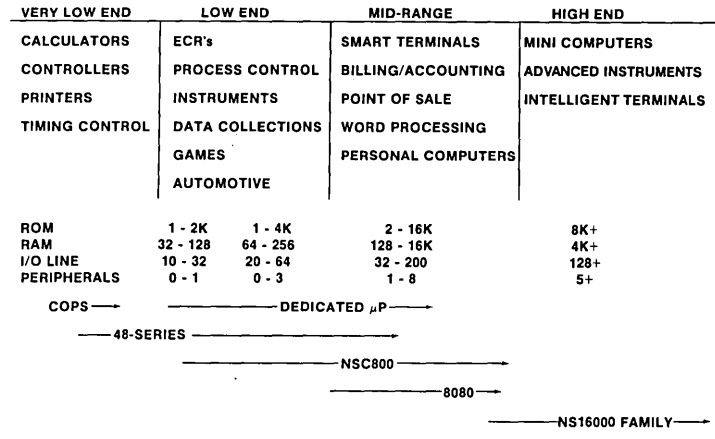


FIGURE 2-1. National Semiconductor Microprocessors Performance Spectrum 72-75

Z80™ Trademark of Zilog Corp.; 8085™ Trademark of Intel Corp.

Chapter 3

NSC800 Family

3.1 HIGH-PERFORMANCE/LOW-POWER CONCEPT

Since the development of integrated circuits, logic designers have had to consider the performance versus power tradeoff when implementing designs. Generally, as speed (one measure of performance) increased, there was the inherent penalty of increased power consumption. The logic designer has a full spectrum of speed versus power devices to select from the 7400 logic devices. Starting with the original 7400TTL family, higher speed (higher power requirements) devices were provided by the 74H or 74S component families; lower speed (with the resultant lower power requirements) devices were provided by the 74L family.

As integrated circuit technologies advanced, the speed versus power tradeoff has been partially relieved. The development of the 74LS low-power Schottky devices provided 74H family speeds combined with the power requirements of the standard 7400 type devices.

3.1.1 CMOS Development

In terms of low-power consumption for integrated circuit devices, the development of CMOS integrated circuits provided designers with a choice, as typified by National's 74C Family of CMOS logic components. However, CMOS devices could not achieve the performance (speed) of 74L types of devices, though the power consumption was lower. In addition, the 74C Family required a tradeoff in the area of functional capabilities. Die size, as compared to circuit density, set a limit to the functions a CMOS device could perform.

3.1.2 NMOS Development

The development of the N-channel MOS process provided the highest level of functional capabilities. As typified by many of National's MOS-LSI and Series 8000 Microprocessor devices, NMOS provides sophisticated devices with a high level of functional capabilities such as the INS8080A microprocessor and its related peripheral devices. However, in terms of power consumption, the traditional tradeoff of high-power requirements still exist.

3.1.3 P²CMOS Development

Additional advances in integrated circuit process technology provides a method to improve the speed versus power performance of products in the MOS spectrum. National's newly developed P²CMOS process approaches the functional capabilities of NMOS combined with the lower power requirements of CMOS devices. This combination of high-performance with low-power is the primary concept behind the development of the NSC800 Microprocessor Family. *Figure 3-1* illustrates a cross-section of the P²CMOS process.

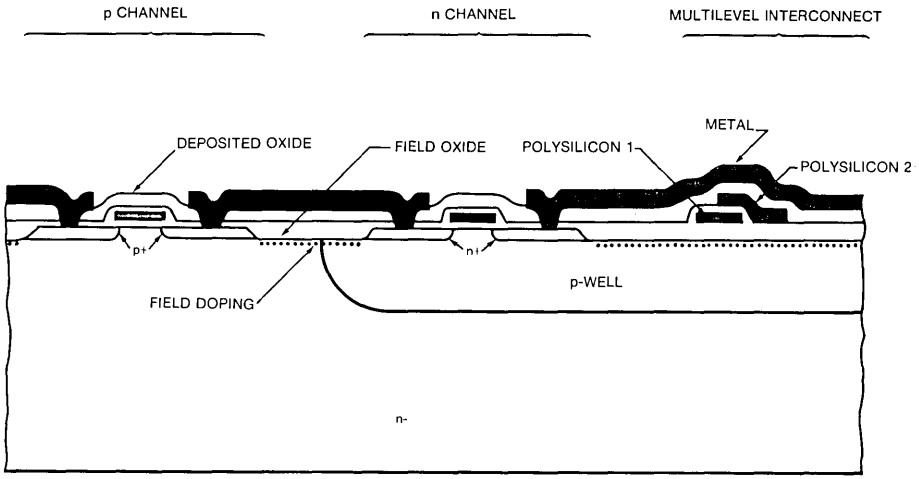


FIGURE 3-1. P²CMOS Cross-Section

3.2 APPLICATIONS

Since the NSC800 Microprocessor Family has the functionality of the most advanced 8-bit, commercially available, mid-range microprocessors, they satisfy all the application areas where these processors can be used (See *Figure 2-1*). Of particular interest are the expanded capabilities brought to these applications by the CMOS nature of the NSC800 Family.

3.2.1 Power Consumption

The most obvious CMOS characteristic shared by the P²CMOS NSC800 Family is the very low power consumption requirements. On an average, P²CMOS implementation of functions will require 90 to 95% less power than a comparable NMOS device. This dramatic power savings makes the NSC800 a natural choice for those microprocessor applications that require battery operation.

3.2.2 Power Supply

As previously mentioned, the size of the power supply can be reduced since less wattage is required for systems operation. The NSC800 Family's advantage of lower power consumption provides the ability to reduce systems power requirements. In systems that can not be battery operated, but have excessive power demands as compared to the system's capacity for heat dissipation, or because of the absolute size limitation of the given system power supply, the NSC800 may offer the solution to either, or both problems.

Like standard CMOS technology, P²CMOS contains the feature of operating with a wide range of input operating voltages while sustaining full functional capabilities. This feature translates into the possibility of using very loosely regulated power supplies for NSC800 Family based systems. This too reduces power supply requirements in terms of size, regulation, complexity, and therefore, cost.

3.2.3 Electrical Noise

A capability that goes hand in hand with the NSC800 Family's capacity to operate within a wider range of supply voltages is the tolerances for input signal levels, that is, noise immunity. With voltage levels determined as a percentage of supply voltage, systems that operate in severe electrical environments, with large amounts of noise, are prime candidates for NSC800 Family based systems.

3.2.4 Temperature

While NSC800 based systems operate particularly well in severe electrical environments, they work equally well in other types of rugged environments, specifically the environments where systems operation is required over wider or extended temperature ranges. As with most CMOS devices, the NSC800 Family devices are specified to operate over the full military operating range, from -55°C to +125°C. This enables the NSC800 Family to bring advanced microprocessor performance to many more applications, for example, those in process control, remote instrumentation and the like.

In addition to operating well in extended temperature applications, the NSC800 devices generate very little heat because of lower power dissipation. This fact helps alleviate other temperature related problems in the system as a whole, reducing, for example, the mechanical packaging problems for providing sufficient cooling.

3.3 FAMILY MEMBERS

Presently three key members of the of the NSC800 Family are defined. The three devices, the NSC800, NSC810 and NSC830 combine to provide a three-chip minimum system, as shown in *Figure 3-2*.

Each basic device is briefly described in this chapter. Detailed information for each device is provided in subsequent chapters.

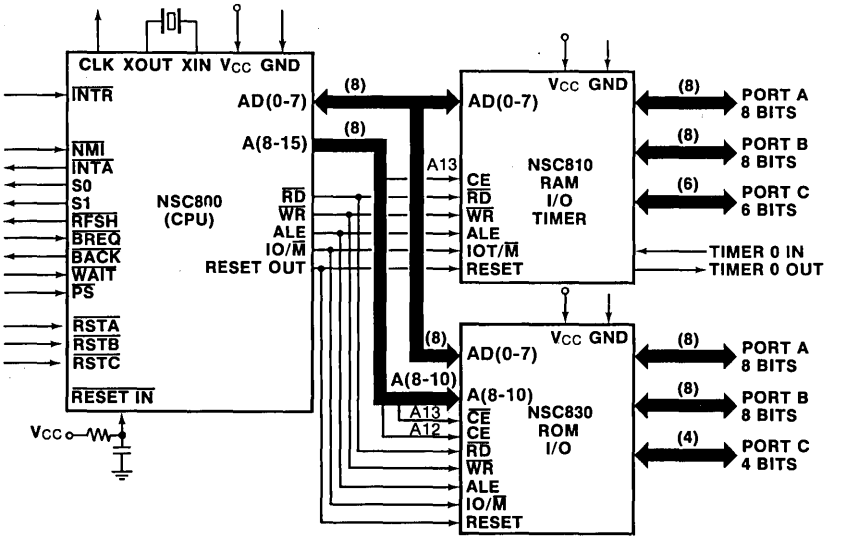


FIGURE 3-2. Minimum NSC800 System

72-9

3.3.1 NSC800 CPU

The NSC800 CPU is the heart of the NSC800 Family. This 8-bit CPU is fabricated with National's P²CMOS process providing it with the speed and functions of available NMOS CPU's while consuming only 5% of the power. The NSC800 operates with a single 5V power supply. Availability of extended voltage versions will be Announced.

The external architecture of the NSC800 provides for a very high degree of integration. With a multiplexed address/data bus, the NSC800 is also able to provide on-chip clock generation, extensive on-chip interrupt capability, and dynamic memory refresh control in addition to many other hardware features. Refer to paragraph 4.2 for external architecture details.

Internally, a multiplicity of programmable registers, a sophisticated set of addressing modes and a large instruction set make the NSC800 a very powerful tool from a software, as well as hardware perspective.

Other system capabilities include provisions for direct memory access, multiprocessing and operation in power save move.

3.3.2 NSC810 RAM-I/O-Timer

The NSC810 RAM-I/O-Timer is a dedicated memory peripheral that combines many system level functions into a single device.

In addition to 128 bytes of random access memory (RAM), the NSC810 provides a total of 22 separately programmable input/output signals that are partitioned into two 8-bit ports plus one 6-bit port. This I/O capability is extremely flexible in that it is completely programmable in defining modes of operation for each I/O port and each bit therein.

One port, Port A, is also usable in strobed modes of operation in both input and output configurations. In these strobed modes, additional handshake signals are used to interface with other external devices. These handshake signals are additional functions performed by signals in Port C (the 6-bit port). The system designer has the option to tradeoff these bits as regular I/O functions or as strobed I/O control functions.

The third function in the NSC810 is as a timer. Two 16-bit, fully programmable timers are contained in the device. Each timer can operate in any of six programmable modes. Like the strobed modes of Port A operation, full use of the two timers involves the use of dual function pins in Port C. Detailed operational descriptions can be found in Chapter 6.

3.3.3 NSC830 ROM-I/O

The NSC830 ROM-I/O dedicated memory peripheral combines system level functions in a single device. The NSC830 provides 2K (2048) bytes of mask programmable read-only-memory (ROM). In addition to the ROM, the NSC830 includes 20 separately programmable Input/Output (I/O) signal lines partitioned into two, 8-bit ports plus one, 4-bit port. The I/O capability is designed for the greatest flexibility in allowing complete programmability in defining modes of operation for each I/O port and each bit therein.

Like the NSC810, Port A, on the NSC830 can operate in one of three strobed modes. When operating in these modes, additional handshake signals are required for interfacing to external devices. These handshake signals are available on the 4-bit (Port C) port as a tradeoff of I/O functions. Complete details on the NSC830 operation can be found in chapter 7.

3.4 RELATED COMPONENTS

The three basic devices described in section 3.3 are sufficient to implement many small to medium sized systems without any additional circuitry. However, to implement larger systems, use standard memories, or to allow more flexible interfacing to external devices, other types of circuitry are required.

Certainly the circuits required are available in standard logic families and are represented by such functions as NAND, NOR, AND, OR, inverter gates, D-type flip-flops, decoders, octal latches, etc. At National, these circuits are available in a number of TTL families (7400, 74LS, 74S, 74H, 74L). In addition, for low power types of applications, these functions are also available in CMOS implementation, i.e., the 74C family.

While the 74C family provides a multiplicity of functions from which the system designer can choose, in some applications the speeds available may not be sufficient for straightforward interfacing to an NSC800 based system. One option available to the designer is the use of a TTL functional equivalent, most likely a 74LS implementation. Of course, here the tradeoff is in terms of the power required for the auxiliary function. In an effort to alleviate this type of tradeoff difficulty, a number of standard support circuits of the types discussed have also been implemented in the P²CMOS process to bring the high-performance/low-power benefits to a broader spectrum of devices.

Refer to the data books from other groups within National for more complete information, specifically the MEMORY, CMOS, and DATA ACQUISITION manuals. Contact your local National sales office for such publications and informative literature.

3.4.1 Memory

The following memory components are applicable to NSC800 designs in that they supply low power memory.

a. NMC6504 4K x 1 RAM
b. NMC6514 1K x 4 RAM
c. NMC27C16 2K x 8 EPROM
d. NMC67C16 2K x 8 EPROM (latched version)

3.4.2 Logic

The following logic components have been implemented in National's P²CMOS process specifically for support of the NSC800 High-Performance/Low-Power Microprocessor Family. See the CMOS databook for a more complete listing of other CMOS devices.

a.	MM82PC12	8-bit I/O
b.	MM74PC138	1 of 8 decoder
c.	MM82PC08	8-bit driver
d.	MM74PC00	Quad Nand Gate
e.	MM74PC02	Quad Nor Gate
f.	MM74PC04	Hex Inverter
g.	MM74PC08	Quad And Gate
h.	MM74PC32	Quad Or Gate
i.	MM74PC74	Dual D Flip Flop

3.4.3 Data Acquisition

The following devices are some of those devices that will provide a low power CMOS data acquisition function. See the Data Acquisition handbook for complete detailed information.

a.	ADC3511	3.5 Digit Integrating A/D Converter
b.	ADC3711	3.75 Digit Integrating A/D Converter
c.	ADC0816/17	16-Channel, 8-Bit A/D Converter
d.	ADC0808/09	8-Channel, 8-Bit A/D Converter

Chapter 4

NSC800 Hardware Description

4.1 INTRODUCTION

This chapter reviews the architecture in terms of the register array and the flag status that reflects the status of the CPU. In paragraph 4.3, each external function will be explained, as well as the resulting internal effect. The chapter focuses on the functional aspects of the CPU from a hardware perspective, including details in terms of timing.

4.2 ARCHITECTURE

As illustrated in *Figure 4-1*, the NSC800 is an 8-bit parallel device. The major functional areas are: the ALU, register array, interrupt control, timing and control logic. These areas are interconnected via the internal 8-bit data bus.

4.3 CPU WORKING AND ALTERNATE REGISTER SETS

4.3.1 CPU Working Registers

The portion of the register array shown in *Figure 4-2* represents the CPU working registers. These sixteen 8-bit registers are considered general-purpose registers because they perform a multitude of functions, depending on the instruction being executed. They are also grouped together because of the manipulative types of instruction that they perform, particularly alternate set operations.

The F (flag) register could be considered a special-purpose register because its contents are less a result of program choice than of machine status. The F register is included because of its interaction with the A register, and its manipulations in the alternate register set operations.

4.3.2 Alternate Registers

The NSC800 registers designated as CPU working registers have one common feature: the existence of a duplicate register in an alternate register set. This architectural concept simplifies programming during operations such as interrupt response, when the machine status represented by the contents of the registers must be saved.

CPU Main Working Register Set

Accumulator A	(8)	Flags F	(8)
Register B	(8)	Register C	(8)
Register D	(8)	Register E	(8)
Register H	(8)	Register L	(8)

CPU Alternate Working Register Set

Accumulator A'	(8)	Flags F'	(8)
Register B'	(8)	Register C'	(8)
Register D'	(8)	Register E'	(8)
Register H'	(8)	Register L'	(8)

FIGURE 4-2. CPU Working and Alternate Registers

The alternate register concept makes one set of registers available to the programmer at any given time. With two instructions (EX AF,A'F' and EXX), the contents of the current working set of registers can be exchanged with their respective alternate set. One performs the exchange between the A and F registers and their respective duplicates (A' and F'). This exchange allows the saving of the primary status information contained in the accumulator and the flag register. The second exchange instruction performs the exchange between the remaining registers, B, C, D, E, H, and L, and their respective alternates B', C', D', E', H', and L'. This essentially saves the contents of the original register while providing the programmer with a usable register.

4.4 REGISTER FUNCTIONS

4.4.1 Accumulator (A Register)

The A register serves as a source or destination register for data manipulation instructions. In addition, it serves as the accumulator for the results of 8-bit arithmetic and logic operations.

The A register also has a special status in some types of operations; that is, certain addressing modes are reserved for the A register only, although the function is available for all the other registers. For example, any register can be loaded by immediate, register indirect, or indexed source addressing modes. The A register, however, can also be loaded with an additional register indirect capability or by extended direct source addressing.

Another special feature of the A register is that it produces more efficient memory coding than equivalent instruction functions directed to other registers. Any register can be rotated; however, while it requires a two-byte instruction to normally rotate any register, a single-byte instruction is available for rotating the contents of the accumulator (A register).

A study of the instruction set in paragraph 5.2 will indicate these and other features of the A register.

4.4.2 F Register - Flags

The NSC800 flag register actually consists of six status flags that contain information regarding the results of previous CPU operations. It is not a register in the sense that the program does not have a direct access to read or write the contents. The program can read the contents of

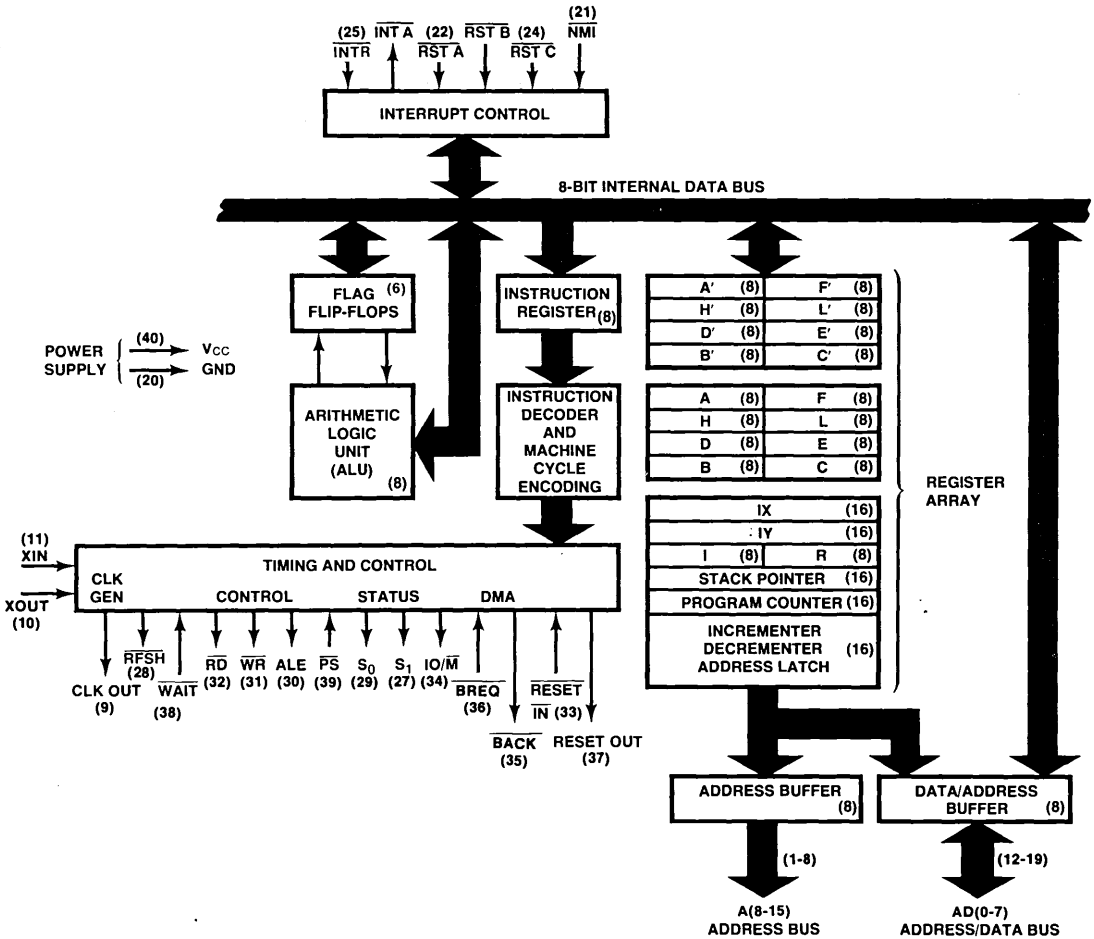


FIGURE 4-1. NSC800 CPU Functional Block Diagram

72-1

the flag register by first pushing the contents onto the stack, and then accessing the contents in memory. It is included here because of its affiliation with the accumulator and the existence of a duplicate register for use in exchange instructions with the accumulator.

Of the six flags shown in *Figure 4-3*, only four can be directly tested by the programmer via conditional jump, call, and return instructions. They are the Sign (S), Zero (Z), Parity/Overflow (P/V), and Carry (C) flags. The Half Carry (H) and Add/Subtract (N) flags are used for internal operations related to BCD arithmetic.

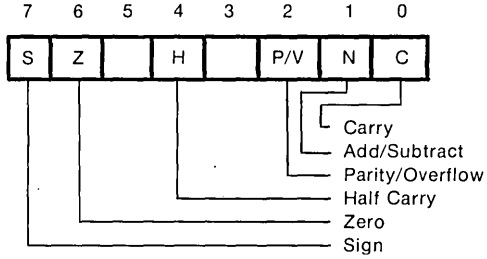


FIGURE 4-3. Flag Register

4.4.2.1 Carry (C)

This flag is set by the carry from the highest order bit of the accumulator during an add instruction, or a borrow generated during a subtraction instruction. Specific shift and rotate instructions also affect this bit.

Two specific instructions in the NSC800 instruction repertoire are included to set, or to complement the carry flag.

Other operations that affect the C flag are as follows:

- Adds
- Subtracts
- Logic Operations (always resets C flag)
- Rotate Accumulator
- Rotate and Shifts
- Decimal Adjust
- Negation of Accumulator

Other operations have no effect on the C flag.

4.4.2.2 Adds/Subtract (N)

This flag is used in conjunction with the H flag to ensure that the proper BCD correction algorithm is used during the decimal adjust instruction (DAA). The correction algorithm depends on whether an add or subtract was previously done with BCD operands.

The operations that set the N flag are:

- Subtractions
- Decrements (8-bit)
- Complementing of the Accumulator
- Block I/O
- Block Searches
- Negation of the Accumulator

The operations that reset the N flag are:

- Adds
- Increments
- Logic Operations
- Rotates
- Set and Complement Carry
- Input Register Indirect
- Block Transfers
- Load of the I or R Registers
- Bit Tests

Other operations do not affect the N flag.

4.4.2.3. Parity/Overflow (P/V)

The Parity/Overflow flag is a dual-purpose flag that indicates results of logic and arithmetic operations. In logic operations, the P/V flag indicates the parity of the result; the flag is set (high) if the result is even, reset (low) if the result is odd. In arithmetic operations, it represents an overflow condition when the result, interpreted as signed two's complement arithmetic, is out of range for the eight-bit accumulator (i.e. -128 to +127).

Two special instructions, load of the I or R register into the A register, allow the contents of the interrupt flag (IFF_2) to be copied into this bit for test purposes.

The following operations affect the P/V flag according to the parity of the result of the operation:

- Logic Operations
- Rotate and Shift
- Rotate Digits
- Decimal Adjust
- Input Register Indirect

The following operations affect the P/V flag according to the overflow result of the operation.

- Adds (16 bit with carry, 8-bit with/without carry)
- Subtracts (16 bit with carry, 8-bit with/without carry)
- Increments and Decrements
- Negation of Accumulator

The following operations have an indeterminate effect on the P/V flag:

- Block I/O
- Bit Tests

In block transfers and compares, the P/V flag indicates the status of the BC register, always ending in the reset state after an auto repeat of a block move. Other operations leave the P/V flag unaffected.

4.4.2.4 Half Carry (H)

This flag indicates a BCD carry, or borrow, result from the low-order four bits of operation. It can be used to correct the results of a previously packed decimal add, or subtract, operation by use of the Decimal Adjust Instruction (DAA).

The H flag will be affected according to the results of the following instructions:

- Adds (8-bit)
- Subtracts (8-bit)
- Increments and Decrements
- Decimal Adjust
- Negation of Accumulator

It is also affected by the following operations:

- Always Set by: Logic AND
  Complement Accumulator
  Bit Testing

- Always Reset By: Logic OR's and XOR's
  Rotates and Shifts
  Set Carry
  Input Register Indirect
  Block Transfers
  Loads of I and R Registers

The H flag is left in an indeterminate state by the following operations:

- 16-bit Adds with/without carry
- 16-Bit Subtracts with carry
- Complement of the carry
- Block I/O
- Block Searches

Other operations do not affect the H flag.

4.4.2.5 Zero Flag (Z)

The zero flag is set when a zero is loaded into the accumulator, when the zero is the result of an operation. Otherwise the zero flag bit remains cleared, or is unaffected.

The zero flag is affected by the following operations according to the result:

- Adds (16-bit with carry, 8-bit with/without carry)
- Subtracts (16-bit with carry, 8-bit with/without carry)
- Logic Operations
- Increments and Decrements
- Rotate and Shifts
- Rotate Digits
- Decimal Adjust
- Input Register Indirect
- Block I/O (always set after auto repeat block I/O)
- Block Searches
- Load of I and R Registers
- Bit Tests
- Negation of Accumulator

The Z flag is left in an indeterminate state after the following operation:

- Block Transfers

The zero flag is unaffected by other operations.

4.4.2.6 Sign Flag (S)

The sign flag stores the state of bit 7 (the most-significant bit and sign bit) of the accumulator following an arithmetic operation. This flag is of use when dealing with signed numbers.

The sign flag is affected by the following operation according to the result:

- Adds (16-bit with carry, 8-bit with/without carry)
- Subtracts (16-bit with carry, 8-bit with/without carry)
- Logic Operations
- Increments and Decrements
- Rotate and Shifts
- Rotate Digits
- Decimal Adjust
- Input Register Indirect
- Block Search
- Load of I and R Registers
- Negation of Accumulator

The sign flag is left in an indeterminate state after the following operations:

- Block I/O
- Block Transfers
- Bit Tests

The sign flag is unaffected by other operations.

4.4.3 Additional General-Purpose Registers

The other general-purpose registers are the B, C, D, E, H and L registers and their alternate register set, B', C', D', E', H' and L'. These registers can be used interchangeably as true, general-purpose registers for a large number of data manipulation instructions.

In addition, the B and C registers can perform special functions in the NSC800 expanded I/O capabilities, particularly block I/O operations. In these functions, the C register can be used in register indirect addressing modes to address I/O ports; the B register can be used in a counter function. The B register is also used in conjunction with a special conditional jump instruction (DJNZ) to again serve this counting function.

Further details on the register usage and the special application can be found in Chapter 5.

4.4.4 Alternate Configurations

The six 8-bit general-purpose registers can also be used in an alternate configuration: three 16-bit registers. This is accomplished by the concatenation of the B and C

registers to form the BC register, the D and E registers form the DE register, and the H and L registers form the HL register.

Having these 16-bit registers allows 16-bit data handling, thereby expanding the number of 16-bit registers available for memory addressing modes. The HL register is widely used for code efficient use of register indirect addressing of the memory.

The DE register provides a second memory pointer register for the NSC800's powerful block transfer operations. The BC register also provides an assist to the block transfer operations by acting as a byte-counter for these operations.

A further explanation of the addressing modes and use of the 16-bit registers is provided in Chapter 5.

4.5 DEDICATED REGISTERS

There are 6 dedicated registers in the NSC800: two 8-bit and four 16-bit registers (see *Figure 4-4*).

Although their contents are under program control, the program has no control over their operational functions, as is the case with the CPU working registers. The function of each dedicated register is described as follows:

CPU Dedicated Registers

Index Register IX	(16)
Index Register IY	(16)
Interrupt Vector Register I	(8)
Memory Refresh Register R	(8)
Stack Pointer SP	(16)
Program Counter PC	(16)

FIGURE 4-4. Dedicated Registers

4.5.1 Index Register (IX and IY)

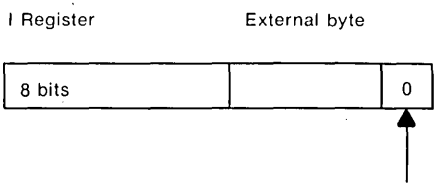
The NSC800 contains two index registers to hold independent, 16-bit base addresses used in indexed addressing modes. In this mode, an index register, either IX or IY, is used as a base to point to an area in memory that is being addressed.

In all instructions employing indexed modes of operation, another byte is included that acts as a signed two's complement displacement. This addressing mode enables easy data table manipulations.

More information on the addressing mode can be found in section 5.2.

4.5.2 Interrupt Register (I)

When the NSC800 is operating with Mode 2 type interrupts enabled (via program control), the response to an interrupt is an indirect call to any memory location. The pointer to this memory location is formed from the data in two consecutive memory locations; the first is always an even location. The even location 16-bit address is formed from the internal I register (8 bits) and an external byte (8 bits) supplied by the interrupting device in the following manner:



The even location contains the low-order byte, the next consecutive location contains the high-order byte of the pointer to the beginning address of the interrupt service routine.

For another description of this operation see 4.6.6.7.

4.5.3 Refresh Register (R)

For systems that use dynamic memories rather than static RAM's, the NSC800 provides an integral 8-bit memory refresh counter. After each opcode fetch, the contents are incremented and are made available to the system, along with a refresh control signal. This provides a totally transparent refresh cycle.

In terms of programming, the R register can be read and written, although this is usually done for test purposes. Other details on the refresh timing can be found in section 4.6.9.

4.5.4 Stack Pointer (SP)

The 16-bit stack pointer contains the address of the current top of stack that is located in external system RAM. The stack is organized in a last-in, first-out (LIFO) type of structure, with the pointer being decremented when data is pushed onto the stack, and incremented when data is popped from the stack.

As detailed in paragraph 5.2, various operations can be used to store, or retrieve, data on the stack. This, along with the usage for subroutine calls and interrupts, allows simple implementation of subroutine and interrupt nesting as well as alleviating many problems of data manipulation.

4.5.5 Program Counter (PC)

The PC is the most basic control register in a programmable processor. The 16-bit PC in the NSC800 contains the address of the next instruction to be fetched for execution. It is automatically incremented to affect program flow.

There are many conditional and unconditional jumps, calls, and return instructions in the NSC800's instruction repertoire that enable easy manipulation of this register in controlling the program execution. They are detailed in paragraph 5.2.

4.6 CPU FUNCTIONS

The following paragraphs detail the several CPU functions as they appear to the external system. By understanding these functions, the system designer can design efficient systems to meet the requirements of his particular application.

The overall NSC800 CPU pinout and functional pin description is shown in *Figure 4-5* and *Table 4-1.*

4.6.1 Reset

The NSC800 Reset function (see *Figure 4-6*) is designed for easy implementation. By including a Schmitt trigger input on the $\overline{\text{RESET IN}}$ pin, a simple RC network accomplishes the power-on reset function. The timing required for the power-on reset is shown in *Figure 4-7*, as is the timing if a logic signal is used to reinitialize during system operation.

TABLE 4-1. Functional Pin Descriptions

OUTPUT SIGNALS

Bus Acknowledge ($\overline{\text{BACK}}$): Active low. $\overline{\text{BACK}}$ indicates to the bus requesting device that the CPU bus and its control signals are in the TRI-STATE™ mode. The requesting device may then take control of the bus and its control signals.

Address Bits 8-15 [A(8-15)]: Active high. These are the most significant 8 bits of the memory address bus, or of the input/output address. During a $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ cycle, the A(8-15) bus is in the TRI-STATE mode.

Reset Out (RESET OUT): Active high. When RESET OUT is high, it indicates the CPU is being reset. The signal is normally used to reset the peripheral devices.

Input/Output/Memory (IO/$\overline{\text{M}}$): An active high on the IO/$\overline{\text{M}}$ output signifies that the current machine cycle is relative to an input/output device. An active low on the IO/$\overline{\text{M}}$ output signifies that the current machine cycle is relative to memory. It is TRI-STATE during $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ cycles.

Refresh ($\overline{\text{RFSH}}$): Active low. The refresh output indicates that the dynamic RAM refresh cycle is in progress. $\overline{\text{RFSH}}$ goes low during T3 and T4 states of all M1 cycles.

Address Latch Enable (ALE): ALE is active only during the T1 state of M cycles and T3 state of M1 cycles. The high to low transition of ALE indicates that a valid memory/I-O/refresh address is available on the AD(0-7) lines.

Read Strobe ($\overline{\text{RD}}$): Active low. On the trailing edge of the $\overline{\text{RD}}$ strobe, data are input to the CPU via the AD(0-7) lines. The $\overline{\text{RD}}$ line is in the TRI-STATE mode during $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ cycles.

Write Strobe ($\overline{\text{WR}}$): While the $\overline{\text{WR}}$ line is low, valid data are output by the CPU on the AD(0-7) lines. The $\overline{\text{WR}}$ line is in the TRI-STATE mode during $\overline{\text{BREQ}}$/$\overline{\text{BACK}}$ cycles.

Clock (CLK): CLK is an output provided for use as a system clock. The CLK output is a squarewave at one half of the input frequency.

Interrupt Acknowledge ($\overline{\text{INTA}}$): Active low. The $\overline{\text{INTA}}$ output is activated in the next machine (M) cycle (during T-2 state) following the last M cycle of an instruction (the $\overline{\text{INTR}}$ input is sampled during the last T-state of the last M-cycle of an instruction). The $\overline{\text{INTA}}$ output is normally used to gate the interrupt response vector from the peripheral controller onto the AD(0-7) lines. It is used in two of the three interrupt modes. In Mode 0, an instruction is gated onto the AD(0-7) lines during $\overline{\text{INTA}}$. In Mode 2, a single interrupt response vector is gated onto the data bus.

Status (S0, S1): Bus status outputs indicate encoded information regarding the ensuing M cycle as follows:

S1	S0	STATE
0	0	HALT
0	1	WRITE
1	0	READ
1	1	OPCODE FETCH

INPUT/OUTPUT SIGNALS

POWER (V_{CC}): +3 to +12-volt supply.

Ground (GND): 0-volt reference.

Crystal or R-C (XIN, XOUT): XIN may be used as an external clock input.

Multiplexed Address/Data [AD(0-7)]: Active High.

At $\overline{\text{RD}}$ Time:	Input data to CPU
At $\overline{\text{WR}}$ Time:	Output data from CPU
At Falling Edge of ALE Time:	Least significant byte of address during memory reference cycle. 8-bit port address during I/O reference cycle.
During $\overline{\text{BREQ}}$/ $\overline{\text{BACK}}$ Cycle:	High Impedance

INPUT PROTECTION

All inputs are protected from static charge with diode clamps to both V_{CC} and GND. Normal precautions taken with all MOS devices are recommended.

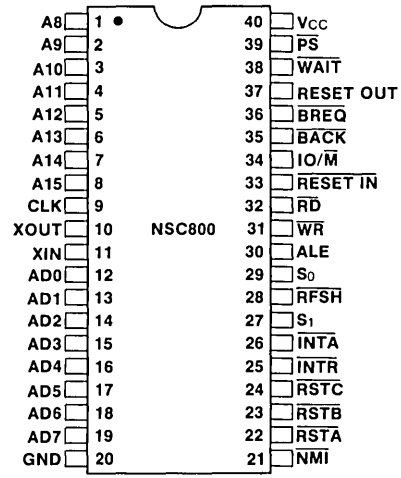


FIGURE 4-5. Pin Configuration

45-6

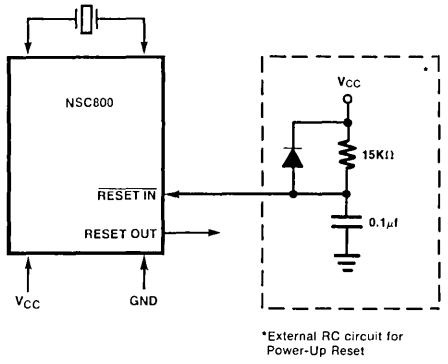


FIGURE 4-6. NSC800 RESET Function

Reset initialization resets the CPU to a known state so that program execution can begin. In particular, this initialization includes:

a. Clear the PC to X'0000 so program execution begins at memory location zero.

b. Clear both the Interrupt (I) and Refresh (R) registers to (X'00).

c. Disable system interrupts.

d. Setting the 4-bit interrupt control register ICR (see 4.6.6) to X'01 masking on $\overline{INTR}$, masking off $\overline{RSTA}$, $\overline{B}$, and $\overline{C}$.

e. Enable Interrupt Mode 0.

Note: *Reset initialization steps d and e sets up the 8080 interrupt modes of operation.*

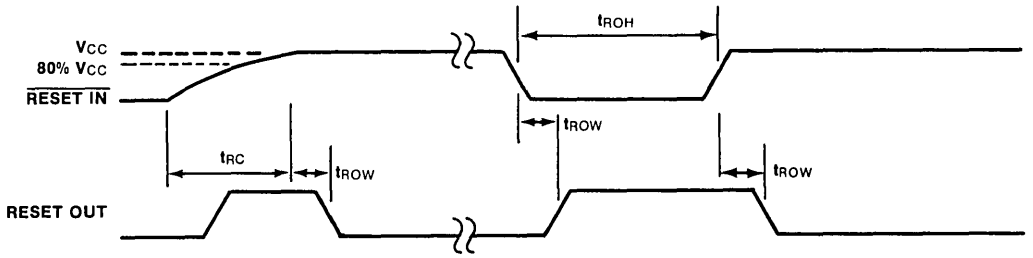
The contents of other registers are undefined.

During the reset time, the address/data bus [A(8-15) and AD(0-7)] are in a high impedance mode and all CPU strobes are inactive.

4.6.2 Timing

The NSC800 CPU provides an on-chip oscillator so only an external timing element, usually a crystal (see *Figure 4-8*), is required. An RC network or external clock may also be used. The clock circuitry divides this external frequency by two and provides a square wave clock signal to the rest of the system at one-half the input frequency. This clock signal, CLK, is the basic timing signal for all CPU operations. *Figure 4-9* shows the various configurations and requirements for the timing inputs to the NSC800.

The maximum clock frequency of the NSC800 is 2.5 MHz at 5v. The faster NSC800A (availability to be announced)is specified at a 4 MHz clock rate at 5v. *Figure 4-10* illustrates the maximum clock frequencies as a function of Vcc.



t = Period of clock output
t_{RC} = RC time constant (min 100ms if XTAL used)
t_{ROW} = Reset Out Wait ≤ t
t_{ROH} = Reset Out Hold ≥ 3t + 50ns

FIGURE 4-7. Power-On RESET Timing

72-79

This CLK frequency defines the basic timing base for all CPU functions. Each CLK cycle defines a 'T-state' for the CPU. Some number of T-states then combine to define a machine cycle. For example, 4 T-states define an opcode fetch machine cycle. Depending on the instruction to be executed, one or more machine cycles will define an instruction. The simplest instructions, a register-to-register transfer, for example, require only one machine cycle. In this case, a 4 MHz clock yields a 250 ns T-state, and thus, a 1 μs minimum instruction time. Paragraph 5.2 details the timing for the various instructions.

Figure 4-11 illustrates the basic timing machine cycles. In some cases, the timing of these machine cycles can be modified, as mentioned in paragraph 4.6.5, by means of the NSC800 $\overline{\text{WAIT}}$ function. The paragraph also discusses the various access time requirements imposed on memory and I/O devices used with the NSC800.

4.6.3 Address/Data Bus

The NSC800 employs a multiplexed address and data bus scheme (see *Figure 4-12*) with the low-order byte of the address being multiplexed with the 8-bit data bus. Thus, a fewer number of valuable pins are used in the CPU package and a higher degree of integration can be achieved for other CPU functions.

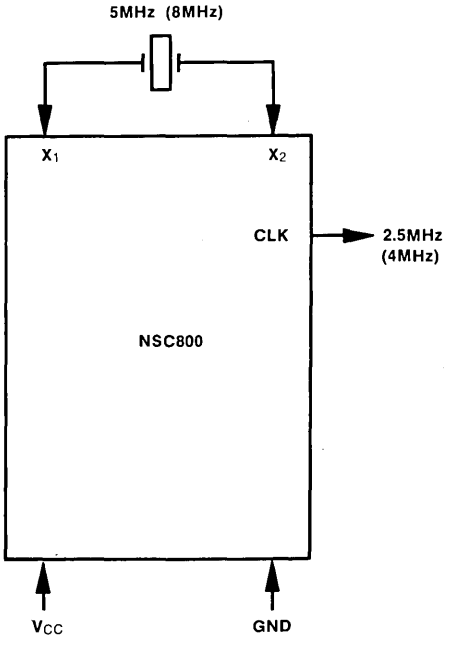


FIGURE 4-8. NSC800 Timing Function 72-80

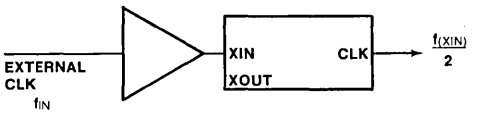
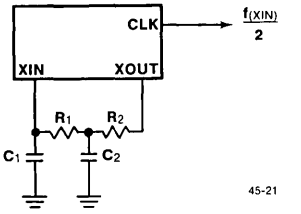
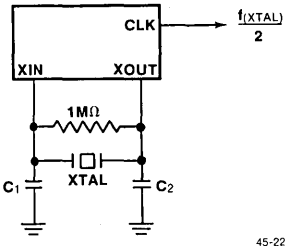


FIGURE 4-9A. External Clock Input 45-20



45-21

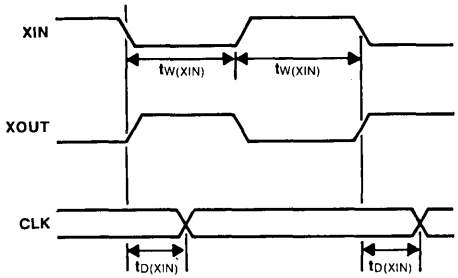
FIGURE 4-9B. RC Network Timing



45-22

FIGURE 4-9C. Crystal Timing

Clock Timing



NSC800A (8MHz) XTAL $0 \leq t_{D(XIN)} \leq 90ns$, $62ns \leq t_{W(XIN)}$

NSC800 (5MHz) $0 \leq t_{D(XIN)} \leq 135ns$, $90ns \leq t_{W(XIN)}$

Output load on clock is 100pf. V_{CC} = 5V

FIGURE 4-9D. Clock Timing Waveforms 72-81

FIGURE 4-9. Timing Control Configurations

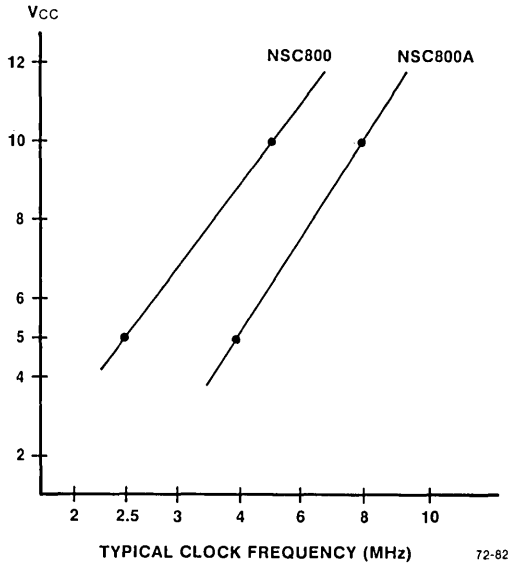


FIGURE 4-10. Typical Frequency vs. Operating Voltage

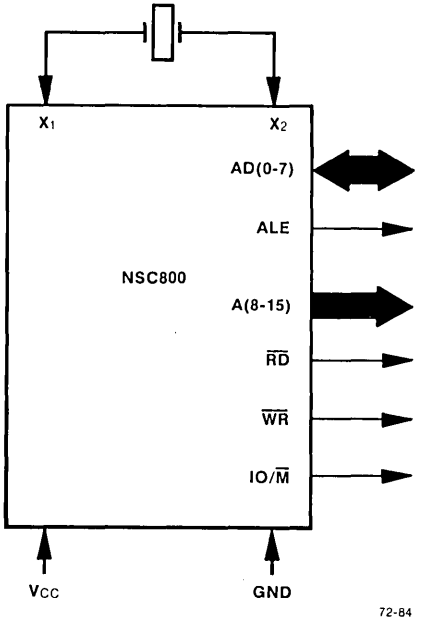


FIGURE 4-12. NSC800 Bus Structure

The key signal in demultiplexing the information on the bus is the Address Latch Enable (ALE) signal. This signal provides gating and strobe information as to when the multiplexed AD(0-7) bus contains address information. The ALE signal is used by peripherals on the bus to strobe the low-order byte of address information that occurs at the beginning of various access cycles. *Figure 4-13* indicates how the ALE signal is used with an octal latch device, MM82PC12, to provide a standard non-multiplexed bus structure.

Three other control signals are involved in a data access via the bus, a generalized read ($\overline{RD}$), or write ($\overline{WR}$) strobe plus a signal that indicates whether the access is to the memory or I/O space ($IO/\overline{M}$).

With a memory addressing capacity of 64K bytes, all 16 bits of address information are used; the I/O capacity of 256 I/O ports only employs 8 bits of address with the I/O

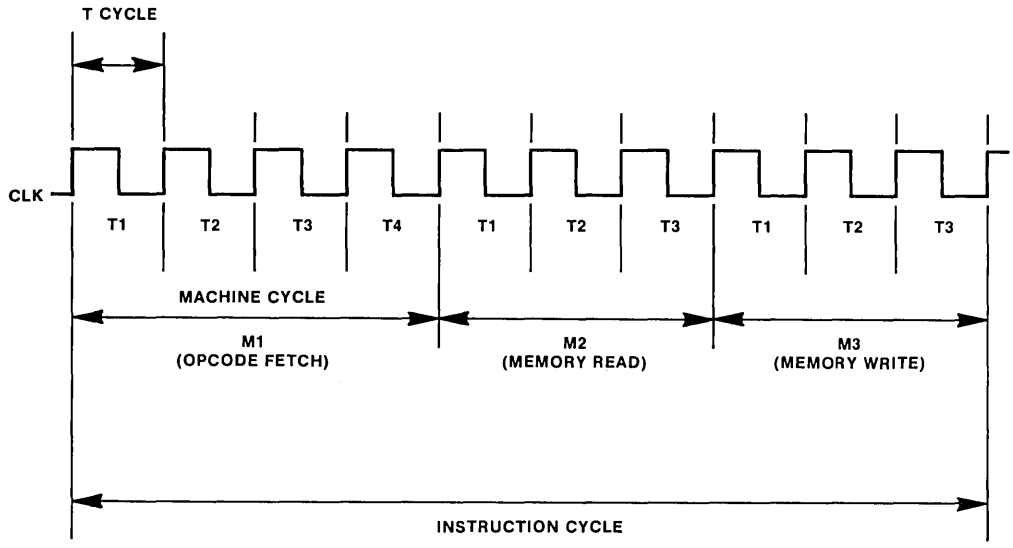
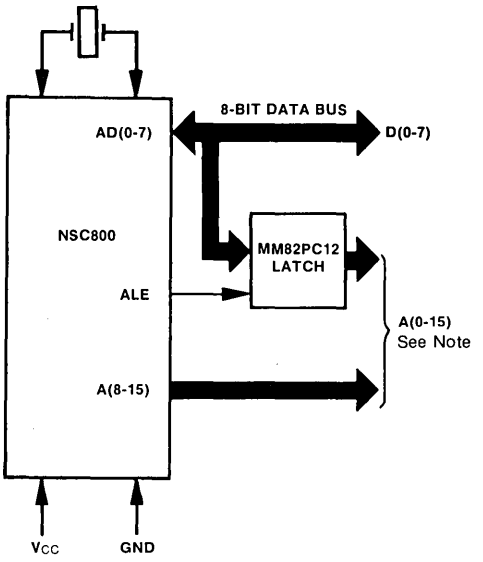


FIGURE 4-11. Basic Timing Cycles



NOTE: 16-Bit Address Bus 72-85

FIGURE 4-13. Demultiplexed Address Data Bus Configuration

address present on both the high-order A(8-15) and low-order AD(0-7) bytes.

Figure 4-14 indicates the timing involved in data accesses over the data bus structure.

4.6.4 Bus Control

There are two general applications that may require the CPU to temporarily relinquish control of the bus to other system components. The first is Direct Memory Access (DMA) cycles where a peripheral device will directly transfer data to or from memory, or to gain throughput, rather than passing each byte through the CPU under program control. The second, somewhat related to DMA, is in multiprocessing applications where a second CPU in the system requires access to the whole, or a portion of I/O or memory space.

For both applications the NSC800 provides a mechanism wherein an external device can make an asynchronous request to the CPU to relinquish control of the bus by means of the Bus Request ($\overline{BREQ}$) signal (see *Figure 4-15*). When the CPU recognizes the request, it will complete execution of the current machine cycle, release the bus by setting the address bus A(8-15), address/data bus AD(0-7), and the control bus ($\overline{RD}$, $\overline{WR}$, and IO/$\overline{M}$) to the high impedance mode. At this time, the NSC800 responds with the Bus Acknowledge ($\overline{BACK}$) handshake signal indicating that it has relinquished bus control. Then, the external device may access any system component on the bus independently. The NSC800 will remain off the bus until the $\overline{BREQ}$ signal is negated.

The timing relationship of the bus control signals are shown in *Figure 4-16*.

4.6.5 Wait

When the timing of an external device (memory or peripheral) does not meet the access requirements of the CPU, the NSC800 $\overline{WAIT}$ function is required, (see *Figure 4-17*). This function allows an external device to extend memory or I/O machine cycles, thus relaxing access time requirements so that slower memories and I/O devices can be used without degrading total system throughput.

The timing relations for the various data bus signals and the effect of the $\overline{WAIT}$ function providing additional T-states ($\overline{WAIT}$ States) during an access cycle are shown in *Figure 4-18. Figures 4-18A and 4-18B* show opcode fetch cycles without and with $\overline{WAIT}$ states. *Figures 4-18C and 4-18D* show Memory Read/Write cycles without and with $\overline{WAIT}$ states. *Figures 4-18E and 4-18F* show Input/Output cycles without and with $\overline{WAIT}$ states. *Table 4-2* lists the access time requirements of the various memory and I/O operations. Note that I/O operations have inserted an automatic (self-generated) wait state.

4.6.6 Interrupts

The NSC800 interrupt function is designed for a high level of integration. By providing five priority levels for interrupt inputs, the CPU will automatically resolve any priority conflicts without external components. One interrupt input can be programmed to operate in any of three separate operational modes. This gives the system designer and programmer flexibility for determining the complexity, level of sophistication, and capability to be implemented in the NSC800 interrupt structure, (see *Figure 4-19*).

This section describes each type of interrupt, as well as the general control and masking capabilities available in the NSC800. The interrupt functions discussed are:

- Non-Maskable Interrupt
- Maskable Restart Interrupt
- Multi-Mode Interrupt
- Interrupt Enable
- Interrupt Control Register

4.6.6.1 Non-Maskable Interrupt ($\overline{NMI}$). The non-maskable interrupt is an edge-sensitive interrupt input that causes a direct restart to a dedicated memory address, location X'0066. There are no programmable actions possible to ignore an interrupt request made on this input.

The action of the restart operation is to recognize the interrupt at the end of the current instruction, decrement the SP register, save the program counter on the external memory stack, and begin execution at the designated location, X'0066.

The $\overline{NMI}$ input has the highest interrupt priority, thus $\overline{NMI}$ is usually reserved for emergency type interrupts such as: power failures, illegal operation traps, etc.

For timing diagram see NSC800 Data Sheet in appendix A.

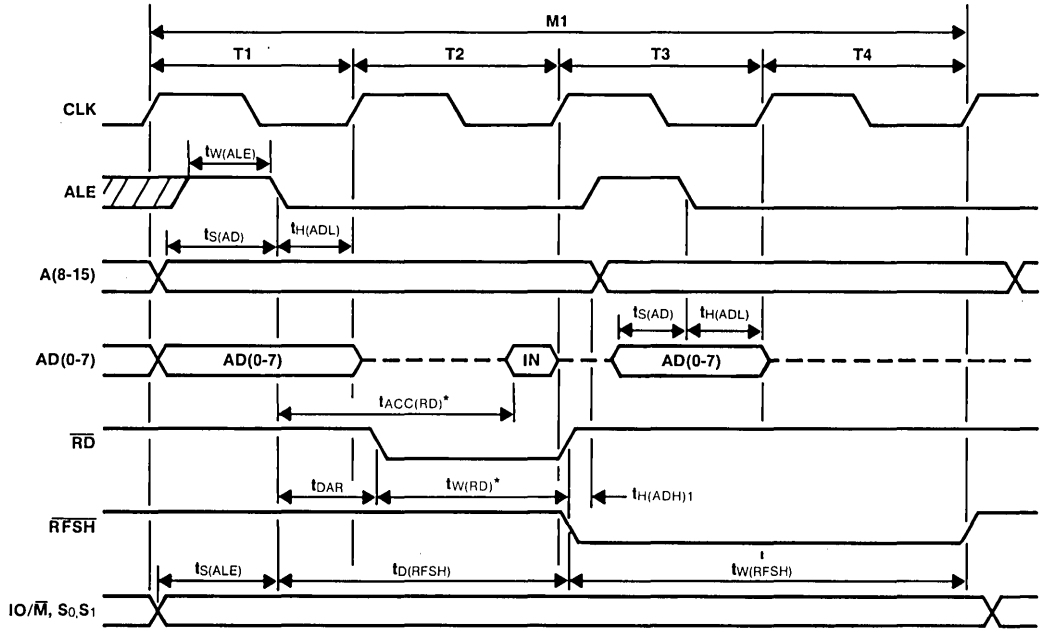
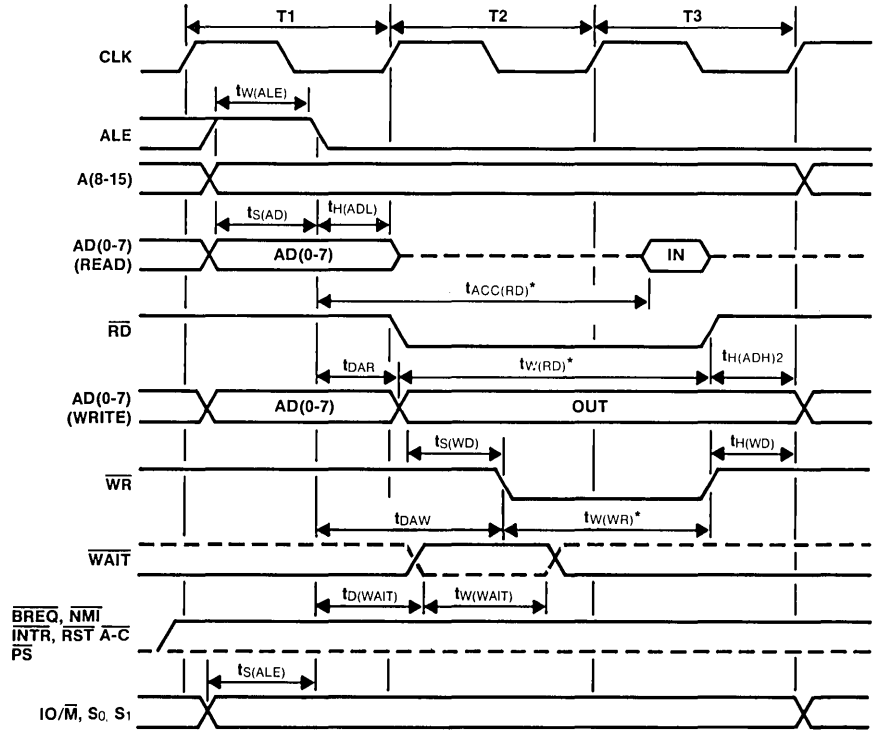


FIGURE 4-14A. Opcode Fetch Cycle

45-18



***NOTES:** 1. Output capacitive loading = 100pf
2. Add t for each WAIT state

FIGURE 4-14B. Memory Read and Write Cycle

45-19

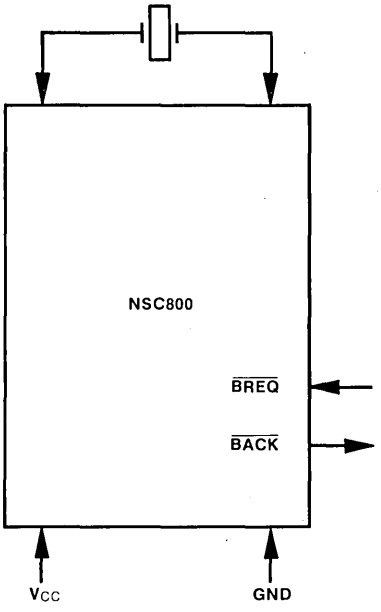


FIGURE 4-15. NSC800 Bus Control Signals

72-86

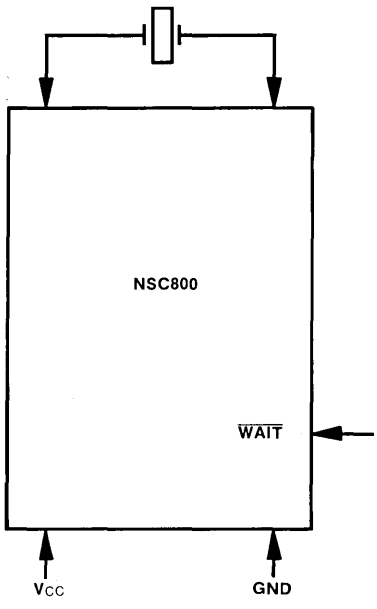
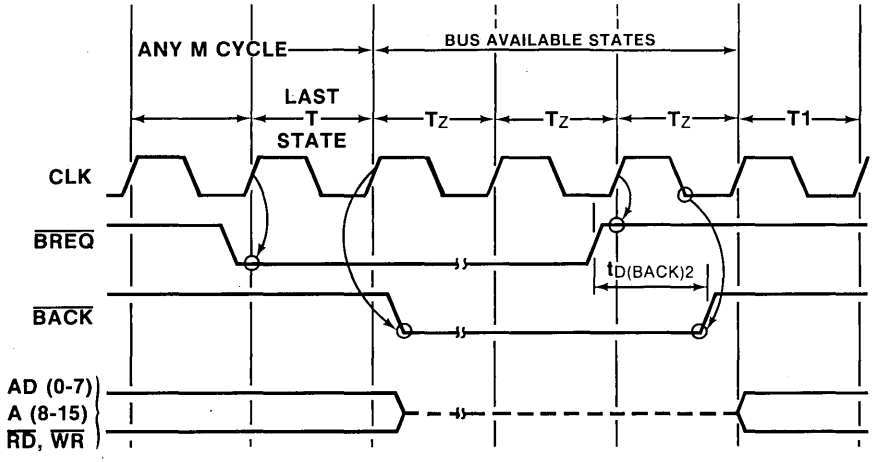


FIGURE 4-17. NSC800 $\overline{\text{WAIT}}$ Function

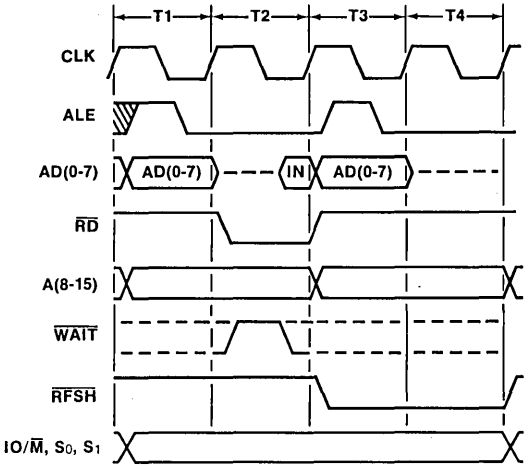
72-87



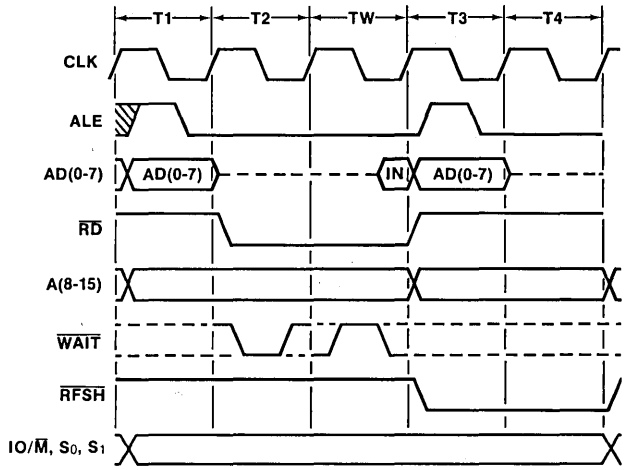
TZ = Times that bus and control signals are in high impedance mode.

45-10

FIGURE 4-16. Bus Acknowledge Cycle

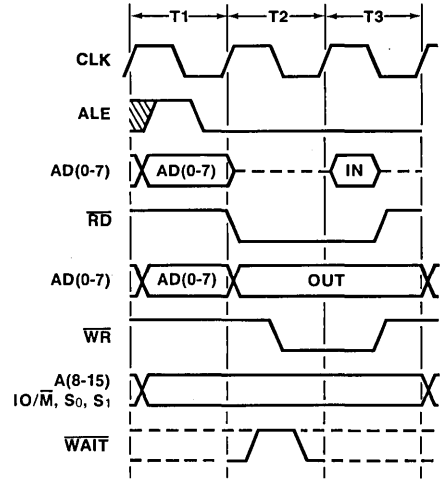


A. Opcode Fetch Cycles Without $\overline{\text{WAIT}}$ States 45-9



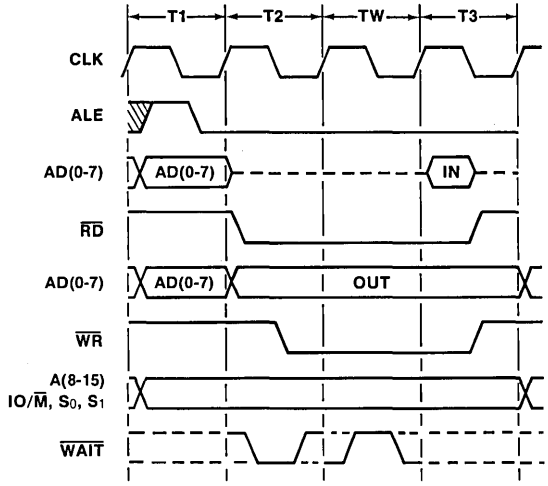
B. Opcode Fetch Cycles With $\overline{\text{WAIT}}$ States 45-14

FIGURE 4-18. Effect of $\overline{\text{WAIT}}$ on Machine Cycles



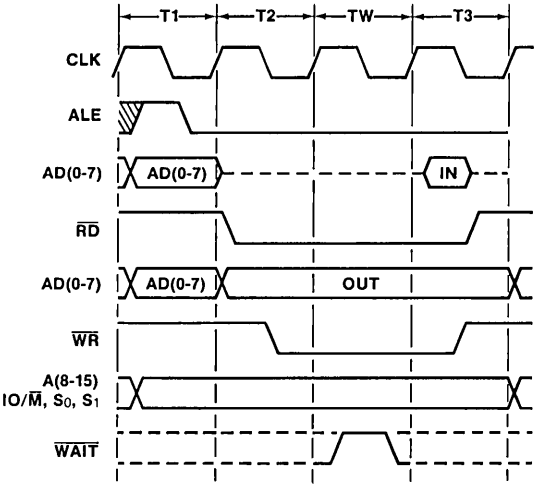
C. Memory Read/Write Cycles Without $\overline{\text{WAIT}}$ States

45-15

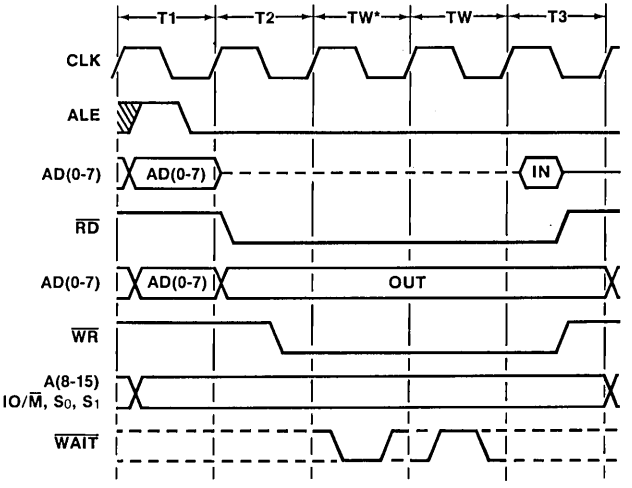


D. Memory Read/Write Cycles With $\overline{\text{WAIT}}$ States 45-7

FIGURE 4-18. Effect of $\overline{\text{WAIT}}$ on Machine Cycles (Cont'd.)



E. Input/Output Cycles Without $\overline{\text{WAIT}}$ States 45-8



*$\overline{\text{WAIT}}$ State User Generated

F. Input/Output Cycles With $\overline{\text{WAIT}}$ States 45-11

FIGURE 4-18. Effect of $\overline{\text{WAIT}}$ on Machine Cycles (Cont'd.)

TABLE 4-2. Access Time Requirements

SYMBOL		PARAMETER	OPCODE FETCH	MEM RD	MEM WR	I/O RD	I/O WR
$t_{ACC(RD)}$	$\frac{3t}{2}$ -125	Read Access Time	475 250	675 375		1075 625	
$t_{W(RD)}$	t -50	Read Strobe Width during Opcode Fetch	350 200	550 325		950 575	
t_{DAR}	$\frac{t}{2}$ -50	Falling Edge of ALE to leading edge of Read Strobe	150 75	150 75		150 75	
t_{DAW}	t -100	Falling Edge of ALE to Leading Edge of WR			300 150		300 150
$t_{W(WR)}$	t -50	Write Strobe Width			350 200		750 450
$t_{S(WD)}$	$\frac{t}{2}$ -75	Write Data Setup Time			125 25		125 25

NOTES:

1. NSC800 = 400ns NSC800
   NSC800A = 250ns NSC800A
2. All times specified in nanoseconds

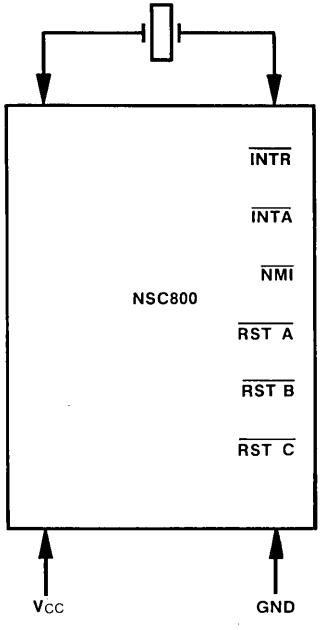


FIGURE 4-19. NSC800 Interrupt Functions

72-88

4.6.6.2 Maskable Interrupts ($\overline{RSTA}$, $\overline{RSTB}$, $\overline{RSTC}$). The NSC800 provides three maskable restart type interrupts designated $\overline{RSTA}$, $\overline{RSTB}$ and $\overline{RSTC}$. These interrupt inputs function similarly to the $\overline{NMI}$ input except for the following:

- The Maskable Interrupts are level sensitive
- The Maskable Interrupts are maskable under program control (see paragraphs 4.6.6.7 and 4.6.6.8)
- Each Maskable Interrupt has a unique dedicated program restart address. The dedicated restart addresses are:

$\overline{RST\ A}$	X'003C
$\overline{RST\ B}$	X'0034
$\overline{RST\ C}$	X'002C

4.6.6.3 Multi-Mode Interrupt ($\overline{INTR}$). The $\overline{INTR}$ input can be operated in three distinct programmable modes: compatible with National's INS8080A microprocessor, equivalent to a restart interrupt, and a unique mixture of hardware and software to allow an indirect call to be executed.

Like the maskable restarts, this interrupt input can be masked on, or off, and is controlled by system interrupt enables, as described in paragraphs 4.6.6.7 and 4.6.6.8.

4.6.6.4 INS8080A Compatible - Mode 0. In mode 0, the NSC800 $\overline{INTR}$ input is similar to the interrupt structure of the INS8080A. If interrupts are enabled, the CPU recognizes the $\overline{INTR}$ input at the end of the current instruction. Rather than executing the next instruction, the CPU responds with the Interrupt Acknowledge ($\overline{INTA}$) signal. The $\overline{INTA}$ is used in place of the $\overline{RD}$ strobe to gate the instruction from the peripheral into the CPU. Using this handshake signal, the interrupting peripheral responds by placing an instruction on the data bus. In effect, the interrupt acknowledge cycle is analogous to an opcode fetch cycle where the interrupting peripheral generates the opcode.

A series of one-byte restart instructions (see paragraph 5.2) are available in the NSC800 instruction repertoire. The restart instructions employ eight dedicated restart locations for interrupt routines through this type of interrupt sequence. The dedicated restart locations, along with the opcode required during $\overline{INTA}$ time, is shown below. The corresponding locations of the maskable restart locations are also listed.

Interrupt	Opcode/Instruction (Mode 0 INTA Responses)	Restart Address
$\overline{INTR}$	B'11000111	X'0000
$\overline{INTR}$	B'11001111	X'0008
$\overline{INTR}$	B'11010111	X'0010
$\overline{INTR}$	B'11011111	X'0018
$\overline{INTR}$	B'11100111	X'0020
$\overline{INTR}$	B'11101111	X'0028
$\overline{RSTA}$	N/A	X'003C
$\overline{INTR}$	B'11110111	X'0030
$\overline{RSTB}$	N/A	X'0034
$\overline{INTR}$	B'11111111	X'0038
$\overline{RSTC}$	N/A	X'002C

These restarts all provide for the saving of the PC on the stack. Another common response to the $\overline{INTA}$ signal is a three-byte CALL sequence, whereby the program address of the interrupt handling subroutine is supplied by the interrupting peripheral. The CALL instruction provides for the saving of the current PC on the stack and execution of code starting at the address contained in the second and third bytes of the CALL instruction.

Figure 4-20 shows the relative timing of the $\overline{INTR}$/$\overline{INTA}$ handshake sequence.

Note that although the single byte restart instructions and CALL instruction are the most commonly used responses to $\overline{INTA}$ in Mode 0, any instruction in the NSC800 instruction repertoire, as detailed in paragraph 5.2, can be used with sufficient $\overline{INTA}$ cycles generated.

4.6.6.5 Restart Input - Mode 1. When operating in Mode 0, or Mode 2, of interrupt,it is clear that additional hardware is required to ensure the proper data is strobed onto the data bus at the proper time (see *Figure 4-20*) . In those systems where chip count is important and the added flexibility is not necessary, the Mode 1 operation allows the $\overline{INTR}$ input to be converted to a fifth restart input. In this mode the dedicated restart address is X'0038, the same as the highest restart instruction.

Note that in this mode, as in Mode 0 and Mode 2, two additional T-states are added to the instruction execution cycle.

4.6.6.6 Indirect Call - Mode 2. Mode 2 interrupts via the $\overline{INTR}$ input provide the greatest flexibility in structuring interrupt handling software. The CPU response to the interrupt request on $\overline{INTR}$, if enabled, is to respond via $\overline{INTA}$, as shown in *Figure 4-21*, when a single vector is read from the interrupting device. The $\overline{INTA}$ is used in place of the $\overline{RD}$ strobe to read the vector from the interrupting device. This vector is used in conjunction with the internal I register to generate a pointer to the address of a table entry that contains the handling routine address, in effect an indirect call. *Figure 4-22* illustrates the addressing chain.

From *Figure 4-22*, note that the I register and interrupt vector can be viewed as supplying the page location of the table of interrupt handling routine addresses, and the number of the entry in that table. The low-order bit of the peripheral supplied vector must be a zero, implying that entries in the table must start in even locations, low-order byte first, high-order byte second.

This mode of operation provides the system programmer the greatest flexibility in dynamically allocating memory resources for his interrupt handling routines. It is possible to move and/or change routines and just keep track of their location in a single table, which itself can be dynamically located and traced via the I register.

4.6.6.7 Interrupt Enable, Interrupt Disable. The NSC800 has two types of interrupt inputs, a non-maskable interrupt and four software maskable interrupts. The non-maskable interrupt ($\overline{NMI}$) cannot be disabled by the programmer and will be accepted whenever a peripheral device requests an interrupt. The $\overline{NMI}$ is usually reserved for important functions that must be serviced when they occur, such as an imminent power failure. The maskable interrupts ($\overline{INT}$, $\overline{RSTA}$, $\overline{RSTB}$, and $\overline{RSTC}$) can be selectively enabled or disabled by the programmer. This selectivity allows the programmer to disable the maskable interrupts during periods when the program has timing constraints that cannot allow itself to be interrupted. There are two interrupt enable flip flops (IFF_1 and IFF_2) on the NSC800. These flip flops are controlled by two instructions, Enable Interrupt (EI) and Disable Interrupt (DI). The state of IFF_1 is used to determine the enabling or disabling of the maskable interrupts, while IFF_2 is used as a temporary storage location for the state of IFF_1.

A reset to the CPU will force both IFF_1 and IFF_2 to the reset state so that maskable interrupts are disabled. They can then be enabled by an EI instruction at any time by the programmer. When an EI instruction is executed, any pending interrupt requests will not be accepted until after the instruction following EI has been executed. This single instruction delay is necessary in situations where the following instruction is a return instruction and interrupts must not be allowed until the return has been completed. The EI instruction sets both IFF_1 and IFF_2 to the enable state. When an interrupt is accepted by the CPU, both IFF_1 and IFF_2 are automatically reset, inhibiting further interrupts until the programmer wishes to issue a new EI instruction. Note that for all the previous cases, IFF_1 and IFF_2 are always equal.

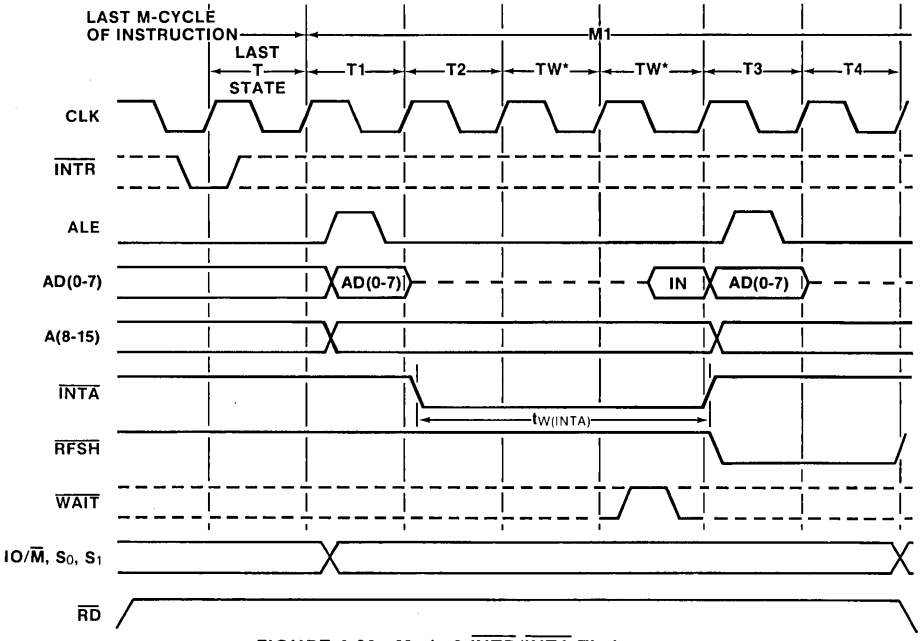


FIGURE 4-20. Mode 0 INTR/INTA Timing 45-16

The function of IFF_2 is to retain the status of IFF_1 when a non-maskable interrupt occurs. When a non-maskable interrupt is accepted, IFF_1 is reset to prevent further interrupts until reenabled by the programmer. Thus, after a non-maskable interrupt has been accepted, maskable interrupts are disabled but the previous state of IFF_1 was saved so that the complete state of the CPU just prior to the non-maskable interrupt may be restored at any time. When a Load Register I (LD A,I) instruction or a Load Register A with Register R (LD A, R) instruction is executed, the state of IFF_2 is copied into the parity flag where it may be tested or stored.

A second method of restoring the status of IFF_1 is through the execution of a Return Non-Maskable Interrupt (RETN) instruction. Since this instruction indicates that the non-maskable interrupt service routine is completed, the contents of IFF_2 are now copied back into IFF_1, so that the status of IFF_1 just prior to the acceptance of the non-maskable interrupt will be automatically restored.

Figure 4-23 depicts the status of the flip flops during a sample series of interrupt instructions.

4.6.6.8 Interrupt Control Register. The interrupt control register (ICR) is a 4-bit, write only register that provides the programmer with a second level of maskable control over the four maskable interrupt inputs.

The ICR is internal to the NSC800 CPU, but is addressed through the I/O space at I/O address port X'BB. Each bit in the register controls a mask bit dedicated to each maskable interrupt, $\overline{RSTA}$, $\overline{RSTB}$, $\overline{RSTC}$ and $\overline{INTR}$. For an interrupt request to be accepted on any of these inputs, the corresponding mask bit in the ICR must be set (= 1) and IFF_1 and IFF_2 must be set. This provides the programmer with control over individual interrupt inputs rather than just a system wide enable or disable.

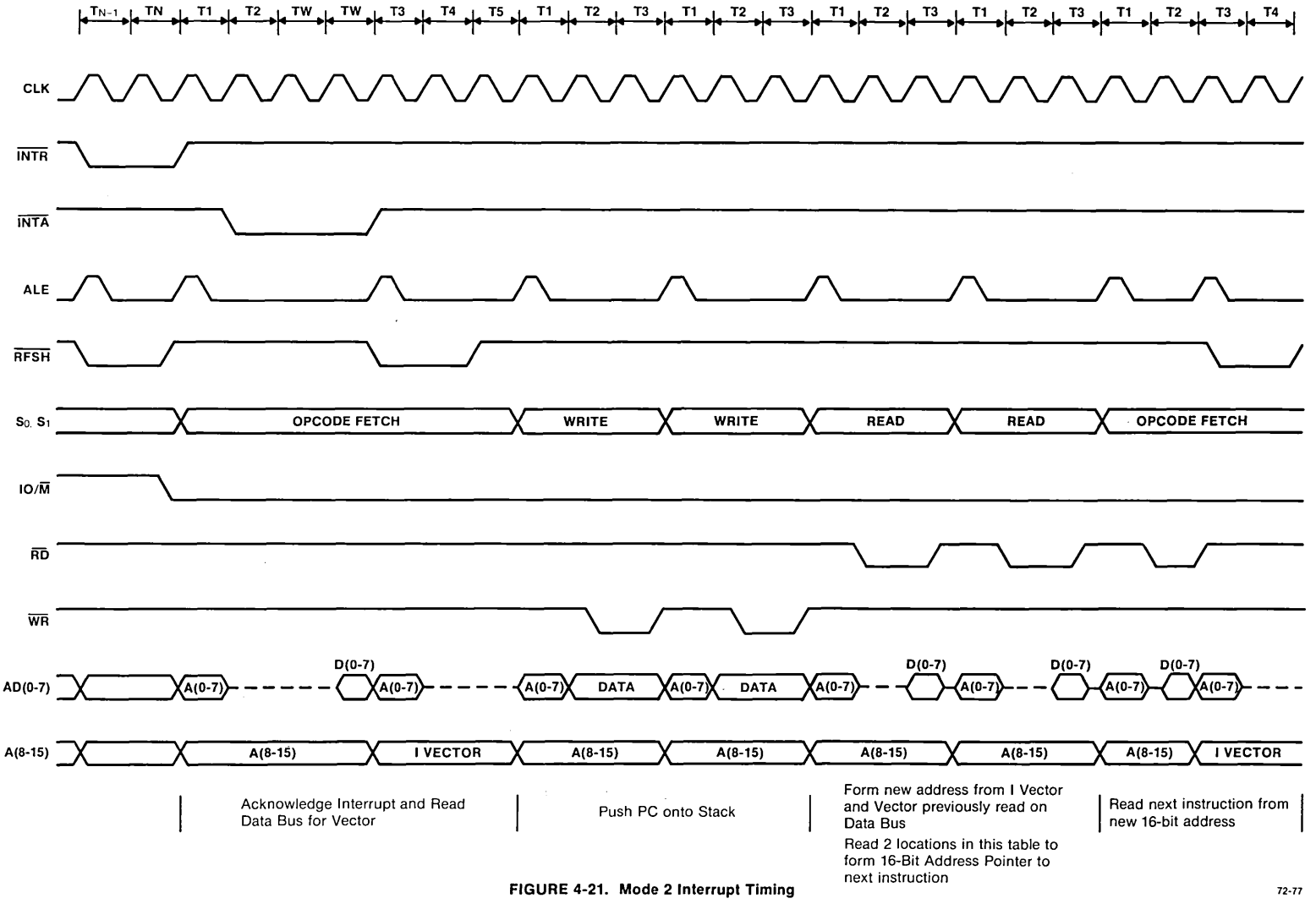


FIGURE 4-21. Mode 2 Interrupt Timing

72-77

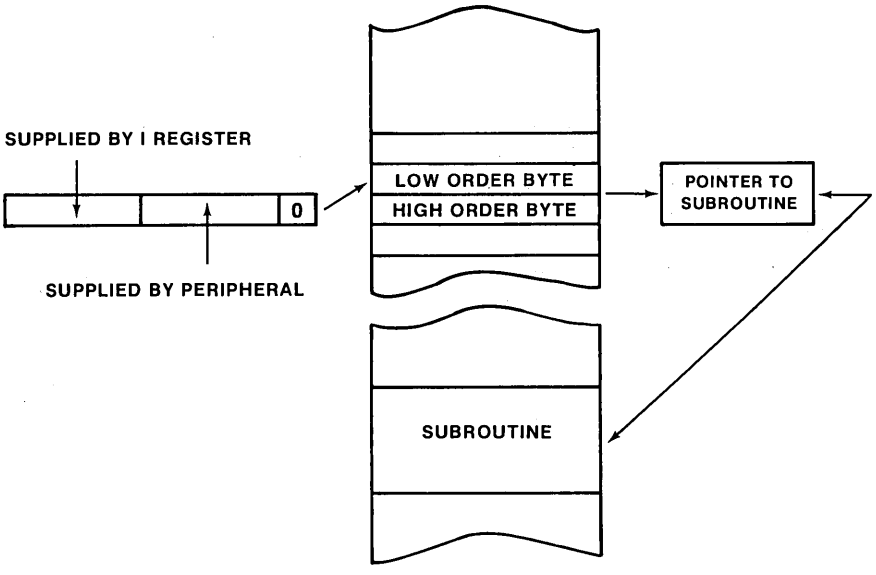


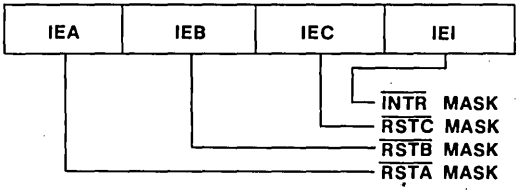
FIGURE 4-22. Mode 2 Interrupt Handling Addressing 45-3

Operation	IFF_1	IFF_2	Comment
INITIALIZE	0	0	Interrupt Disabled
EI	1	1	Interrupt Enabled
$\overline{INTR}$	0	0	Interrupt Disable and $\overline{INTR}$ Being Serviced
EI	1	1	Interrupt Enabled
RET	1	1	Interrupt Enabled
$\overline{NMI}$	0	1	Interrupt Disabled
RETN	1	1	Interrupt Enabled

Operation	IFF_1	IFF_2	Comment
$\overline{INTR}$	0	0	Interrupt Disabled
$\overline{NMI}$	0	0	Interrupt Disabled and $\overline{NMI}$ Being Serviced
RETN	0	0	Interrupt Disabled and $\overline{INTR}$ Being Serviced
EI	1	1	Interrupt Enabled
RET	1	1	Interrupt Enabled

FIGURE 4-23. IFF_1 and IFF_2 Operation

Figure 4-24 shows the bit assignment for the ICR.



For Each Mask: 0 = Disable
1 = Enable

FIGURE 4-24. Interrupt Control Register

4.6.7 Advanced Cycle Status

Two pins (27 and 29) on the NSC800 (see *Figure 4-25*) provide encoded status to the system as to which type of machine cycle is being executed by the CPU. The status can be used during emulation, prototyping, and/or debugging to help determine the status of the system. *Figure 4-26* depicts the advanced cycle state codes when used in conjunction with the IO/$\overline{M}$ output signal.

Depending on the complexity and sophistication of the peripheral configuration in the NSC800 system, the Advanced Cycle Status also performs an operating function. Specifically, in those systems that employ a daisy chain network for interrupt priority, the NSC800 has a mode of operation that requires peripherals to make use of the status information. As interrupts are accepted by the CPU, the status of the interrupt priority daisy chain changes with each recognition of the highest priority interrupting member. When the interrupt handling routine is complete, the daisy chain must be reconfigured to reestablish the enable status down the chain to lower priority peripherals.

In the NSC800 instruction repertoire there is a special instruction for this purpose, Return from Interrupt (RETI). It is the responsibility of the peripheral to recognize the fetch of this two-byte instruction (X'ED/X'4D) from memory by the CPU. To accomplish this, the decode of the opcode fetch cycle (S_1,S_0 = 11 respectively) is vital. Additional logic is required to prevent other X'ED/X'4D sequences from being mistaken for a RETI instruction.

4.6.8 Refresh

To take advantage of the low-power dissipation and high-chip densities, the NSC800 provides a refresh control (see *Figure 4-27*) for dynamic RAMs. To make this refresh operation as transparent as possible to the user, it is

S_1	S_0	IO/$\overline{M}$	STATUS
1	0	0	Memory Read
0	1	0	Memory Write
1	0	1	I/O Read
0	1	1	I/O Write
1	1	0	Opcode Fetch
0	0	0	Halt
1	0	0	Bus IDLE*

*ALE not surpressed during Bus IDLE

FIGURE 4-26. Advanced Cycle State Codes

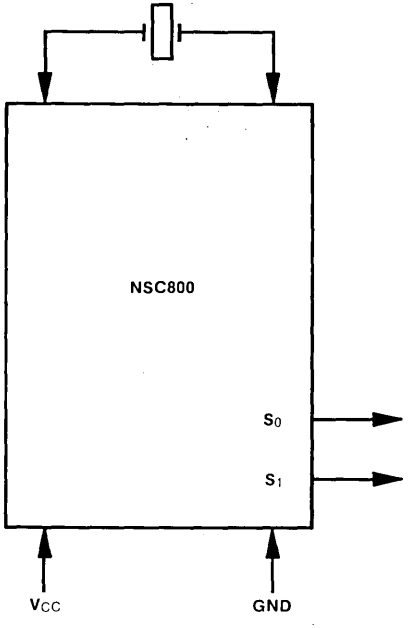


FIGURE 4-25. NSC800 Advanced Cycle Status Function

72-89

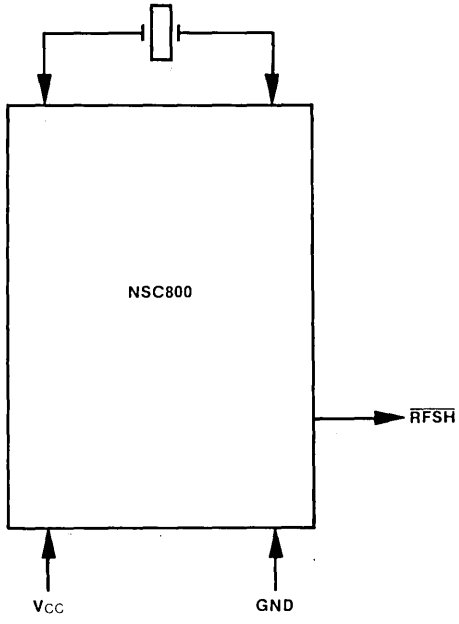


FIGURE 4-27. NSC800 Refresh Function

72-90

included in the second half of each opcode fetch cycle, as shown in *Figure 4-28.* The refresh feature does not continue during a power-save cycle, during bus request operations, or if extended wait requests are used.

The refresh operation provides a refresh address, an ALE strobe, and a refresh strobe. The refresh address is supplied by the R register in the NSC800 CPU. The R register provides a full 8-bit refresh address and is incremented after each fetch.

4.6.9 Power-Save

The NSC800 family offers state-of-the-art mid-range microprocessor performance with extremely low-power consumption (see *Figure 4-29*). Despite the power saving detailed in paragraph 4.7, the power-save function provides the user with the option of reducing the power consumption even further for those applications that require less power.

The power-save function is accomplished by suspending the internal clock in the CPU, thereby stopping CPU operation and halting accesses to other system components (see *Figure 4-30*). The oscillator continues to operate and the CLK signal is still available to other system components, such as the NSC810 timer circuit (see Chapter 6). This permits a system to self-program itself into periods of operation and periods of standby. By suspending operation and system access, power in P²CMOS systems can be reduced by approximately 50%.

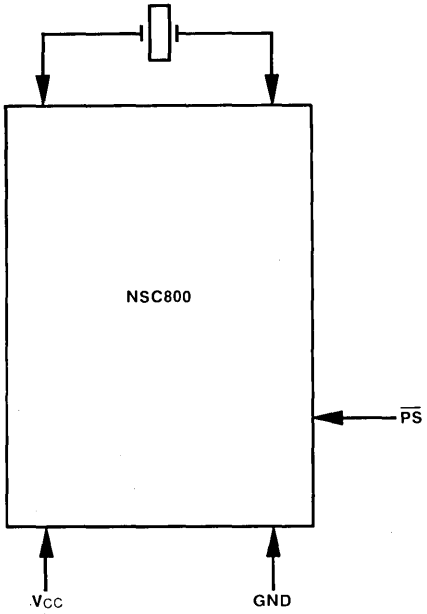


FIGURE 4-29. Power-Save Function

72-92

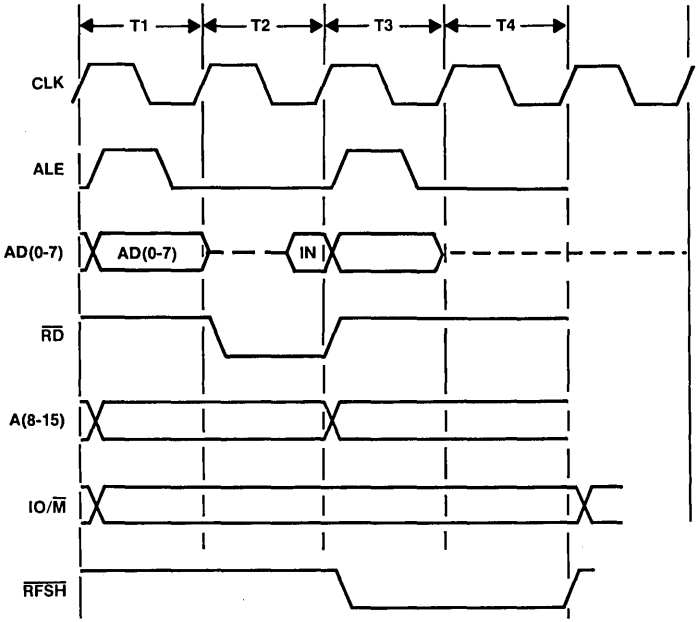


FIGURE 4-28. Refresh Timing

72-91

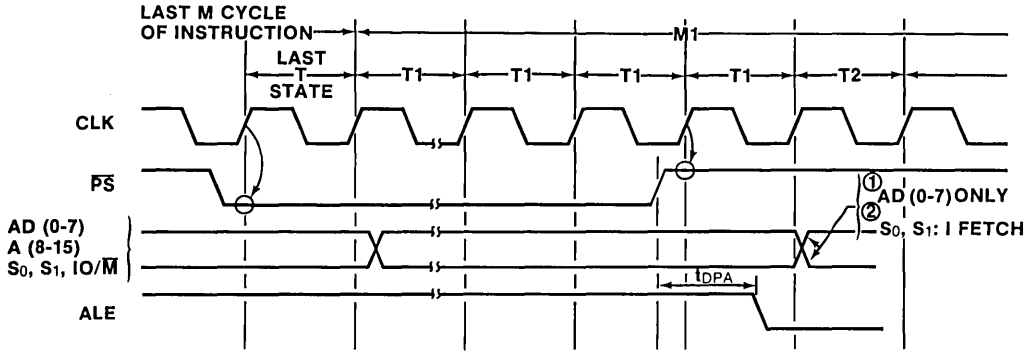


FIGURE 4-30. Power-Save Timing

Another possible use of the power-save feature allows for the implementation of a single instruction mode of operation, as shown in *Figure 4-31.* This is possible because the power-save ($\overline{PS}$) signal is sampled and acted upon at instruction boundaries.

The NSC800 can be held in a power-save mode indefinitely.

4.7 POWER CONSUMPTION

One primary goal of the NSC800 Family is to provide state-of-the-art performance with minimal power consumption. The power consumption of P²CMOS devices such as the NSC800 depends on factors like power supply voltage and operating frequency.

USE OF POWER SAVE ($\overline{PS}$) TO PROVIDE SINGLE INSTRUCTION CYCLE FEATURE

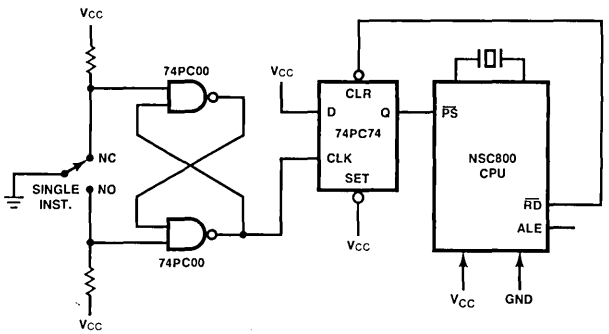


FIGURE 4-31. Power-Save Implementation of Single Instruction

Chapter 5

NSC800 - Software/Programming Description

5.1 INTRODUCTION

This chapter provides the reader with a detailed description of the NSC800 software. Each NSC800 instruction is described in terms of operation, opcode, execution speed, addressing modes supported, and flags affected.

5.2 ADDRESSING MODES

The following paragraphs describe the ten addressing modes supported by the NSC800. Note that particular addressing modes are often restricted to certain types of instructions. Paragraph 5.3 describes each instruction type and the addressing modes applicable to the specific instruction type.

The variety of modes supported, and the combination of modes allowed provide for a very flexible and powerful instruction set.

5.2.1 Register

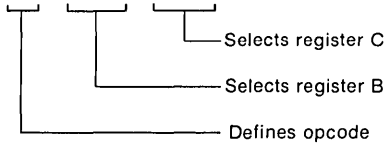
The most basic addressing mode is that which addresses data in the various CPU registers (refer to *Figure 4-1*). In these cases, bits in the opcode select specific registers that are to be addressed by the instruction.

Example:

Instruction: Load register B from register C

Mnemonic: LD B, C

Opcode: 0 1 0 0 0 0 0 1



In this instruction both the B and C registers are addressed by opcode bits.

5.2.2 Implied

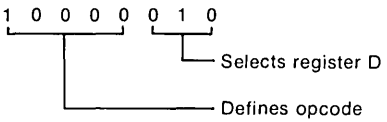
An extention to the register addressing mode is the implied addressing mode. In this mode, a specific register is used in the execution of the instruciton. In particular, arithmetic operations employ implied addressing since the A register is assumed to be the destination register for the result without being specifically referenced in the opcode.

Example:

Instruction: Add the contents of the D register to the Accumulator (A register)

Mnemonic: ADD A, D

Opcode: 1 0 0 0 0 0 1 0



In this instruction the D register is addressed with register addressing while the use of the A register is implied by the opcode.

5.2.3 Immediate

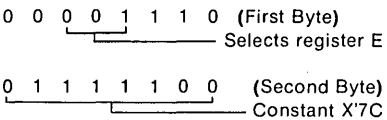
Before data within the CPU register set can be manipulated, the data must first be introduced into these registers. The most straightforward way this is accomplished is through immediate addressing where the data is contained in an additional byte of multi-byte instructions.

Example:

Instruction: Load the E register with the constant value X'7C.

Mnemonic: LD E, X'7C

Opcode: 0 0 0 0 1 1 1 0 (First Byte)



In this instruction, the E register is addressed with register addressing while the constant X'7C is immediate data in the second byte of the instruction.

5.2.4 Immediate Extended

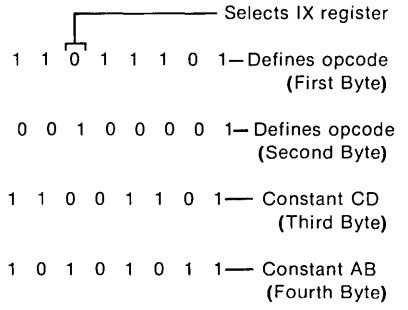
As immediate addressing is used to designate a byte of data, immediate extended addressing allows 16 bits of data to be supplied as an operand, in two additional bytes of the instruction.

Example:

Instruction: Load the 16-bit IX register with the constant value X'ABCD.

Mnemonic: LD IX, X'ABCD

Opcode:



In this instruction, the IX register is selected via register addressing while the 16-bit quantity X'ABCD is immediate data supplied as immediate extended format.

5.2.5 Direct Addressing

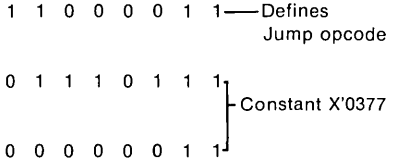
Direct addressing is the most straightforward way of addressing a location in the memory space. In direct addressing, 16 bits of memory address information are supplied in two bytes of data as part of the instruction. The memory address could be either data, source of destination, or a location for program execution, as in program control instructions.

Example:

Instruction: Jump to location X'0377

Mnemonic: JP X'0377

Opcode:



In this instruction, the Program Counter (PC) is loaded with the constant in the second and third bytes of the instruction. The program counter contents are transferred via direct addressing.

5.2.6 Register Indirect

Next to direct addressing, register indirect addressing provides the second most straightforward means of addressing memory. In register indirect addressing the address of the desired memory location is contained in a specified 16-bit register pair. The instruction references the register pair and the register contents define the memory location of the operand.

Example:

Instruction: Add the contents of memory location X'0254 to the A register. The HL register contains X'0254.

Mnemonic: ADD A, (HL)

Opcode: 1 0 0 0 0 1 1 0

This instruction uses implied addressing of the A and HL registers and register indirect addressing to access the data pointed to by the HL register.

5.2.7 Indexed

The most flexible mode of memory addressing is the indexed mode. This is similar to the register indirect mode of addressing because the base memory address is contained in one of two index registers, the IX or IY register. In addition, a byte of data is included in the instruction to act as a displacement to the address in the index register.

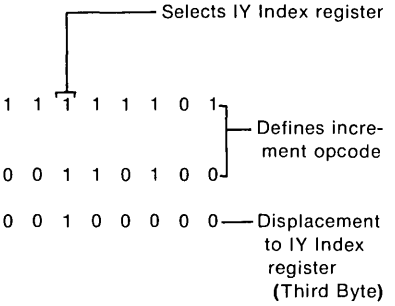
Indexed addressing is particularly useful in dealing with lists of data.

Example:

Instruction: Increment the data in memory location X'1020. The IY register contains X'1000.

Mnemonic: INC (IY + X'20)

Opcode:



The indexed addressing mode uses the contents of index register IY as a base address from which a pointer is formed to where the data is incremented.

5.2.8 Relative

Certain instructions allow memory locations to be addressed as a position relative to the PC register. These instructions are program control operations that allow jumps to locations around the memory location of the instruction. The displacement together with the current program location is determined through a displacement byte included in the instruction. The formation of this displacement byte is explained more fully in paragraph 5.3.

Example:

Instruction: Jump to a memory location seven bytes beyond the current location.

Mnemonic: JR $+7

Opcode:
```
0 0 0 1 1 0 0 0—Defines relative jump opcode
0 0 0 0 0 1 0 1—Displacement to be applied to the PC.
```

The program will continue at a location seven locations past the current PC.

5.2.9 Modified Page Zero

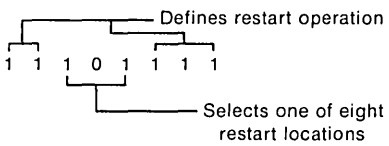
A limited set of NSC800 instructions (the Restart instructions) provides a code-efficient single byte instruction that allows CALLs to be performed to any one of eight dedicated locations in page zero (locations X'0000 to X'00FF). Normally a CALL is a three-byte instruction employing direct memory addressing.

Example:

Instruction: Perform a restart call to location X'0028.

Mnemonic: RST X'5

Opcode:



Program execution continues at location X'28 after execution of a single byte call employing modified page zero addressing.

5.2.10 Bit

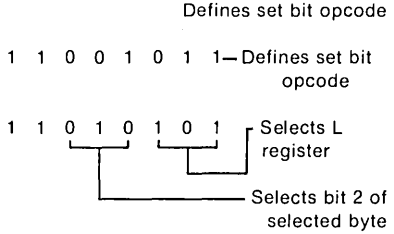
The NSC800 allows setting, resetting, and testing of individual bits in registers and memory data bytes.

Example:

Operation: Set bit 2 in the L register

Mnemonic: SET 2,L

Opcode:



Bit addressing allows the selection of bit 2 in the L register selected by register addressing.

5.3 INSTRUCTION SET

This paragraph details the entire NSC800 instruction set in terms of:

- Instruction
- Operation
- Opcode
- Timing
- Addressing Modes

The notations used in the instruction mnemonics and explanations are found in *Table 5-1.*

Other special notations used in the detailed object code explanations are found in *Table 5-2.*

TABLE 5-1. Mnemonic Notation Instruction Set

In the following instruction set listing, the notations used are shown below.

b: Used in instructions employing bit mode addressing to designate one bit in a register or memory location.

cc: Designates condition codes used in conditional Jumps, Calls, and Return instructions; may be:

NZ = Non Zero (Z flag = 0)
Z = Zero (Z flag = 1)
NC = Non Carry (C flag = 0)
C = Carry (C flag = 1)
PO = Parity Odd or No Overflow (P/V = 0)
PE = Parity Even or Overflow (P/V = 1)
P = Positive (S = 0)
M = Negative (S = 1)

TABLE 5-1. Mnemonic Notation Instruction Set (Cont'd.)

d: Used in instructions employing relative or indexed modes of addressing to designate an 8-bit signed complement displacement.

kk: Subset of cc condition codes used in conjunction with conditional relative jumps; may be NZ, Z, NC or C.

m_1: Used in instructions employing register indirect or indexed modes of addressing; may be **(HL)**, **(IX+d)** or **(IY+d)**.

m_2: Used in instructions employing register indirect or direct modes of addressing; may be **(BC)**, **(DE)** or **(nn)**.

n: Any 8-bit binary number.

nn: Any 16-bit binary number

pp: Used in 16-bit arithmetic instructions employing register modes of addressing; may be BC, DE, SP or register designated as destination operand.

qq: Used in instructions employing register modes of addressing; may be BC, DE, HL, AF, IX or IY.

r: Used in instructions employing register mode of addressing; may be A, B, C, D, E, H or L.

rr: Used in instructions employing register modes of addressing; may be BC, DE, HL, SP, IX or IY.

ss: Used in instructions employing register mode of addressing; may be HL, IX or IY.

T: Used in restart instructions employing modified page zero addressing mode; may take on hex values of 0, 8, 10, 18, 20, 28, 30 or 38.

X_L: Subscript L indicates the low order byte of a 16-bit register.

X_H: Subscript H indicates the high order byte of a 16-bit register.

( ): Parenthesis indicate the contents are considered a pointer to a memory or I/O location.

TABLE 5-2. Assembled Object Code Notation

REGISTER:

r	Register
000	B
001	C
010	D
011	E
100	H
101	L
111	A

TABLE 5-2. Assembled Object Code Notation (Cont'd.)

rp

00	BC
01	DE
10	HL
11	SP

rs

00	BC
01	DE
10	HL
11	AF

CONDITION CODES:

cc	Mnemonic	True Flag Condition
000	NZ	Z = 0
001	Z	Z = 1
010	NC	C = 0
011	C	C = 1
100	PO	P/V = 0
101	PE	P/V = 1
110	P	S = 0
111	M	S = 1

kk	Mnemonic	True Flag Condition
00	NZ	Z = 0
01	Z	Z = 1
10	NC	C = 0
11	C	C = 1

RESTART ADDRESSES

t	T
000	X'00
001	X'08
010	X'10
011	X'18
100	X'20
101	X'28
110	X'30
111	X'38

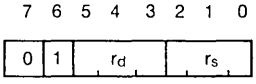
5.3.1 8-Bit Loads

5.3.1.1 Register to Register

LD r_d, r_s

Load register r_d with r_s

$r_d \leftarrow r_s$ No flags affected



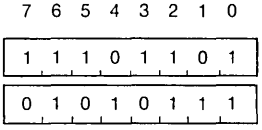
Timing: 4 T States

Addressing Mode: Register

LD A, I

Load Accumulator with the contents of the I register.

A ← I
- S: Set according to sign
- Z: Set according to zero equality
- H: Reset
- P/V: Set according to IFF_2 (zero if interrupt occurs during operation)
- N: Reset
- C: Not affected



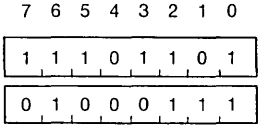
Timing: 9 T states

Addressing Mode: Register

LD I, A

Load Interrupt Vector Register (I) with the contents of A

I ← A No flags affected



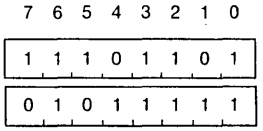
Timing: 9 T states

Addressing Mode: Register

LD A, R

Load Accumulator with contents of R register

A ← R
- S: Set according to sign
- Z: Set according to zero equality
- H: Reset
- P/V: Set according to IFF_2
- N: Reset
- C: Not affected



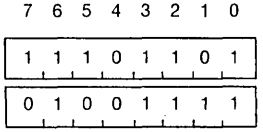
Timing: 9 T states

Addressing Mode: Register

LD R, A

Load Refresh Register (R) with contents of the Accumulator

R ← A No flags affected



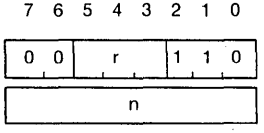
Timing: 9 T states

Addressing Mode: Register

LD r, n

Load register r with immediate data n

r ← n No flags affected



Timing: 7 T states

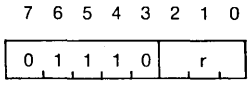
Addressing Mode: Source - Immediate
Destination - Register

5.3.1.2 Register to Memory

LD m_1, r

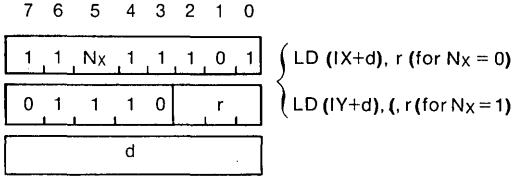
Load memory from register r

$m_1 \leftarrow r$ No flags affected



Timing: 7 T states

Addressing Mode: Source - Register
Destination - Register Indirect



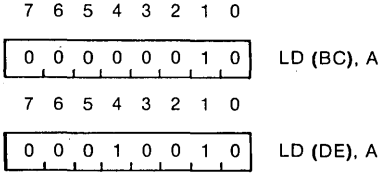
Timing: 19 T states

Addressing Mode: Source - Register
Destination - Indexed

LD m_2, A

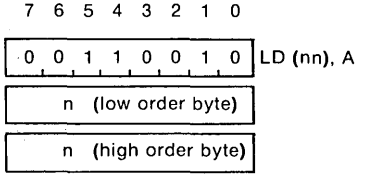
Load memory from the Accumulator

$m_2 \leftarrow A$ No flags affected



Timing: 7 T states

Addressing Mode: Source - Register (Implied)
Destination - Register Indirect



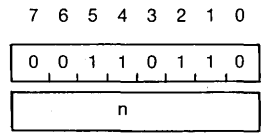
Timing: 13 T states

Addressing Mode: Source - Register (Implied)
Destination - Direct

LD m_1, n

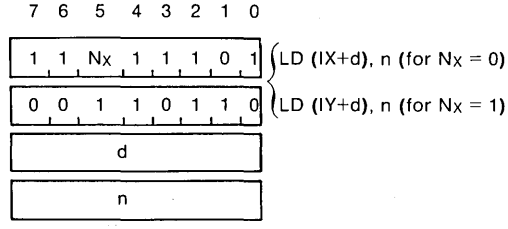
Load memory with immediate data

$m_1 \leftarrow n$ No flags affected



Timing: 10 T states

Addressing Mode: Source - Immediate
Destination - Register Indirect



Timing: 19 T states

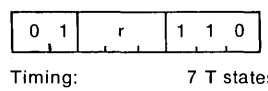
Addressing Mode: Source - Immediate
Destination - Indexed

5.3.1.3 Memory to Register

LD r, m_1

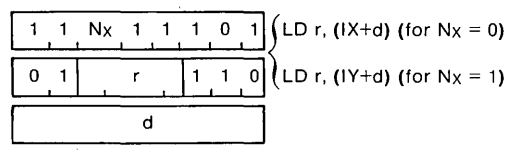
Load register r from memory location m_1

$r \leftarrow m_1$ No flags affected



Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Register



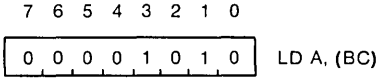
Timing: 19 T states

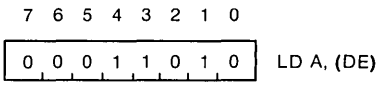
Addressing Mode: Source - Indexed
Destination - Register

LD A, m_2

Load the Accumulator from memory location m_2

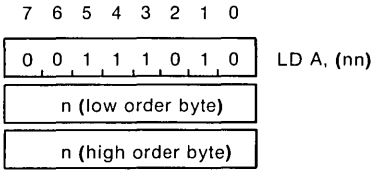
A ← m_2 No flags affected





Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Register (Implied)



Timing 13 T states

Addressing Mode: Source - Immediate Extended
Destination - Register (Implied)

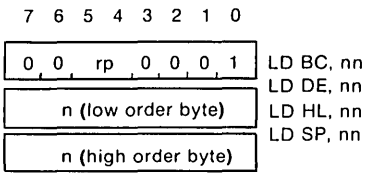
5.3.2 16-Bit Loads

5.3.2.1 Register to Register

LD rr, nn

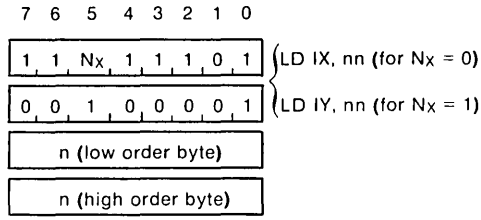
Load 16-bit register pair with immediate data

rr← nn No flags affected



Timing: 10 T states

Addressing Mode: Source - Immediate Extended
Destination - Register



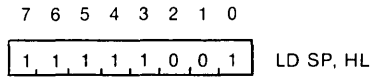
Timing: 14 T states

Addressing Mode: Source - Immediate Extended
Destination - Register

LD SP, ss

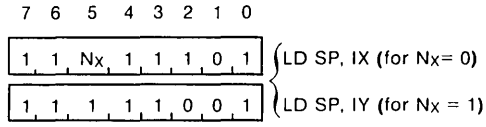
Load the SP from 16-bit register ss

SP ← ss No flags affected



Timing: 6 T states

Addressing Mode: Source - Register
Destination - Register (Implied)



Timing: 10 T states

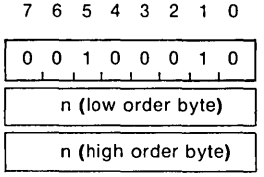
Addressing Mode: Source - Register
Destination - Register (Implied)

5.3.2.2 Register to Memory

LD (nn), rr

Load memory location nn with contents of 16-bit register rr

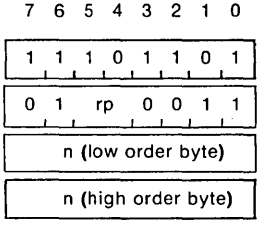
(nn) ← rr_L No flags affected
(nn+1) ← rr_H



LD (nn), HL (note an alternate opcode below)

Timing: 16 T states

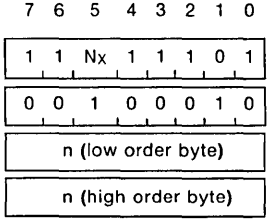
Addressing Mode: Source - Register
Destination - Direct



LD (nn), BC
LD (nn), DE
LD (nn), HL
LD (nn), SP

Timing: 20 T states

Addressing Mode: Source - Register
Destination - Direct



LD (nn), IX (for N_X = 0)
LD (nn), IY (for N_X = 1)

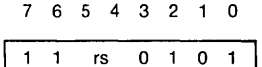
Timing: 20 T states

Addressing Mode: Source - Register
Destination - Direct

PUSH qq

Push the contents of register pair qq onto the memory stack.

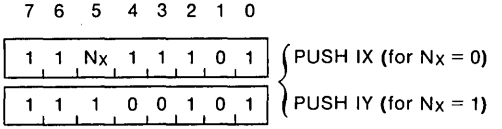
(SP-1) ← qq_H No flags affected
(SP-2) ← qq_L
SP ← SP-2



PUSH BC
PUSH DE
PUSH HL
PUSH AF

Timing: 11 T states

Addressing Mode: Source - Register
Destination - Register Indirect
(Stack)



PUSH IX (for N_X = 0)
PUSH IY (for N_X = 1)

Timing: 15 T states

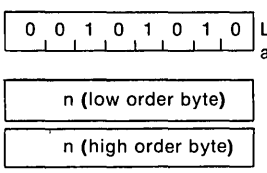
Addressing Mode: Source - Register
Destination - Register Indirect
(Stack)

5.3.2.3 Memory to Register

LD rr, (nn)

Load 16-bit register from memory location nn

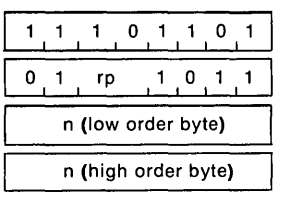
rr_L ← (nn) No flags affected
rr_H ← (nn+1)



LD HL, (nn) (note an alternate opcode below)

Timing: 16 T states

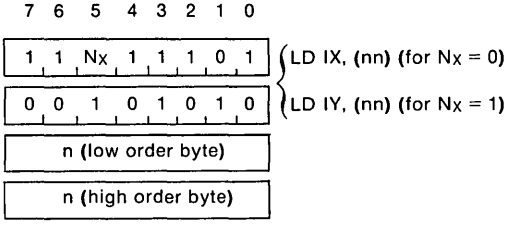
Addressing Mode: Source - Direct
Destination - Register



LD BC, (nn)
LD DE, (nn)
LD HL, (nn)
LD SP, (nn)

Timing: 20 T states

Addressing Mode: Source - Direct
Destination - Register



LD IX, (nn) (for N_X = 0)
LD IY, (nn) (for N_X = 1)

Timing: 20 T states

Addressing Mode: Source - Direct
Destination - Register

POP qq

Pop the top contents of the memory stack to register qq.

$qq_L \leftarrow (SP)$ No flags affected
$qq_H \leftarrow (SP+1)$
$SP \leftarrow SP+2$

7	6	5	4	3	2	1	0	
1	1	rs	0	0	0	1		POP BC POP DE POP HL POP AF

Timing: 10 T states

Addressing Mode: Source - Register Indirect (Stack)
Destination - Register

7	6	5	4	3	2	1	0	
1	1	N_X	1	1	1	0	1	POP IX (for $N_X = 0$)
1	1	1	0	0	0	0	1	POP IY (for $N_X = 1$)

Timing: 14 T states

Addressing Mode: Source - Register Indirect (Stack)
Destination - Register

5.3.3 8-Bit Arithmetic

5.3.3.1 Register Addressing Arithmetic

ADD A,r

Add contents of register r to the Accumulator.

$A \leftarrow A + r$
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry out of bit 3
P/V: Set according to overflow condition
N: Reset
C: Set according to carry from bit 7

7	6	5	4	3	2	1	0
1	0	0	0	0	r		

Timing: 4 T states

Addressing Mode: Source - Register
Destination - Implied

ADC A,r

Add contents of register r, plus the carry flag, to the Accumulator.

$A \leftarrow A + r + CY$
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry out of bit 3
P/V: Set according to overflow condition
N: Reset
C: Set according to carry from bit 7

7	6	5	4	3	2	1	0
1	0	0	0	1	r		

Timing: 4 T states

Addressing Mode: Source - Register
Destination - Implied

SUB r

Subtract the contents of register r from the Accumulator.

$A \leftarrow A - r$
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow

7	6	5	4	3	2	1	0
1	0	0	1	0	r		

Timing 4 T states

Addressing Mode: Source - Register
Destination - Implied

SBC A,r

Subtract contents of register r and the carry bit C from the Accumulator.

$A \leftarrow A - r - CY$
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow

7	6	5	4	3	2	1	0
1	0	0	1	1	r		

Timing: 4 T states

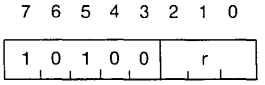
Addressing Mode: Source - Register
Destination - Implied

AND r

Logically AND the contents of the r register and the Accumulator.

$A \leftarrow A \wedge r$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set
P/V: Set according to parity of result
N: Reset
C: Reset



Timing: 4 T states

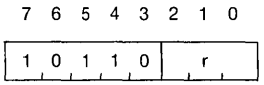
Addressing Mode: Source - Register
Destination - Implied

OR r

Logically OR the contents of the r register and the Accumulator.

$A \leftarrow A \vee r$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Reset



Timing: 4 T states

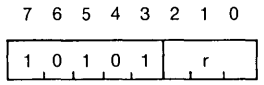
Addressing Mode: Source - Register
Destination - Implied

XOR r

Logically exclusively OR the contents of the r register with the Accumulator.

$A \leftarrow A \oplus r$

S: Set according to sign of the result
Z: Set according to equality of the result to zero
H: Reset
P/V: Set according to parity of the result
N: Reset
C: Reset



Timing: 4 T states

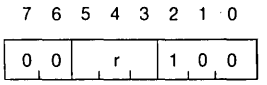
Addressing Mode: Source - Register
Destination - Implied

INC r

Increment register r

$r \leftarrow r + 1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry from bit 3
P/V: Set only if r was X'7F before operation
N: Reset
C: N/A



Timing: 4 T states

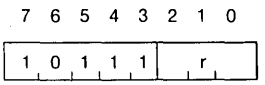
Addressing Mode: Source - Register
Destination - Register

CP r

Compare the contents of register r with the Accumulator and set the flags accordingly.

A - r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow
N: Set
C: Set according to borrow



Timing: 4 T states

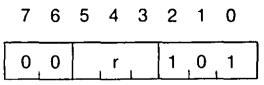
Addressing Mode: Source - Register
Destination - Implied

DEC r

Decrement the contents of register r

$r \leftarrow r - 1$

S: Set according to sign of result
Z: Set according to equality of the result at zero
H: Set according to a borrow from bit 4
P/V: Set only if r was X'80 prior to operation
N: Set
C: N/A



Timing: 4 T states

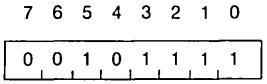
Addressing Mode: Source - Register
Destination - Register

CPL

Complement the Accumulator (1's complement)

$A \leftarrow \overline{A}$

S: N/A
Z: N/A
H: Set
P/V: N/A
N: Set
C: N/A



Timing: 4 T states

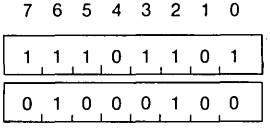
Addressing Mode: Implied

NEG

Negate the Accumulator (2's complement)

A ← 0 - A

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set only if Accumulator was X'80 prior to operation
N: Set
C: Set only if accumulator was not X'00 prior to operation



Timing: 8 T states

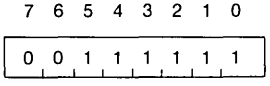
Addressing Mode: Implied

CCF

Complement the carry flag.

$CY \leftarrow \overline{CY}$

S: N/A
Z: N/A
H: Previous Carry
P/V: N/A
N: Reset
C: Complement of previous carry



Timing: 4 T states

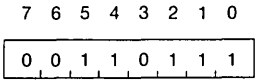
Addressing Mode: Implied

SCF

Set the carry flag

CY ← 1

S: N/A
Z: N/A
H: Reset
P/V: N/A
N: Reset
C: Set



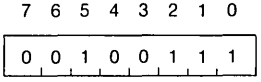
4 T states

Addressing Mode: Implied

DAA

Adjust the Accumulator for BCD addition and subtraction operations. To be executed after BCD data has been operated upon the standard binary ADD, ADC, INC, SUB, SBC, DEC or NEG instructions, (see *Table 5-3.*)

S: Set according to bit 7 of result
Z: Set according to equality of Accumulator to zero
H: Set according to instructions
P/V: Set according to parity of result
N: N/A
C: Set according to instructions



Timing: 4 T states

Addressing Mode: Implied

TABLE 5-3. Register Addressing Arithmetic

OPERATION	C BEFORE DAA	HEX VALUE IN UPPER DIGIT (bit 7-4)	H BEFORE DAA	HEX VALUE IN LOWER DJGIT (bit 3-0)	NUMBER ADDED TO BYTE	C AFTER DAA
	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
ADD	0	0-9	1	0-3	06	0
ADC	0	A-F	0	0-9	60	1
INC	0	9-F	0	A-F	66	1
	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
SUB	0	0-9	0	0-9	00	0
SBC	0	0-8	1	6-F	FA	0
DEC	1	7-F	0	0-9	A0	1
NEG	1	6-F	1	6-F	9A	1

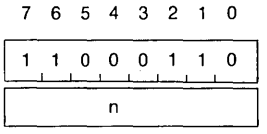
5.3.3.2 Immediately Addressed Arithmetic

ADD A,n

Add the immediate data n to the Accumulator.

A ← A + n

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry from bit 3
P/V: Set according to overflow condition
N: Reset
C: Set according to carry from bit 7



Timing: 7 T states

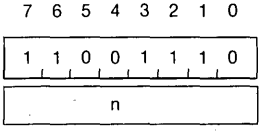
Addressing Mode: Source - Immediate
Destination - Implied

ADC A,n

Add, with carry, the immediate data n and the Accumulator.

A ← A + n + CY

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry from bit 3
P/V: Set according to overflow condition
N: Reset
C: Set according to carry from bit 7



Timing: 7 T states

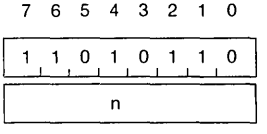
Addressing Mode: Source - Immediate
Destination - Implied

SUB n

Subtract the immediate data n from the Accumulator.

A ← A - n

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition



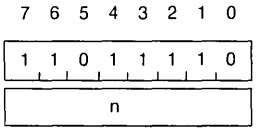
Timing: 7 T states

Addressing Mode: Source - Immediate
Destination - Implied

SBC A,n

Subtract, with carry, the immediate data n from the Accumulator.

A ← A - n - CY
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition



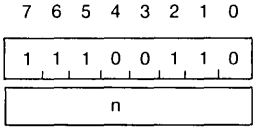
Timing: 7 T states

Addressing Mode: Source - Immediate
Destination - Implied

AND n

The immediate data n is logically AND'ed to the Accumulator.

A ← A∧n
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set
P/V: Set according to parity of result
N: Reset
C: Reset



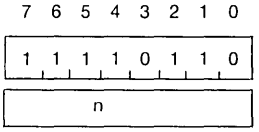
Timing: 7 T states

Addressing Mode: Source - Immediate
Destination - Implied

OR n

The immediate data n is logically OR'd to the contents of the Accumulator.

A ← A∨ s
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Reset



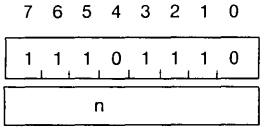
Timing: 7 T states

Addressing Mode: Source - Immediate
Destination - Implied

XOR n

The immediate data n is exclusively OR'd with the Accumulator.

A ← A ⊕ n
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Reset



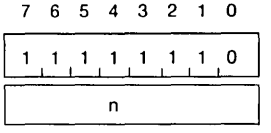
Timing: 7 T states

Addressing Mode: Source - Immediate
Destination - Implied

CP n

Compare the immediate data n with the contents of the Accumulator via subtraction and return the appropriate flags. The contents of the Accumulator are not affected.

A - n
S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition



Timing: 7 T states

Addressing Mode: Immediate

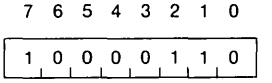
5.3.3.3 Memory Addressed Arithmetic

ADD A, m_1

Add the contents of the memory location m_1 to the Accumulator.

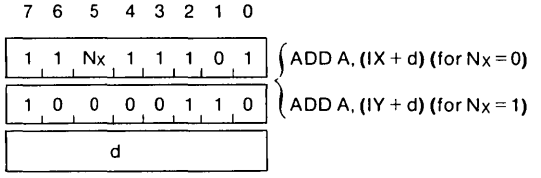
$A \leftarrow A + m_1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry from bit 3
P/V: Set according to overflow condition
N: Reset
C: Set according to carry from bit 7



Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Implied



Timing: 19 T states

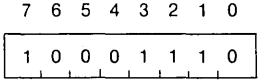
Addressing Mode: Source - Indexed
Destination - Implied

ADC A,m_1

Add the contents of the memory location m_1 plus the carry to the Accumulator.

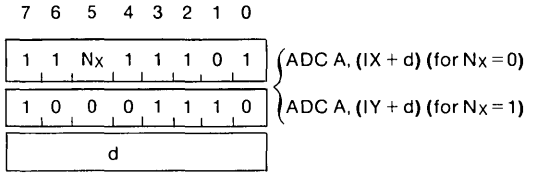
$A \leftarrow A + m_1 + CY$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry from bit 3
P/V: Set according to overflow condition
N: Reset
C: Set according to carry from bit 7



Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Implied



Timing: 19 T states

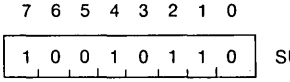
Addressing Mode: Source - Indexed
Destination - Implied

SUB m_1

Subtract the contents of memory location m_1 from the Accumulator.

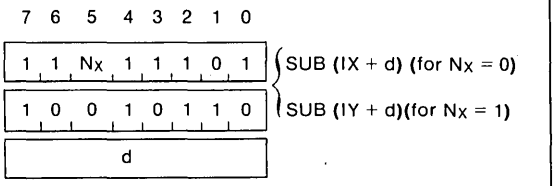
$A \leftarrow A - m_1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition



Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Implied



Timing: 19 T states

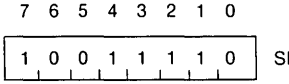
Addressing Mode: Source - Indexed
Destination - Implied

SBC A,m_1

Subtract, with carry, the contents of memory location m_1 from the Accumulator.

$A \leftarrow A - m_1 - CY$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition



Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Implied

7	6	5	4	3	2	1	0	
1	1	N_X	1	1	1	0	1	SBC A, (IX + d) (for $N_X = 0$)
1	0	0	1	1	1	1	0	SBC A, (IY + d) (for $N_X = 1$)
d								

Timing: 19 T states

Addressing Mode: Source - Indexed
Destination - Implied

AND m_1

The data in memory location m_1 is logically AND'ed to the Accumulator.

$A \leftarrow A \wedge m_1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set
P/V: Set according to parity of the result
N: Reset
C: Reset

7	6	5	4	3	2	1	0	
1	0	1	0	0	1	1	0	AND (HL)

Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Implied

7	6	5	1	1	1	0	1	
1	1	N_X	1	1	1	0	1	AND (IX + d) (for $N_X = 1$)
1	0	1	0	0	1	1	0	AND (IY + d) (for $N_X = 0$)
d								

Timing: 19 T states

Addressing Mode: Source - Indexed
Destination - Implied

OR m_1

The data in memory location m_1 is logically OR'ed with the Accumulator.

$A \leftarrow A \vee m_1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Reset

7	6	5	4	3	2	1	0	
1	0	1	1	0	1	1	0	OR (HL)

Timing: 7 T states

Addressing Mode: Source - Register Indexed
Destination - Implied

7	6	5	4	3	2	1	0	
1	1	N_X	1	1	1	0	1	OR (IX + d) (for $N_X = 0$)
1	0	1	1	0	1	1	0	OR (IY + d) (for $N_X = 1$)
d								

Timing: 19 T states

Addressing Mode: Source - Indexed
Destination - Implied

XOR m_1

The data in memory location m_1 is exclusively OR'ed with the data in the Accumulator.

$A \leftarrow A \oplus m_1$

S: Set according to sign of result
Z: Set according to equality of result of zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Reset

7	6	5	4	3	2	1	0	
1	0	1	0	1	1	1	0	XOR (HL)

Timing: 7 T states

Addressing Mode: Source - Register Indexed
Destination - Implied

7	6	5	4	3	2	1	0	
1	1	N_X	1	1	1	0	1	XOR (IX + d) (for $N_X = 0$)
1	0	1	0	1	1	1	0	XOR (IY + d) (for $N_X = 1$)
d								

Timing: 19 T States

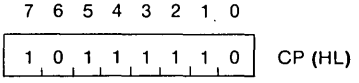
Addressing Mode: Source - Indexed
Destination - Implied

CP m_1

Compare the data in memory location m_1 with the data in the Accumulator via subtraction.

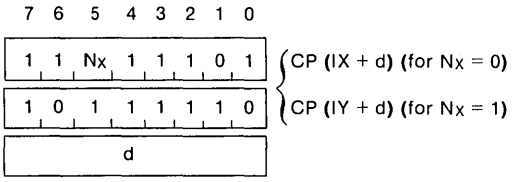
A - m_1

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition



Timing: 7 T states

Addressing Mode: Source - Register Indirect
Destination - Implied



Timing: 19 T states

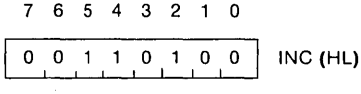
Addressing Mode: Source - Indexed
Destination - Implied

INC m_1

Increment data in memory location m_1.

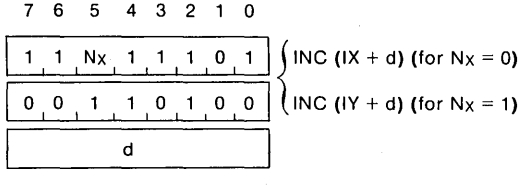
$m_1 \leftarrow m_1 + 1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry from bit 3
P/V: Set if data was X'7F before operation
N: Reset
C: N/A



Timing: 11 T states

Addressing Mode: Source - Register Indexed
Destination - Register Indexed



Timing: 23 T states

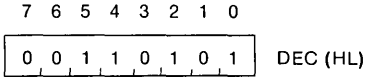
Addressing Mode: Source - Indexed
Destination - Indexed

DEC m_1

Decrement data in memory location m_1.

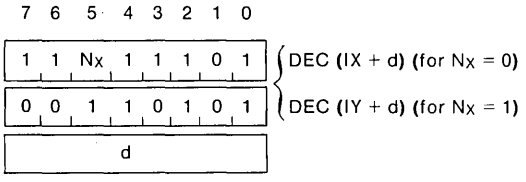
$m_1 \leftarrow m_1 - 1$

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 4
P/V: Set only if m_1 was X'80 before operation
N: Set
C: N/A



Timing: 11 T states

Addressing Mode: Source - Register Indexed
Destination - Register Indexed



Timing: 23 T states

Addressing Mode: Source - Indexed
Destination - Indexed

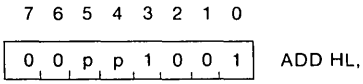
5.3.4 16-Bit Arithmetic

ADD ss, pp

Add the contents in the 16-bit register pp to the contents of the 16-bit register ss.

ss $\leftarrow$ ss + pp

S: N/A
Z: N/A
H: Set according to carry from bit 11
P/V: N/A
N: Reset
C: Set according to carry from bit 15



Timing: 11 T states

Addressing Mode: Source - Register
Destination - Register

7	6	5	4	3	2	1	0	
1	1	Nx	1	1	1	0	1	ADD IX, pp (for Nx = 0)
0	0	p	p	1	0	0	1	ADD IY, pp (for Nx = 1)

Timing: 15 T states

Addressing Mode: Source - Register
Destination - Register

ADC HL, pp

The contents of the 16-bit register pp are added, with the carry bit, to the HL register.

HL ← HL + pp + CY

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to carry out of bit 11
P/V: Set according to overflow condition
N: Reset
C: Set according to carry out of bit 15

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
0	1	p	p	1	0	1	0

Timing: 15 T states

Addressing Mode: Source - Register
Destination - Register

SBC HL, pp

Subtract, with carry, the contents of the 16-bit pp register from the 16-bit HL register.

HL ← HL - pp - CY

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Set according to borrow from bit 12
P/V: Set according to overflow condition
N: Set
C: Set according to borrow condition

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
0	1	p	p	0	0	1	0

Timing: 15 T states

Addressing Mode: Source - Register
Destination - Register

INC rr

Increment the contents of the 16-bit register rr.

rr ← rr +1 No flags affected

7	6	5	4	3	2	1	0	
0	0	rp			0	0	1	1	INC BC INC DE INC HL INC SP

Timing: 6 T states

Addressing Mode: Register

7	6	5	4	3	2	1	0	
1	1	Nx	1	1	1	0	1	INC IX (for Nx = 0)
0	0	1	0	0	0	1	1	INC IY (for Nx = 1)

Timing: 10 T states

Addressing Mode: Register

DEC rr

Decrement the contents of the 16-bit register rr.

rr ← rr - 1 No flags affected

7	6	5	4	3	2	1	0	
0	0	rp			1	0	1	1	DEC BC DEC DE DEC HL DEC SP

Timing: 6 T states

Addressing Mode: Register

7	6	5	4	3	2	1	0	
1	1	Nx	1	1	1	0	1	DEC IX (for Nx = 0)
0	0	1	0	1	0	1	1	DEC IY (for Nx = 1)

Timing: 10 T states

Addressing Mode: Register

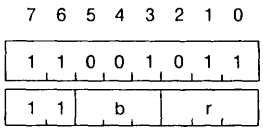
5.3.5 Bit Set, Reset, and Test

5.3.5.1 Register

SET b, r

Bit b in register r is set.

$R_b \leftarrow 1$ No flags affected



Timing: 8 T states

Addressing Mode: Bit/Register

RES b, r

Bit b in register r is reset.

$r_b \leftarrow 0$ No flags affected

Timing: 8 T states

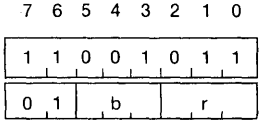
Addressing Mode: Bit/Register

BIT b, r

Bit b in register r is tested with the result put in the Z flag.

$Z \leftarrow \overline{r_b}$

S: Undefined
Z: Inverse of tested bit
H: Set
P/V: Undefined
N: Reset
C: N/A



Timing: 8 T states

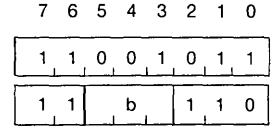
Addressing Mode: Bit/Register

5.3.5.2 Memory

SET b, m_1

Bit b in memory location m_1 is set.

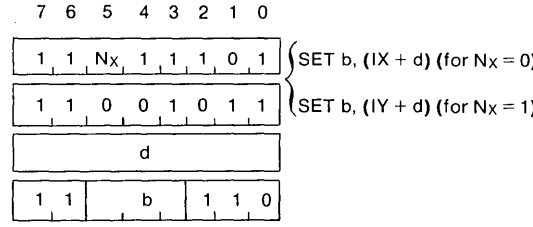
$m_{1b} \leftarrow 1$ No flags affected



SET b, (HL)

Timing: 15 T states

Addressing Mode: Bit/Register Indirect



SET b, (IX + d) (for $N_X = 0$)
SET b, (IY + d) (for $N_X = 1$)

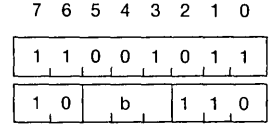
Timing: 23 T states

Addressing Mode: Bit/Indexed

RES b, m_1

Bit b in memory location m_1 is reset.

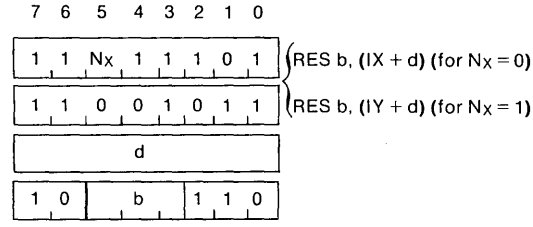
$m_{1b} \leftarrow 0$ No flags affected



RES b, (HL)

Timing: 15 T states

Addressing Mode: Bit/Register Indirect



RES b, (IX + d) (for $N_X = 0$)
RES b, (IY + d) (for $N_X = 1$)

Timing: 23 T states

Addressing Mode: Bit/Indexed

BIT b, m_1

Bit b in memory location m_1 is tested via the Z flag.

$Z \leftarrow \overline{m_{1b}}$

S: Undefined
Z: Inverse of tested bit
H: Set
P/V: Undefined
N: Reset
C: N/A

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	BIT b, (HL)
0	1	b			1	1	0	

Timing: 12 T states

Addressing Mode: Bit/Register Indirect

7 6 5 4 3 2 1 0

1	1	Nx	1	1	1	0	1	BIT b, (IX + d) (for Nx = 0)
1	1	0	0	1	0	1	1	BIT b, (IY + d) (for Nx = 1)
d								
0	1	b			1	1	0	

Timing: 20 T states

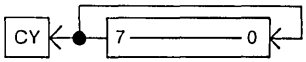
Addressing Mode: Bit/Indexed

5.3.6 Rotate and Shift

5.3.6.1 Register

RLC r

Rotate register r left circular



r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 7 of r

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	RLC r
0	0	0	0	0	r			(Note alternate for A register below)

Timing: 8 T states

Addressing Mode: Register

7 6 5 4 3 2 1 0

0	0	0	0	0	1	1	1	RLCA

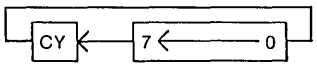
Timing: 4 T states

Addressing Mode: Implied

(Note RLCA does not affect S, Z, or P/V flags)

RL r

Rotate register r left through carry.



r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 7 of r

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	RL r
0	0	0	1	0	r			(Note alternate for A register below)

Timing: 8 T states

Addressing Mode: Register

7 6 5 4 3 2 1 0

0	0	0	1	0	1	1	1	RLA

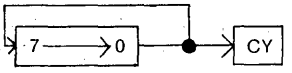
Timing: 4 T states

Addressing Mode: Implied

(Note RLA does not affect S, Z or P/V flags)

RRC r

Rotate register r right circular



r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of r

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	RRC r

0	0	0	0	1	r			(Note alternate for A register below)

Timing: 8 T states

Addressing Mode: Register

7 6 5 4 3 2 1 0

0	0	0	0	1	1	1	1	RRCA

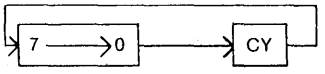
Timing: 4 T states

Addressing Mode: Implied

(Note RRCA does not affect S, Z, or P/V flags)

RR r

Rotate register r right through carry.



r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of r

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	RR r

0	0	0	1	1	r			(Note alternate A register below)

Timing: 8 T states

Addressing Mode: Register

7 6 5 4 3 2 1 0

0	0	0	1	1	1	1	1	RRA

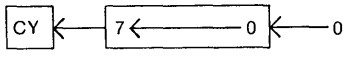
Timing: 4 T states

Addressing Mode: Implied

(Note RRA does not affect S, Z, or P/V flags)

SLA r

Shift register r left, arithmetic.



r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 7 of r

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	

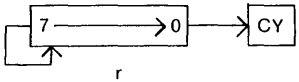
0	0	1	0	0	r			

Timing: 8 T states

Addressing Mode: Register

SRA r

Shift register r right arithmetic.



r

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of r

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1	

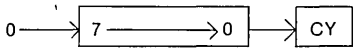
0	0	1	0	1	r			

Timing: 8 T states

Addressing Mode: Register

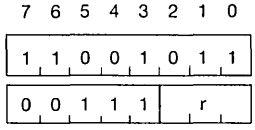
SRL r

Shift register r right logical.



r

S: Reset
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of r



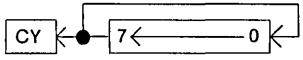
Timing: 8 T states

Addressing Mode: Register

5.3.6.2 Memory

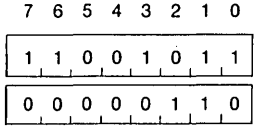
RLC m_1

Rotate data in memory location m_1 left circular.



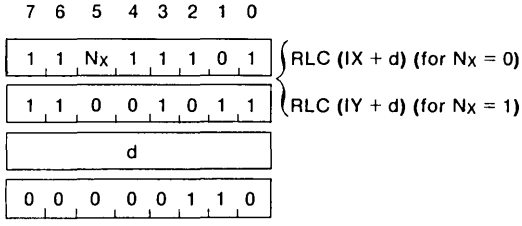
m_1

S:Set according to the sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 7 of m_1



Timing: 15 T states

Addressing Mode: Register Indirect

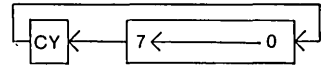


Timing: 23 T states

Addressing Mode: Indexed

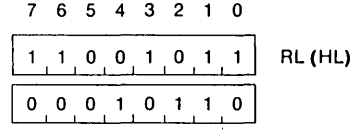
RL m_1

Rotate the data in memory location m_1 left through carry.



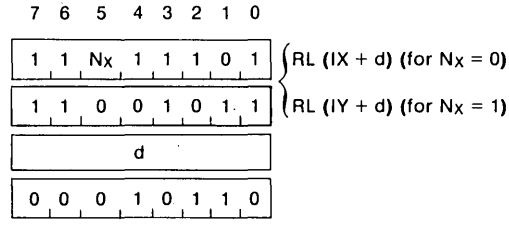
m_1

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 7 of m_1



Timing: 15 T states

Addressing Mode: Register Indirect

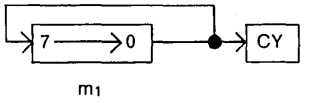


Timing: 23 T states

Addressing Mode: Indexed

RRC m_1

Rotate the data in memory location m_1 right circular.



m_1

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of m_1

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1
0	0	0	0	1	1	1	0

RRC (HL)

Timing: 15 T states

Addressing Mode: Register Indirect

7 6 5 4 3 2 1 0

1	1	N_X	1	1	1	0	1
1	1	0	0	1	0	1	1
d							
0	0	0	0	1	1	1	0

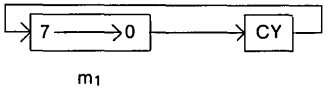
RRC (IX + d) (for N_X = 0)
RRC (IY + d) (for N_X = 1)

Timing: 23 T states

Addressing Mode: Indexed

RR m_1

Rotate the data in memory location m_1 right through the carry.



m_1

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of the result
N: Reset
C: Set according to data from bit 0 of m_1

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1
0	0	0	1	1	1	1	0

RR (HL)

Timing: 15 T states

Addressing Mode: Register Indirect

7 6 5 4 3 2 1 0

1	1	N_X	1	1	1	0	1
1	1	0	0	1	0	1	1
d							
0	0	0	1	1	1	1	0

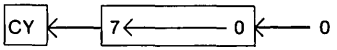
RR (IX + d) (for N_X = 0)
RR (IY + d) (for N_X = 1)

Timing: 23 T states

Addressing Mode: Indexed

SLA m_1

Shift the data in memory location m_1 left arithmetic.



m_1

S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 7 of m_1

7 6 5 4 3 2 1 0

1	1	0	0	1	0	1	1
0	0	1	0	0	1	1	0

SLA (HL)

Timing: 15 T states

Addressing Mode: Register Indirect

7 6 5 4 3 2 1 0

1	1	N_X	1	1	1	0	1
1	1	0	0	1	0	1	1
d							
0	0	1	0	0	1	1	0

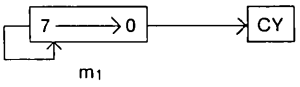
SLA (IX + d) (for N_X = 0)
SLA (IY + d) (for N_X = 1)

Timing: 23 T states

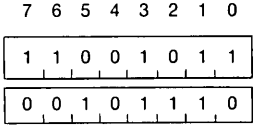
Addressing Mode: Indexed

SRA m_1

Shift the data in memory location m_1 right arithmetic.

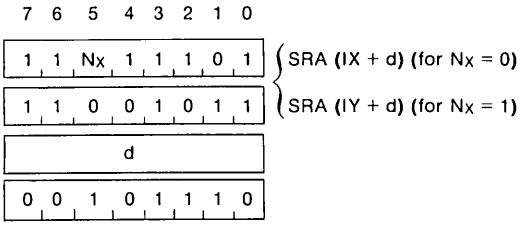


S: Set according to sign of result
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of m_1



Timing: 15 T states

Addressing Mode: Register Indirect

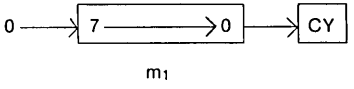


Timing: 23 T states

Addressing Mode: Indexed

SRL m_1

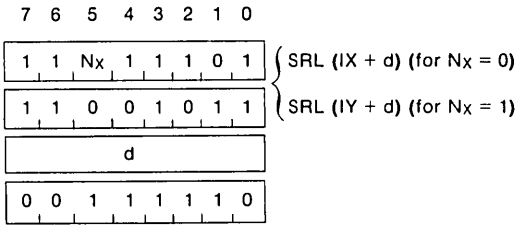
Shift left logical the data in memory location m_1.



S: Reset
Z: Set according to equality of result to zero
H: Reset
P/V: Set according to parity of result
N: Reset
C: Set according to bit 0 of m_1

Timing: 15 T states

Addressing Mode: Register Indirect



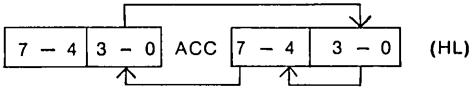
Timing: 23 T states

Addressing Mode: Indexed

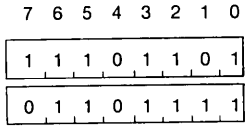
5.3.6.3 Register/Memory

RLD

Rotate digit left and right between the Accumulator and memory (HL).



S: Set according to sign of Accumulator
Z: Set according to equality of Accumulator to zero
H: Reset
P/V: Set according to parity of Accumulator
N: Reset
C: N/A

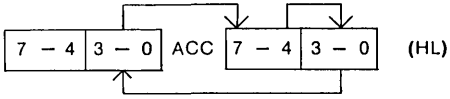


Timing: 18 T states

Addressing Mode: Implied/Register Indirect

RRD

Rotate digit right and left between the Accumulator and memory (HL).



S: Set according to sign of Accumulator
Z: Set according to quality of Accumulator to zero
H: Reset
P/V: Set according to parity of the Accumulator
N: Reset
C: N/A

7 6 5 4 3 2 1 0

1	1	1	0	1	1	0	1
0	1	1	0	0	1	1	1

Timing: 18 T states

Addressing Mode: Implied/Register Indirect

5.3.7 Exchanges

5.3.7.1 Register/Register

EX DE, HL

Exchange the contents of the 16-bit register pairs DE and HL.

DE ↔ HL No flags affected

7 6 5 4 3 2 1 0

1	1	1	0	1	0	1	1

Timing: 4 T states

Addressing Mode: Register

EX AF, A'F'

The contents of the Accumulator and flag register are exchanged with their corresponding alternate registers, that is A and F are exchanged with A' and F'

A ↔ A' No flags affected
F ↔ F'

7 6 5 4 3 2 1 0

0	0	0	0	1	0	0	0

Timing: 4 T states

Addressing Mode: Register

EXX

Exchange the contents of the BC, DE, and HL registers with their corresponding alternate register.

BC ↔ B'C' No flags affected
DE ↔ D'E'
HL ↔ H'L'

7 6 5 4 3 2 1 0

1	1	0	1	1	0	0	1

Timing: 4 T states

Addressing Mode: Implied

5.3.7.2 Register/Memory

EX (SP), ss

Exchange the two bytes at the top of the external memory stack with the 16-bit register ss.

(SP) ↔ SS_L No flags affected
(SP + 1) ↔ SS_H

7 6 5 4 3 2 1 0

1	1	1	0	0	0	1	1

EX (SP), HL

Timing: 19 T states

Addressing Mode: Register/Register Indirect

7 6 5 4 3 2 1 0

1	1	N_X	1	1	1	0	1

EX (SP), IX (for $N_X = 0$)

1	1	1	0	0	0	1	1

EX (SP), IY (for $N_X = 1$)

Timing: 23 T states

Addressing Mode: Register/Register Indirect

5.3.8 Memory Block Moves and Searches

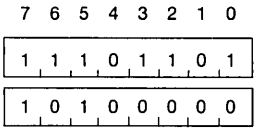
5.3.8.1 Single Operations

LDI

Move data from memory location (HL) to memory location (DE), increment memory pointers, and decrement byte counter BC.

(DE) ← (HL)
DE ← DE + 1
HL ← HL + 1
BC ← BC - 1

S: N/A
Z: N/A
H: Reset
P/V: Set if BC - 1 ≠ 0, otherwise reset
N: Reset
C: N/A



Timing: 16 T states

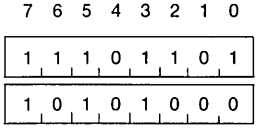
Addressing Mode: Register Indirect

LDD

Move data from memory location (HL) to memory location (DE), and decrement memory pointer and byte counter BC.

(DE) ← (HL)
DE ← DE - 1
HL ← HL - 1
BC ← BC - 1

S: N/A
Z: N/A
H: Reset
P/V: Set if BC - 1 ≠ 0, otherwise reset
N: Reset
C: N/A



Timing: 16 T states

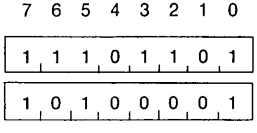
Addressing Mode: Register Indirect

CPI

Compare data in memory location (HL) to the Accumulator, increment the memory pointer, and decrement the byte counter. The Z flag is set if the comparison is equal.

A - (HL)
HL ← HL + 1
BC ← BC - 1
Z ← 1 if A = (HL)

S: Set according to sign of result of comparison subtract
Z: Set according to result of comparison
H: Set according to borrow from bit 4
P/V: Set if BC - 1 ≠ 0, otherwise reset
N: Set
C: N/A



Timing: 16 T states

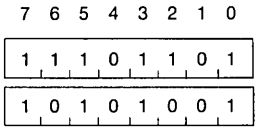
Addressing Mode: Register Indirect

CPD

Compare data in memory location (HL) to the Accumulator, and decrement the memory pointer and byte counter. The Z flag is set if the comparison is equal.

A - (HL)
HL ← HL - 1
BC ← BC - 1
Z ← 1 if A = (HL)

S: Set according to sign of result of comparison subtraction
Z: Set according to result of comparison
H: Set according to borrow from bit 4
P/V: Set if BC - 1 ≠ 0, otherwise reset
N: Set
C: N/A



Timing: 16 T states

Addressing Mode: Register Indirect

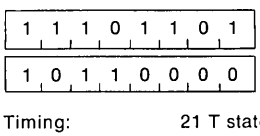
5.3.8.2 Repeat Operations

LDIR

Move data from memory location (HL) to memory location (DE), increment memory pointers, decrement byte counter BC, and repeat until BC = 0.

(DE) ← (HL)
DE ← DE + 1
HL ← HL + 1
BC ← BC - 1
Repeat until BC = 0

S: N/A
Z: N/A
H: Reset
P/V: Reset
N: Reset
C: N/A



Timing: 21 T states
16 T states (Last Cycle)

Addressing Mode: Register Indirect

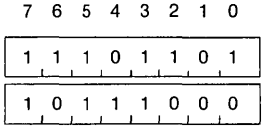
(Note, that each repeat is accomplished by a decrement of the PC so that refresh, etc. continues for each cycle.)

LDDR

Move data from memory location (HL) to memory location (DE), decrement memory pointers and byte counter BC, and repeat until BC = 0.

(DE) ← (HL)
DE ← DE - 1
HL ← HL - 1
BC ← BC - 1
Repeat Until BC = 0

S: N/A
Z: N/A
H: Reset
P/V: Reset
N: Reset
C: N/A



Timing: 21 T states
16 T states (last cycle only)

Addressing Mode: Register Indirect

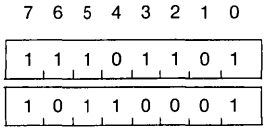
(Note, that each repeat is accomplished by a decrement of the PC so that refresh, etc. continues for each cycle.)

CPIR

Compare data in memory location (HL) to the Accumulator, increment the memory, decrement the byte counter BC, and repeat until BC = 0 or (HL) equals A.

A - (HL)
HL ← HL + 1
BC ← BC 1
Repeat until BC = 0 or A = (HL)

S: Set according to sign of subtraction performed for comparison
Z: Set if A = (HL), otherwise reset
H: Set according to borrow from bit 4
P/V: Set if BC - 1 ≠ 0, otherwise reset
N: Set
C: N/A



Timing: 21 T states
16 T states (last cycle only)

Addressing Mode: Register Indirect

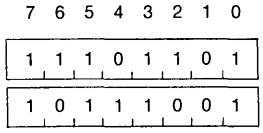
(Note, that each repeat is accomplished by a decrement of the PC so that refresh, etc. continues for each cycle.)

CPDR

Compare data in memory location (HL) to the contents of the Accumulator, decrement the memory pointer and byte counter BC, and repeat until BC = 0, or until (HL) equals the Accumulator.

A - (HL)
HL ← HL - 1
BC ← BC - 1
Repeat until BC = 0
or A = (HL)

S: Set according to sign of Subtraction performed for comparison
Z: Set according to equality of A and (HL), set if true
H: Set according to borrow from bit 4
P/V: Set if BC - 1 ≠ 0, otherwise reset
N: Set
C: N/A



Timing: 21 T states
16 T states (last cycle only)

Addressing Mode: Register Indirect

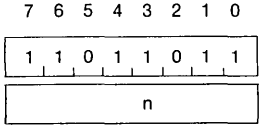
(Note, that each repeat is accomplished by a decrement of the PC so that refresh, etc. continues for each cycle.)

5.3.9 Input/Output

IN A, (n)

Input data to the Accumulator from the I/O device at address N.

A ← (n) No flags affected



Timing: 11 T states

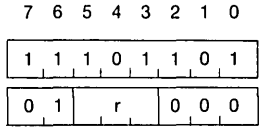
Addressing Mode: Source - Direct
Destination - Register

IN r, (C)

Input data to register r from the I/O device addressed by the contents of register C.

r ← (C)

S: Set according to sign of input data
Z: Set according to equality of input data to zero
H: Reset
P/V: Set according to parity of input data
N: Reset
C: N/A



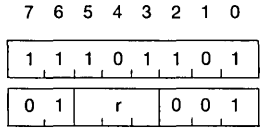
Timing: 12 T states

Addressing Mode: Source - Register Indirect
Destination - Register

OUT (C), r

Output register r to the I/O device addressed by the contents of register C.

(C) ← r No flags affected



Timing: 12 T states

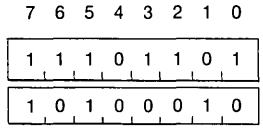
Addressing Mode: Source - Register
Destination - Register Indirect

INI

Input data from the I/O device addressed by the contents of register C to the memory location pointed to by the contents of the HL register. The HL pointer is incremented and the converter B is decremented.

(HL) ← (C)

S: Undefined
Z: Set if 8 -1 = 0, otherwise reset
H: Undefined
P/V: Undefined
N: Set
C: N/A



Timing: 16 T States

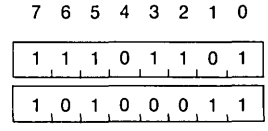
Addressing Mode: Implied Source - Register Indirect
Destination - Register Indirect

OUTI

Output data from memory location (HL) to the I/O device at port address (C), increment the memory pointer, and decrement the byte counter, B.

(C) ← (HL)
HL ← HL + 1
B ← B - 1

S: Undefined
Z: Set if B -1 = 0, otherwise reset
H: Undefined
P/V: Undefined
N: Set
C: N/A



Timing: 16 T states

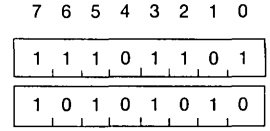
Addressing Mode: Implied Source - Register Indirect
Destination - Register Indirect

IND

Input data from I/O device at port address (C) to memory location (HL) and decrement HL memory pointer and byte counter B.

(HL) ← (C)
HL ← HL - 1
B ←

S: Undefined
Z: Set if B-1 = 0, otherwise reset
H: Undefined
P/V: Undefined
N: Set
C: N/A



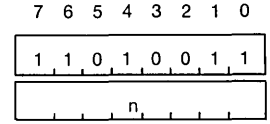
Timing: 16 T states

Addressing Mode: Implied - Source - Register Indirect
Destination - Register Indirect

OUT (n), A

Output the accumulator to the I/O device at address n.

(n) ← A No flags affected



Timing: 11 T states

Addressing Mode: Source - Register
Destination - Direct

OUTD

Data is output from memory location (HL) to the I/O device at port address (C) and then decrement the HL memory pointer and byte counter, B.

(C) ← (HL)
S: Undefined
Z: Set if B -1 = 0, otherwise reset
H: Underfined
P/V: Undefined
N: Set
C: N/A

Timing: 16 T states

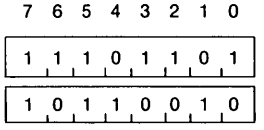
Addressing Mode: Implied/Source - Register Indirect
Destination - Register Indirect

INIR

Data is input from I/O device at port address (C) to memory location (HL), the HL memory pointer is incremented, and the byte counter B is decremented. The cycle is repeated until B = 0.

(Note that B is tested for zero after it is decremented. By loading B initially with zero, 256 data transfers will take place.)

(HL) ← (C)
HL ← HL + 1
B ← B - 1
Repeat until B = 0
S: Undefined
Z: Set
H: Undefined
P/V: Undefined
N: Set
C: N/A



Timing: 21 T states
16 T states (last cycle only)

Addressing Mode: Implied/Source - Register Indirect
Destination - Register Indirect

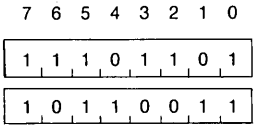
(Note, that at the end of each data transfer cycle, interrupts may be recognized and 2 refresh cycles will be performed.)

OTIR

Data is output to the I/O device at port address (C) from memory location (HL), the HL memory pointer is incremented, and the byte counter B is decremented. The cycles are repeated until B = 0.

(Note that B is tested for zero after it is decremented. By loading B initially with zero, 256 data transfers will take place.

(C) ← (HL)
HL ← HL + 1
B ← B - 1
Repeat until B = 0
S: Undefined
Z: Set
H: Undefined
P/V: Undefined
N: Set
C: N/A



Timing: 21 T States
16 T states (last cycle only)

Addressing Mode: Implied/Source - Register Indirect
Destination - Register Indirect

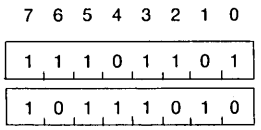
(Note that after each cycle interrupts may be recognized and 2 refresh cycles will be performed.)

INDR

Data is input from the I/O device at address (C) to memory location (HL), then the HL memory pointer and byte counter, B is decremented. The cycle is repeated until B = 0.

(Note that B is tested for zero after it is decremented. By loading B initially with zero, 256 data transfers will take place.

(HL) ← (C)
HL ← HL - 1
B ← B - 1
Repeat until B = 0
S: Undefined
Z: Set
H: Undefined
P/V: Undefined
N: Set
C: N/A



Timing: 21 T states
16 T states (last cycle only)

Addressing Mode: Implied/Source - Register Indirect
Destination - Register Indirect

(Note that after each data transfer cycle, interrupts may be recognized and 2 refresh cycles are performed.)

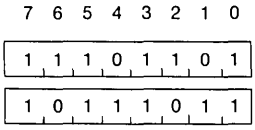
OTDR

Data is output from memory location (HL) to the I/O device at port address (C). Then the HL memory pointer and byte counter B is decremented. The cycle is repeated until B = 0.

(Note that B is tested for zero after it is decremented. By loading B initially with zero, 256 data transfers will take place.

(C) ← (HL)
HL ← HL - 1
B ← B - 1
Repeat until B = 0

S: Undefined
Z: Set
H: Undefined
P/V: Undefined
N: Set
C: N/A



Timing: 21 T states
16 T states (last cycle only)

Addressing Mode: Implied/Source - Register Indirect
Destination - Register Indirect

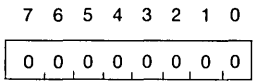
(Note that after each data transfer cycle the NSC800 will accept interrupts and perform 2 refresh cycles.)

5.3.10 CPU Control

NOP

The CPU performs no operation.

---- No flags affected



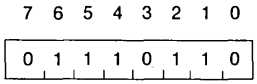
Timing: 4 T states

Addressing Mode: N/A

HALT

The CPU halts execution of the program. Internally NOP's are performed to keep the refresh circuits active until the CPU is interrupted or reset from the halted state.

---- No flags affected



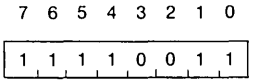
Timing: 4 T states

Addressing Mode: N/A

DI

Disable system level interrupts.

$IFF_1 \leftarrow 0$ No flags affected
$IFF_2 \leftarrow 0$



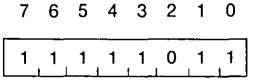
Timing: 4 T states

Addressing Mode: N/A

EI

The system level interrupts are enabled. During execution of this instruction, and the next one, the maskable interrupts will be disabled.

$IFF_1 \leftarrow 1$ No flags affected
$IFF_2 \leftarrow 1$



Timing: 4 T states

Addressing Mode: N/A

IM 0

The CPU is placed in interrupt mode 0. (See paragraph 4.6.6.4).

---- No flags affected

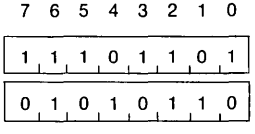
Timing: 8 T states

Addressing Mode: N/A

IM 1

The CPU is placed in interrupt mode 1. (See paragraph 4.6.6.5).

---- No flags affected



Timing: 8 T states

Addressing Mode: N/A

IM 2

The CPU is placed in interrupt mode 2. (See paragraph 4.6.6.6).

---- No flags affected

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
0	1	0	1	1	1	1	0

Timing: 8 T states

Addressing Mode: N/A

5.3.11 Program Control Group

5.3.11.1 Jumps

JP nn

Unconditional jump to program location nn.

PC ← nn No flags affected

7	6	5	4	3	2	1	0
1	1	0	0	0	0	1	1
n (low order byte)							
n (high order byte)							

Timing: 10 T states

Addressing Mode: Direct

JP (ss)

Unconditional jump to program location pointed to by register ss.

PC ← ss No flags affected

7	6	5	4	3	2	1	0	
1	1	1	0	1	0	0	1	JP (HL)

Timing: 4 T states

Addressing Mode: Register Indirect

7	6	5	4	3	2	1	0	
1	1	Nx	1	1	1	0	1	JP (IX) (for Nx = 0)
1	1	1	0	1	0	0	1	JP (IY) (for Nx = 1)

Timing: 8 T states

Addressing Mode: Register Indirect

JP cc, nn

Conditionally jump to program location nn based on testable flag states.

If cc true, No flags affected
PC ← nn;
otherwise continue

7	6	5	4	3	2	1	0
1	1	cc			0	1	0
n (low order byte)							
n (high order byte)							

Timing: 10 T states

Addressing Mode: Direct

JR d

Unconditional jump to program location calculated with respect to the program counter and the displacement d.

PC ← PC + d No flags affected

7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0
d - 2							

Timing: 12 T states

Addressing Mode: PC Relative

JR kk, d

Conditionally jump to program location calculated with respect to the program counter and the displacement d based on limited testable flag states.

If kk true, No flags affected
PC ← PC + d,
otherwise continue

7	6	5	4	3	2	1	0
0	0	1	kk		0	0	0
d-2							

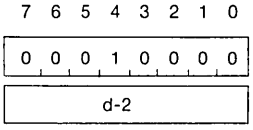
Timing: 12 T states (if kk met)
7 T states (if kk not met)

Addressing Mode: PC Relative

DJNZ, d

Decrement the B register and conditionally jump to program location calculated with respect to the program counter and the displacement d, based on the contents of the B register.

B ← B - 1 No flags affected
If B = 0 continue,
else PC ← PC + d



Timing: 13 T states (If B ≠ 0)
8 T states (If B = 0)

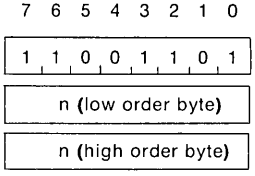
Addressing Mode: PC relative

5.3.11.2 Calls

CALL nn

Unconditional call to subroutine at location nn.

(SP -1) ← PC_H No flags affected
(SP -2) ← PC_L
SP ← SP -2
PC ← nn



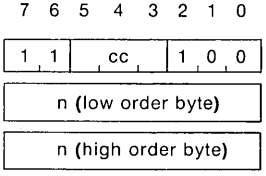
Timing: 17 T states

Addressing Mode: Direct

CALL cc, nn

Conditional call to subroutine at location nn based on testable flag states.

If cc true, No flags affected
(SP -1) ← PC_H
(SP -2) ← PC_L
SP ← SP -2
PC ← nn,
else continue



Timing: 17 T states (If cc true)
10 T states (If cc not true)

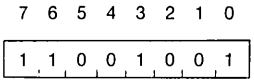
Addressing Mode: Direct

5.3.11.3 Returns

RET

Unconditional return from subroutine or other return to program location pointed to by the top of the stack.

PC_L ← (SP) No Flags affected
PC_H ← (SP + 1)
SP ← SP + 2



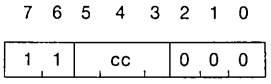
Timing: 10 T states

Addressing Mode: Register Indirect

RET cc

Conditional return from subroutine or other return to program location pointed to by the top of the stack.

If cc true, No flags affected
PC_L ← (SP)
PC_H ← (SP + 1)
SP ← SP + 2; else continue



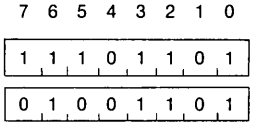
Timing: 11 T states (If cc true)
5 T states (If cc not true)

Addressing Mode: Register Indirect

RETI

Unconditional return from interrupt handling subroutine. Functionally identical to RET instruction. Unique opcode allows external monitoring by peripheral circuit.

PC_L ← (SP) No flags affected
PC_H ← (SP + 1)
SP ← SP + 2



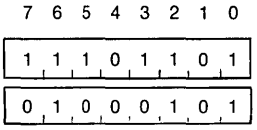
Timing: 14 T states

Addressing Mode: Register Indirect

RETN

Unconditional return from non-maskable interrupt handling subroutine. Functionally similar to RET instruction except interrupt enable state is restored to that prior to non-maskable interrupt.

$PC_L \leftarrow (SP)$ No flags affected
$PC_H \leftarrow (SP + 1)$
$SP \leftarrow SP + 2$
$IFF_1 \leftarrow IFF_2$



Timing: 14 T states

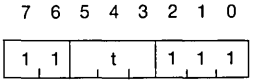
Addressing Mode: Register Indirect

5.3.11.4 Restarts

RST T

The present contents of the PC are pushed onto the memory stack and the PC is loaded with dedicated program locations as determined by the specific restart executed.

$(SP - 1) \leftarrow PC_H$ No flags affected
$(SP - 2) \leftarrow PC_L$
$SP \leftarrow SP - 2$
$PC \leftarrow T$



Timing: 11 T states

Addressing Mode: Modified Page Zero

5.4 COMPARISON TO INS8080A

The NSC800 Instruction Set is downward compatible with the INS8080A. That is, the INS8080A instruction set is a subset of the NSC800 instruction set. Because the software compatibility is on an object code basis, this section lists the INS8080A instruction set and indicates the equivalent NSC800 instruction.

The 8080A instruction set and detailed information is available in National's Series-8000 Microprocessor Family Handbook.

The improvement in the NSC800 instruction set focus on the additional indexed addressing mode that is available for transfer of data to and from memory. These are the instructions such as LD (IX + d), r; LD r, (IX + d); and LD (IY + d), n. In addition, the NSC800 provides for automatic loading of register pairs into consecutive memory locations with single instructions such as LD (nn), BC.

Of course, added NSC800 instructions exist for data manipulation within the unique NSC800 registers, particularly the I and R registers.

5.4.1 Data Transfer Group

Instruction	INS8080A Mnemonic	NSC800 Mnemonic
Move Register	MOV R_1, r_2	LD r_1, r_2
Move from Memory	MOV r, M	LD r, (HL)
Move to Memory	MOV M,r	LD (HL), r
Move Immediate	MVI r, n	LD r, n
Move to Memory Immed.	MVI M, n	LD (HL), n
Load Register Pair Immed.	LXI rr, nn	LD rr, nn
Load Accumulator Direct	LDA nn	LD A, (nn)
Store Accumulator Direct	STA nn	LD (nn), A
Load H and L Direct	LHLD nn	LD HL, (nn)
Store H and L Direct	SHLD nn	LD (nn), HL
Load Accumulator Indirect	LDAX B LDAX D	LD A, (BC) LD A, (DE)
Store Accumulator Indirect	STAX B STAX D	LD (BC), A LD (DE), A
Exchange HL and DE	XCHG	EX DE, HL

5.4.2 Arithmetic Group

Instruction	INS8080AMnemonic	NSC800 Mnemonic
Add Register	ADD r	ADD A,r
Add Memory	ADD M	ADD A, (HL)
Add Immediate	ADI n	ADD A, n
Add Register with Carry	ADC r	ADC A, r
Add Memory with Carry	ADC M	ADC A, (HL)
Add Immediate with Carry	ACI n	ADC A, n
Subtract Register	SUB r	SUB r
Subtract Memory	SUB M	SUB (HL)
Subtract Immediate	SUI n	SUB n
Subtract Register with Borrow	SBB r	SBC A, r
Subtract Memory with Borrow	SBB M	SBC A, (HL)
Subtract Immediate with Borrow	SBI n	SBC A, n
Increment Register	INR r	INC r
Increment Memory	INR M	INC (HL)
Increment Register Pair	INX rr	INC rr
Decrement Register	DCR r	DEC r
Decrement Memory	DCR M	DEC (HL)
Decrement Register Pair	DCX rr	DEC rr
Add Register Pair to HL	DAD rr	ADD HL, rr
Decimal Adjust Accumulator	DAA	DAA

5.4.3 Logical Group

Instruction	INS8080A Mnemonic	NSC800 Mnemonic
AND Register	ANA r	AND r
AND Memory	ANA M	AND (HL)
AND Immediate	ANI n	AND n
Exclusive OR Register	XRA r	XOR r
Exclusive OR Memory	XRA M	XOR (HL)
Exclusive OR Immediate	XRI n	XOR n
OR Register	ORA r	OR r
OR Memory	ORA M	OR (HL)
OR Immediate	ORI n	OR n
Compare Register	CMP r	CP r
Compare Memory	CMP M	CP (HL)
Compare Immediate	CPI n	CP n
Rotate Left Thru Carry	RAL	RLA
Rotate Right Thru Carry	RAR	RRA
Complement Accumulator	CMA	CPL
Complement Carry	CMC	CCF
Set Carry	STC	SCF
Rotate Left	RLC	RLCA
Rotate Right	RRC	RRCA

5.4.4 Branch Group

Instruction	INS8080A Mnemonic	NSC800 Mnemonic
Unconditional Jump	JMP nn	JP nn
Conditional Jump	JZ nn	JP Z, nn
	JNZ nn	JP NZ, nn
	JC nn	JP C, nn
	JNC nn	JP NC, nn
	JPO nn	JP PO, nn
	JPE nn	JP PE, nn
	JP nn	JP P, nn
	JM nn	JP M, nn
Unconditional Call	CALL nn	CALL nn
Conditional CALL	CZ nn	CALL Z, nn
	CNZ nn	CALL NZ, nn
	CC nn	CALL C, nn
	CNC nn	CALL NC, nn
	CPO nn	CALL PO, nn
	CPE nn	CALL PE, nn
	CP nn	CALL P, nn
	CM nn	CALL M, nn
Unconditional Return	RET	RET
Conditional Return	RZ	RET Z
	RNZ	RET NZ
	RC	RET C
	RNC	RET NC
	RPO	RET PO
	RPE	RET PE
	RP	RET P
	RM	RET M
Restart	RST t	RST 8 x t
Load PC from HL	PCHL	JP (HL)

5.4.5 Stack, I/O, and Machine Control Group

Instruction	INS8080A Mnemonic	NSC800 Mnemonic
PUSH Register Pair	PUSH qq	PUSH qq
PUSH Processor Status Word	PUSH PSW	PUSH AF
POP Register Pair	POP qq	POP qq
POP Processor Status Word	POP PSW	POP AF
Exchange Top of Stack with HL	XTHL	EX (SP), HL
Move HL to SP	SPHL	LD SP, HL
Enable Interrupts	EI	EI
Disable Interrupts	DI	DI
Input	IN n	IN A, (n)
Output	OUT n	OUT (n), A
HALT	HLT	HALT
No Operation	NOP	NOP

5.4.6 Flag Manipulation

There are minor variations in the affects of the various instructions on the flags of the NSC800 and the INS8080A. Listed below are the differences between flag action on the two processors.

- Bits 3 and 5 are undefined in the NSC800; they are defined as zero in the INS8080A.
- Bit 2 in the NSC800 is a combined parity/overflow flag as defined in paragraph 4.4.2.3. It reflects parity only on the INS8080A. Therefore, on all 8-bit arithmetic options the NSC800 P/V flag will not always match the INS8080A P flag.
- On double precision adds (DAD), the INS8080A does not affect the auxilliary carry flag, AC. On the NSC800, the corresponding flag, the half carry (H), is left undefined.
- On rotate instructions, the INS8080A does not affect the AC flag while the NSC800 resets the H flag.
- When the Accumulator is complemented the AC is unaffected on the INS8080A; the NSC800 sets the H flag.

NSC800 Flags:

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

INS8080A Flags:

7	6	5	4	3	2	1	0
S	Z	O	AC	O	P	I	CY

- On the NSC800, bit 1 is defined as an add/subtract flag (N) as described in paragraph 4.4.2.2. It is defined as a 1 on the INS8080A.
- When the carry flag is complemented or set via the applicable instruction, the INS8080A leaves the AC flag unaffected while the NSC800's H flag is the previous carry with a complement carry, and reset with a set carry.
- During general comparison instructions the INS8080A P flag is affected according to the parity of the implied subtraction, while the P/V flag on the NSC800 acts as an overflow.

Chapter 6

NSC810 RAM-I/O-Timer

6.1 INTRODUCTION

The NSC810 RAM-I/O-Timer is a supporting device to the NSC800 CPU. The NSC810 is a high-performance, low-power, dedicated memory peripheral that provides RAM memory; flexible, programmable I/O capabilities; and two programmable timers.

6.1.1 Architecture

A detailed block diagram of the NSC810 architecture is shown in *Figure 6-1*.

6.1.2 RAM-I/O-Timer Functions

The following paragraphs detail the NSC810 functions as they appear to the external system. By understanding these functions, the designer can design efficient systems that meet his specific application requirements. The pin configuration is shown in *Figure 6-2* and the functional description for each pin is listed in *Table 6-1*.

6.2 NSC810 INTERFACE

The NSC810 is designed to interface directly with the NSC800 CPU bus structure. Under control of the CPU, the NSC810 is enabled by a logic 1 (high) input on the chip enable (CE) pin. As shown in *Figure 6-1*, the NSC810 contains address/data bus latches that latch the address input/output at the falling edge of the address latch enable (ALE) input signal. Data is then directed according to the other control signals: input/output timer or RAM select (IOT/$\overline{M}$), read ($\overline{RD}$) or write ($\overline{WR}$). Six control signals effect the functions of the NSC810. These control signals are furnished by the NSC800 CPU.

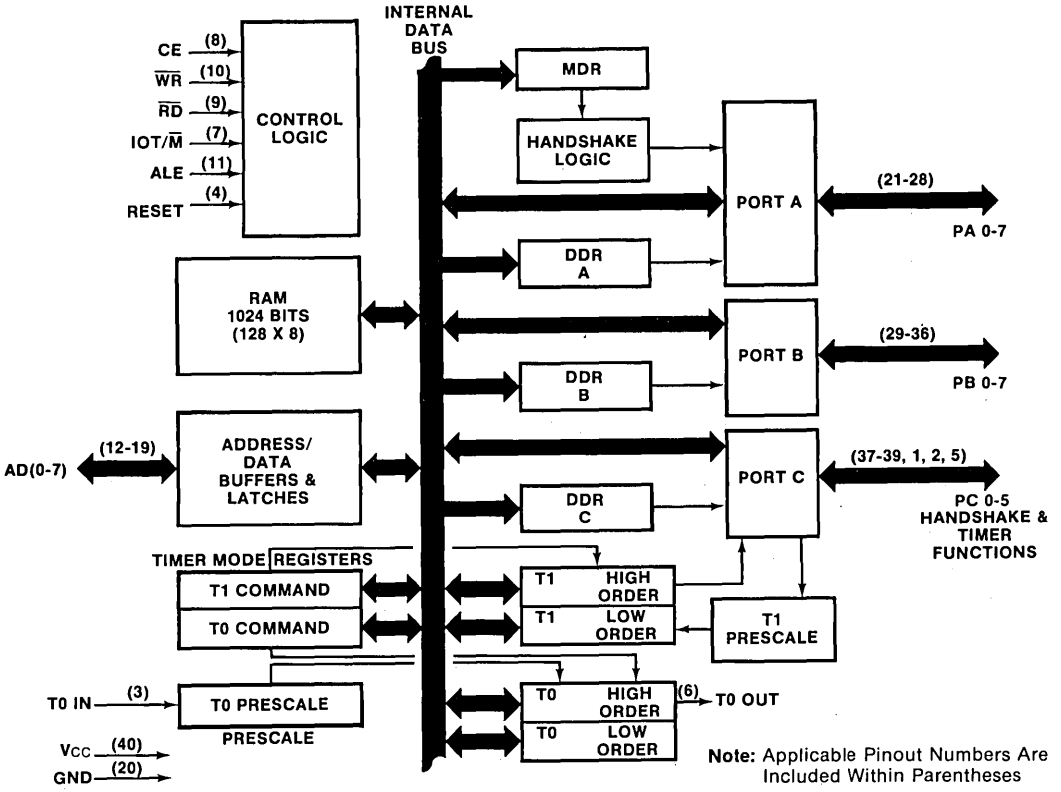


FIGURE 6-1. NSC810 Detailed Block Diagram 20-7

6.2.1 Reset

The reset (RESET) input to the NSC810 will, when activated, cause the device to reset with the existing system. A logic 1 (high) input on the RESET pin causes the NSC810 to initialize and results in the following:

a. All internal registers are written with zeros.

b. The counter/timers are stopped and reset to the initialized condition.

c. The input/output ports (Port A, B and C) revert to the high impedance input mode.

d. The contents of the RAM remain unaltered.

6.2.2 Chip Enable (CE)

The CE input determines whether or not the CPU is going to use the NSC810 for data manipulation. A logic 1 (high) input to the CE pin enables the NSC810; a logic 0 (low) input will disable the NSC810. Although no modifications

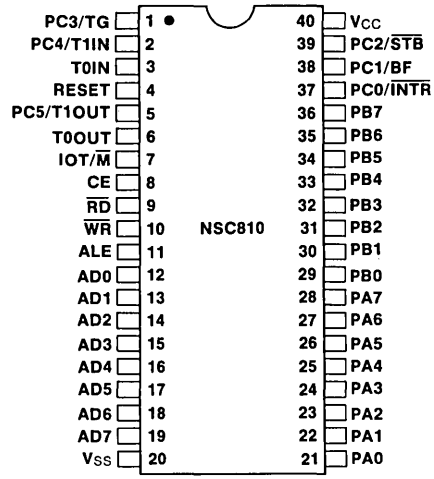


FIGURE 6-2. NSC810 Pin Configuration

20-6

TABLE 6-1. NSC810 Pin Descriptions

INPUT SIGNALS

Reset (RESET): RESET is an active-high input that resets all registers to 0 (low). The RAM contents remain unaltered.

Input/Output Timer or RAM Select (IOT/$\overline{M}$): IOT/$\overline{M}$ is an I/O memory select input line. A logic 1 (high) input selects the I/O-Timer portion of the chip; a logic 0 (low) input selects the RAM portion of the chip. IOT/$\overline{M}$ is latched at the falling edge of ALE.

Chip Enable (CE): CE is an active-high input that allows access to the NSC810. CE is latched at the falling edge of ALE.

Read ($\overline{RD}$): The $\overline{RD}$ is an active-low input that enables a read operation of the RAM or I/O-Timer location.

Write ($\overline{WR}$): The $\overline{WR}$ is an active-low input that enables a write operation to RAM or I/O-Timer locations.

Address Latch Enable (ALE): The falling edge of the ALE input latches AD0-AD7, CE and IOT/$\overline{M}$ inputs to form the address for RAM, I/O or timer.

Timer 0 Input (T0IN): T0IN is the clock input for timer 0.

OUTPUT SIGNALS

Timer 0 Output (T0OUT): T0OUT is the programmable output of timer 0.

POWER SUPPLY PINS

Positive DC Voltage (V_{CC}): V_{CC} is the 3-12 volt supply pin.

Ground (V_{SS}): V_{SS} is the ground reference pin.

INPUT/OUTPUT SIGNALS

Address/Data Bus (AD0-AD7): The multiplexed bidirectional address/data bus, AD0-AD7 pins, are in the high impedance state when the NSC810 is not selected. AD0-AD7 will latch address inputs at the falling edge of ALE. The address will designate a location in RAM, I/O or timer. $\overline{WR}$ input enables 8-bit data to be written into the addressed location. $\overline{RD}$ input enables 8-bit data to be read from the addressed location. The $\overline{RD}$ or $\overline{WR}$ inputs occur while ALE is low .

Port A, 0-7 (PA0-PA7): Port A is an 8-bit basic mode input/output port, also capable of strobed mode I/O utilizing three control signals from Port C. Port A strobed mode outputs can be active or any of the three states of the TRI-STATE output.

Port B, 0-7 (PB0-PB7): Port B is an 8-bit basic mode input/output port (see *Figure 3* Port functions).

Port C, 0-5 (PC0-PC5): Port C is a 6-bit basic mode I/O port. Each pin has a programmable second function, as follows:

PC0/$\overline{INTR}$: $\overline{INTR}$ is an active-low strobed mode interrupt request to the Central Processor Unit (CPU).

PC1/BF: BF is an active-high buffer full output to peripheral devices.

PC2/$\overline{STB}$: $\overline{STB}$ is an active-low strobe input from peripheral devices.

PC3/TG: TG is the timer gating signal.

PC4/T1 IN: T1IN is the clock input for timer 1.

PC5/T1 OUT: T1OUT is the programmable output of timer 1.

of timer and port registers may be made while CE is inactive, they continue to function as specified.

6.2.3 Address Latch Enable (ALE)

The high-to-low transition (falling edge) of the ALE input to the NSC810 latches the address from the AD(0-7) bidirectional bus. When combined with the CE and IOT/$\overline{M}$ inputs, the information on AD(0-7) forms the address for the RAM, I/O, or Timer.

6.2.4 Input/Output Timer or RAM Select (IOT/$\overline{M}$)

The IOT/$\overline{M}$ input select line determines which portion of the NSC810 is to perform the manipulations requested by the CPU. A logic 1 (high) on the IOT/$\overline{M}$ input selects the input/output-timer portion of the device. A logic 0 (low) on the IOT/$\overline{M}$ input selects the RAM.

6.2.5 Read Signal ($\overline{RD}$)

The $\overline{RD}$ is an active-low input that enables a read operation of the RAM or I/O-Timer at the preselected address location. The high-to-low transition (falling edge of the $\overline{RD}$ input allows the NSC810 to drive onto the AD(0-7) bidirectional bus.

6.2.6 Write Signal ($\overline{WR}$)

The $\overline{WR}$ is an active-low input that enables a write operation to RAM or I/O-Timer at the preselected address location. The low-to-high transition (rising edge) of the $\overline{WR}$ input latches the data from the AD(0-7) bidirectional bus.

6.3 INPUT/OUTPUT (I/O) FUNCTIONS

This paragraph discusses the I/O functions of the NSC810. The input/output capabilities of the NSC810 provide the user with a programmable means of communicating with peripherals in the system. The basic configuration consists of three ports, two eight-bit ports (A and B) and one six-bit port (C).

6.3.1 Standard I/O Operation

6.3.1.1 Port Addressing. Data is read from and written to each port in parallel. Each port is assigned an address in the I/O space of the NSC810. The addresses assigned are as follows:

Port A	X'00
Port B	X'01
Port C	X'02

The I/O portion is selected by a logic 1 (high) on the IOT/$\overline{M}$ pin on the NSC810. In an NSC800 based system this pin can be hard wired to the IO/$\overline{M}$ control signal from the CPU and all the NSC810 registers can then be accessed via the set of NSC800 I/O instructions (see paragraph 5.2). For memory-mapped systems, by tying this pin on the NSC810 to a high-order address bit, the I/O portion of the NSC810 will be accessed by addressing a memory location. This allows addressing of the I/O and timing registers as well as the RAM with any of the NSC800's memory reference instructions.

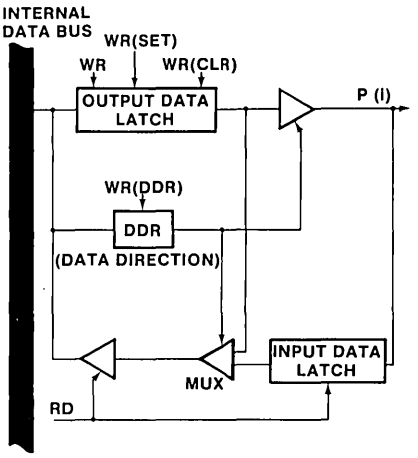


FIGURE 6-3. Block Diagram for Port Bit (I)

20-8

6.3.1.2 Data Direction. The direction of the data in each port (input or output) is defined by the Data Direction Registers (DDR). Each port has a DDR. By writing a pattern of data in the DDR, the port bit position corresponding to the DDR bit position is defined as an input if a "0" is written in the DDR, or an output if a "1" is written in the DDR. When the DDR for Port A is written with the data X'A5 (10100101), this results in PA0, PA2, PA5, and PA7 being defined as outputs, while PA1, PA3, PA4, and PA6 are defined as inputs. The DDR bits cannot be written to individually; the entire DDR byte is affected by a write to the DDR address. Thus, all data must be consistent with the direction desired for each port.

Any write or read operations on a port contradicting the DDR will not affect the port output or input. However, a read from a port bit, defined as an output, will cause a read from the ouput latch, and a write to a port bit, defined as an input, will modify the ouput latch. Refer to *Figure 6-3.*

The addresses of the data direction registers are as follows:

Port A DDR	X'04
Port B DDR	X'05
Port C DDR	X'06

The NSC810 provides loss of data protection from operations that are contrary to the defined data direction. For example, a read of a bit defined as an output will read the output latch, returning the data previously written. Similarly, a write to a bit defined as an input will modify the output latch and not alter the input data.

6.3.1.3 Bit Set/Bit Clear. The NSC810 allows the user to selectively set or clear a single bit or group of bits in any of the ports without having to rewrite the entire port in parallel. This is accomplished by writing particular addresses associated with the set or clear functions. The data written to these addresses causes the corresponding

port bit (or bits) to be set or cleared. The set and clear addresses are as follows:

Port A	Bit Clear	X'08
Port B	Bit Clear	X'09
Port C	Bit Clear	X'0A
Port A	Bit Set	X'0C
Port B	Bit Set	X'0D
Port C	Bit Set	X'0E

For example, if a data pattern of X'A0 (= 1010000) were written to I/O register X'0D (Port B, Bit Set), the results are the setting of PB7 and PB5 with all other port bits unaffected. If the data X'0C (= 00001100) were written to address X'0A (Port C, Bit Clear) the results are PC3 and PC2 reset to zero, and all other Port C bits remain unchanged.

6.3.1.4 Mode Definition Register (MDR). The Mode Definition Register (MDR) defines the operating mode for Port A. While Ports B and C are always in the basic I/O mode, there are four operating modes for Port A:

Mode 0 - Basic I/O (Input or Output)
Mode 1 - Strobed Mode Input
Mode 2 - Strobed Mode Ouput — Active Peripheral Bus
Mode 3 - Strobed Mode Output — TRI-STATE Peripheral Bus

The MDR has the I/O address assignment X'07 and is illustrated for the four modes in *Table 6-2.*

6.3.1.5 Port Functions - Basic I/O. Basic I/O is the mode of operation of Ports B and C and Mode 0 of Port A (defined by the MDR). Read and write byte operations, and bit operations can be executed in the basic I/O mode. The timing for basic input and basic ouput modes is shown in *Figures 6-4* and *6-5.* The AC characteristics for input and output timings are listed in the NSC810 Data Sheet in Appendix A.

When a read occurs, the information is latched from the peripheral (port) bus during the leading (falling) edge of the $\overline{RD}$ strobe. When a write occurs, the port bus is modified after the trailing (rising) edge of the $\overline{WR}$ strobe with data from the AD bus. Port output data remains valid at the ouput pin from one trailing edge of $\overline{WR}$ strobe, to the trailing edge of the next $\overline{WR}$ strobe which then modifies that port.

Table 6-2. Mode Definition Register Bit Assignments

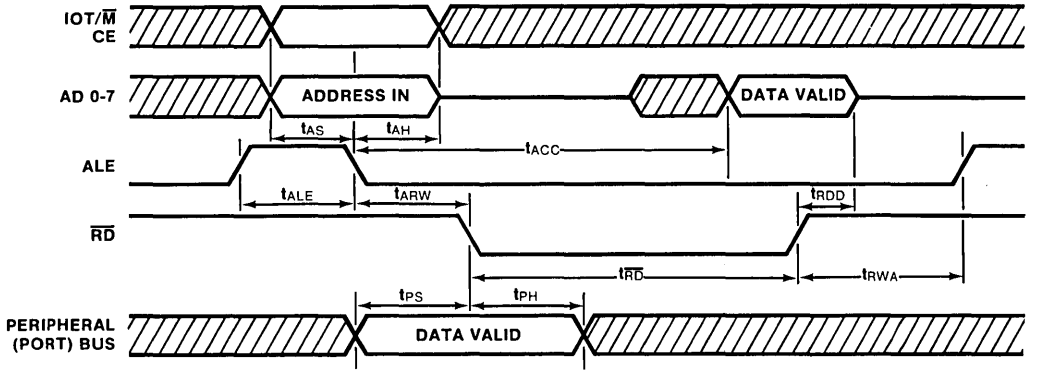
MDR CONTENTS								MODE
7	6	5	4	3	2	1	0	
x	x	x	x	x	x	x	0	Basic I/O (non strobed)
x	x	x	x	x	x	0	1	Strobed Input
x	x	x	x	x	0	1	1	Strobed Output (active peripheral bus)
x	x	x	x	x	1	1	1	Strobed Output (TRI-STATE peripheral bus)

6.3.2 Strobed Mode of Operation

The NSC810 provides strobed mode communication via Port A. By programming the mode definition register (MDR, address X'07), one of three strobed mode operations can be evoked, in addition to the basic standard I/O mode previously described.

The strobed modes allow handshake interfacing to asynchronous external devices. Handshake signals are provided by Port C (pins 0, 1, and 2) to facilitate interfacing between an external device and the CPU.

The output data latch of Port A is written to with the trailing edge of the $\overline{WR}$ strobe (see *Figure 6-6*). If the MDR defines Port A to be in the strobed output mode, active bus, the contents of the ouput data latch fall through onto the output pin. If the MDR specifies TRI-STATE bus, the output data latch performs in a similar manner as the active bus except the output pin remains at high-impedance until the $\overline{STB}$ signal becomes valid. The



NOTE: Diagonal lines indicate interval of invalid data.

FIGURE 6-4. Read From RAM, Port or Timer

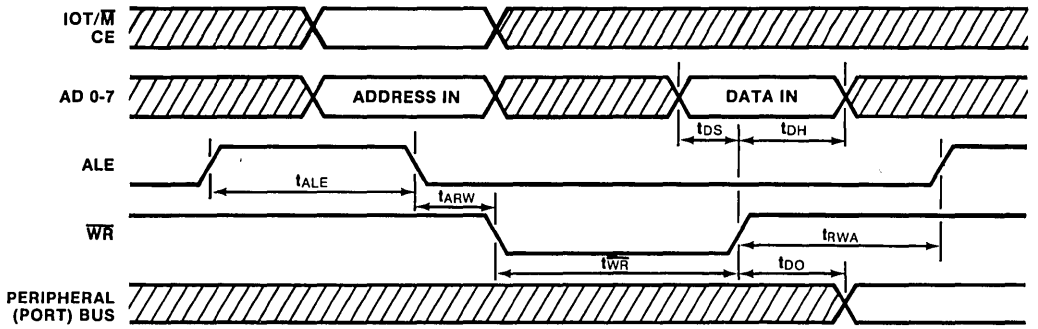
20-1

contents of the output data latch is then transmitted to the port output pin.

6.3.3 Handshake Signals. In the strobed modes of operation, three signals provide the proper interfacing between the CPU and an external device. These signals are:

1. A $\overline{STB}$ strobe to signify when the external device has loaded data into Port A, or taken data from Port A
2. A BF signal to indicate when valid data is present
3. An $\overline{INTR}$ signal to notify the CPU when action on its part is required (an interrupt)

The handshake signals are multiplexed with bits 0 through 2 of Port C, (i.e., PC0, PC1, and PC2). When Port A is in the basic I/O mode of operation (MDR bit 0 = 0), these pins operate as standard Port C bits, as described in 6.3:1. When any strobed mode is selected (MDR bit 0 = 1), these pins take on their handshake functions.



NOTE: Diagonal lines indicate interval of invalid data.

FIGURE 6-5. Write to RAM, Port or Timer 20-2

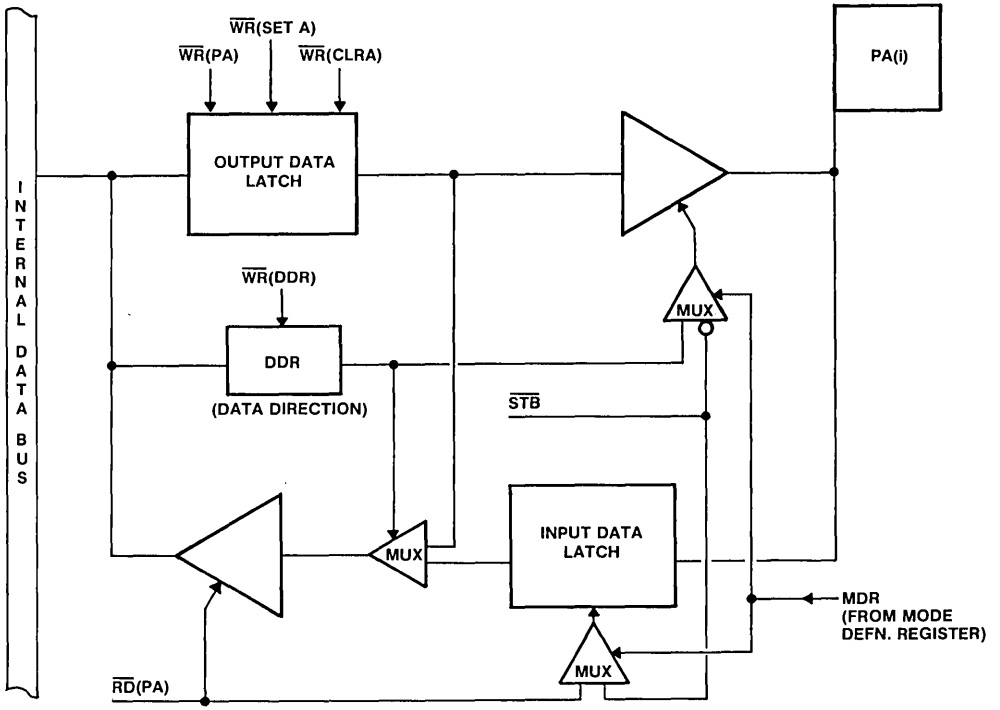


FIGURE 6-6. NSC810 Port A Block Diagram 72-93

The Port C DDR for the bits used for the handshake signals must be initialized to the proper direction definition for the function performed, as follows:

Port C	Handshake Function	Direction
PC0	Interrupt ($\overline{INTR}$)	Output
PC1	Buffer Full (BF)	Output
PC2	Strobe ($\overline{STB}$)	Input

The Port C data direction register (address X'06) should be written with

5	4	3	2	1	0
x	x	x	0	1	1

In addition, the data direction register for Port A must be consistent with the data direction selected by the MDR.

6.3.3.1 Interrupt/PC0 The interrupt handshake signal $\overline{INTR}$ is multiplexed on pin PC0. $\overline{INTR}$ is an active low interrupt signal to the CPU. The $\overline{INTR}$ is true (active low) when the CPU is requested to take some action. In the strobed input mode, this indicates data is available to be read. In the strobed output mode, the $\overline{INTR}$ signifies data is being requested by the external device. The CPU places the requested data on Port A.

By setting or clearing the output latch PC2, the interrupt can be enabled or disabled.

6.3.3.2 Buffer Full/PC1 The buffer full handshake signal BF is multiplexed on pin PC1. BF indicates that the Port A buffer contains valid data. In the input mode, BF signifies that the external device has loaded data and the CPU has not yet retrieved the data. In output mode, BF signifies that the CPU has written to Port A and the external device has not accepted the data.

6.3.3.3 Strobed/PC2. PC2 shares a pin with the handshake strobe, $\overline{STB}$. $\overline{STB}$ is an active low signal that is generated by the external device that is interfacing to the NSC810. It is used to asynchronously signal when the device performs a transfer to or from Port A.

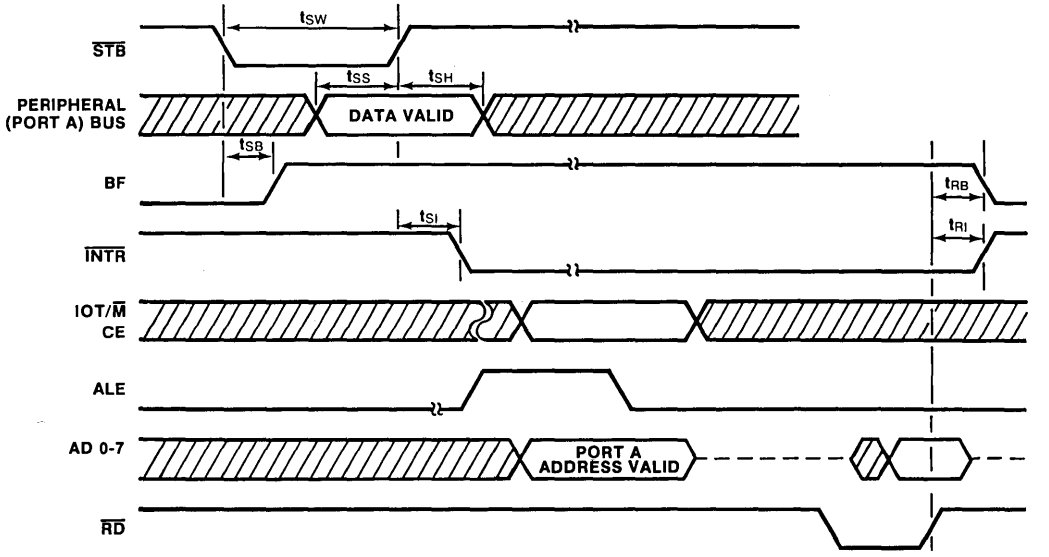
6.3.4 Strobed Input

During strobed input operations, an external device can load data into Port A with the $\overline{STB}$ signal. Data is input to the PA 0-7 input latches on the leading (negative) edge of $\overline{STB}$ (see *Figure 6-7*), causing BF to go high (true). On the trailing (positive) edge of $\overline{STB}$ the interrupt signal, $\overline{INTR}$, becomes valid indicating to the CPU that data is available for reading. $\overline{INTR}$ will become valid only if the interrupt is enabled, that is the output data latch for PC2 is true.

When the CPU reads Port A, address X'00, the trailing edge of the $\overline{RD}$ strobe causes BF and $\overline{INTR}$ to become inactive, indicating that the strobed input cycle was completed.

6.3.5 Strobed Output

During strobed output operations, an external device can read data from Port A with the $\overline{STB}$ signal (see *Figure 6-8*). Data is initially loaded into Port A by the CPU writing to I/O address X'00. On the trailing edge of $\overline{WR}$, $\overline{INTR}$ is set inactive and BF becomes valid indicating data is available for the external device. When the external device is ready



NOTE: Diagonal lines indicate interval of invalid data.

FIGURE 6-7. Strobed Mode Input

20-3

to accept the data in Port A it pulses the $\overline{STB}$ signal. $\overline{STB}$ will reset BF with its rising edge and also activates the $\overline{INTR}$ signal.

$\overline{INTR}$ in this mode indicates a condition that requires CPU intervention, which is the output of the next byte of data.

6.3.6 Strobed Output - TRI-STATE Mode (Mode 3)

The strobed output-TRI-STATE mode and the strobed output-active (peripheral) bus mode function in a similar manner with one exception. The exception is the data signals on PA 0-7 assume the high impedance state at all times except when accessed by the $\overline{STB}$ signal. Thus, in addition to its timing function, $\overline{STB}$ activates Port A outputs to active logic levels. This Mode 3 operation allows other data sources, in addition to the NSC810, to feed a common external device.

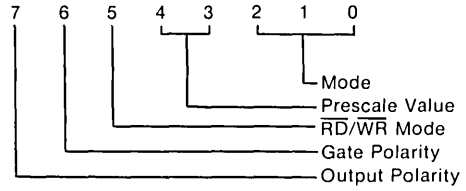
6.4 TIMER

The NSC810 includes two 16-bit programmable timers, T0 and T1. Each timer has an associated command register and prescaler. Note that only timer T0 has dedicated input and output pins.

Timer T1 shares its input and output pins with Port C, bits four and five. The timer gate (TG) also shares its pin with Port C, bit three.

Each timer has a timer mode register (TMR) that selects the mode of operation for the timer. TMR for timer 0 and timer 1 are addressed at I/O addresses X'18 and X'19, respectively.

The bit assignment for the TMR is as follows:



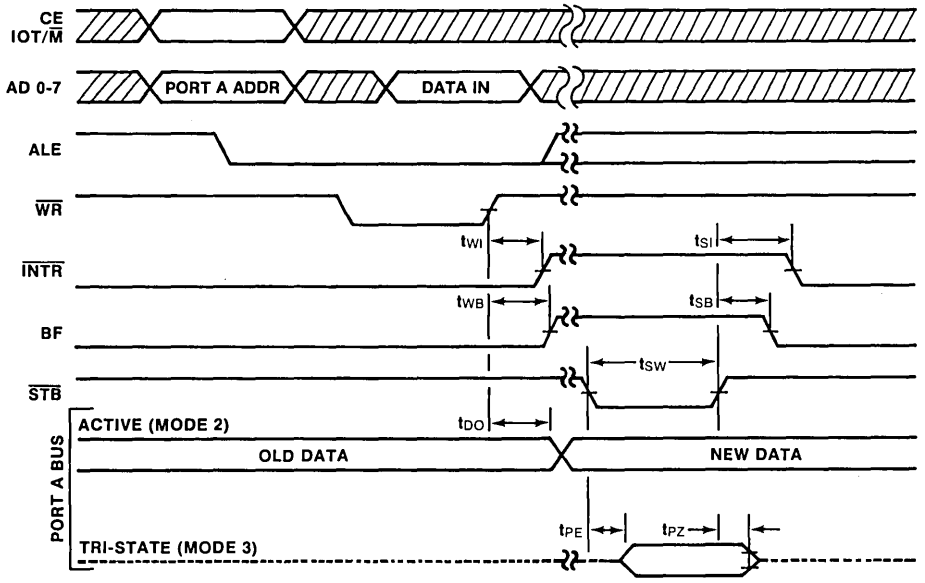
6.4.1 Timer Control Signals

Control for the two NSC810 timers is provided by TIN (timer input), TOUT (timer output), and TG (timer gate). *Figure 6-9* illustrates the NSC810 timer configuration.

6.4.1.1 Timer Input (TIN). The timer clock suplied by the user may have a frequency from DC to 2MHz, or 4MHz if the prescaler has been selected in the mode register (TMR).

6.4.1.2 Timer Gate (TG). The TG input is the hardware control for starting and stopping the timers. For Modes 2 and 3, the timer starts on the gate-active transition assuming the start address was previously written. If the timer gate makes an active transition prior to a write to the start address, the trailing edge of the $\overline{WR}$ strobe starts the timer. However, for Mode 4 the timer always waits for an active gate edge following a write to the start address.

The timer gate can be defined as an active high or active low signal by bit 6 of the TMR. For Mode 4, this is interpreted as defining which transition (high to low or low



NOTE: Diagonal lines indicate interval of invalid data.

FIGURE 6-8. Strobed Mode Output 20-4

to high) is the active edge. The timer gate may be used individually by either timer or simultaneously by both timers.

6.4.2 Timer Modes

The low-order three bits (bits 0, 1, 2) of the Timer Mode Registers (TMR) define the mode of operation for the timers. Each TMR may be written to, or read from, at anytime. However, to ensure accurate timing, it is important to modify the mode of the timer only when the timer is stopped. Inputs of 000 or 111 will define a NOP (no operation) mode, the timer is stopped and the output is inactive. Inputs of 001 through 110 will select one of six distinct timer functions.

In the explanations that follow, assume that the modulus for the timer is loaded with the appropriate value by writing to the low and high bytes of each timer (I/O addresses X'10 and X'11 for timer T0 and X'12 and X'13 for timer T1). Assume also that the timer is started by writing the I/O address X'15 (T0) or X'17 (T1) and the prescaler is not selected. (See paragraph 6.4.3 for prescaler selection.)

6.4.2.1 Event Counter (Mode 1, TMR Bits = 001). In the non-gated mode, the count is decremented for each clock period at the input of the timer (see *Figure 6-10*). When the count reaches zero, the output goes valid and remains valid until the timer count is read by the CPU, or the timer is halted.

The timer is reloaded at the terminal count (= 0) with the modulus and continues to decrement even when the output is valid.

6.4.2.2 Accumulative Timer (Mode 2, TMR Bits = 010). In this gated mode the counter will decrement only when the gate input is active (see *Figure 6-11*). If the gate becomes inactive the counter will hold at its present value and continue to decrement when the gate again becomes active. When the counter decrement is zero, the output becomes valid and remains valid until the count is read by the CPU or the timer is halted.

At the terminal count the timer is reloaded and the count continues as long as the gate is active.

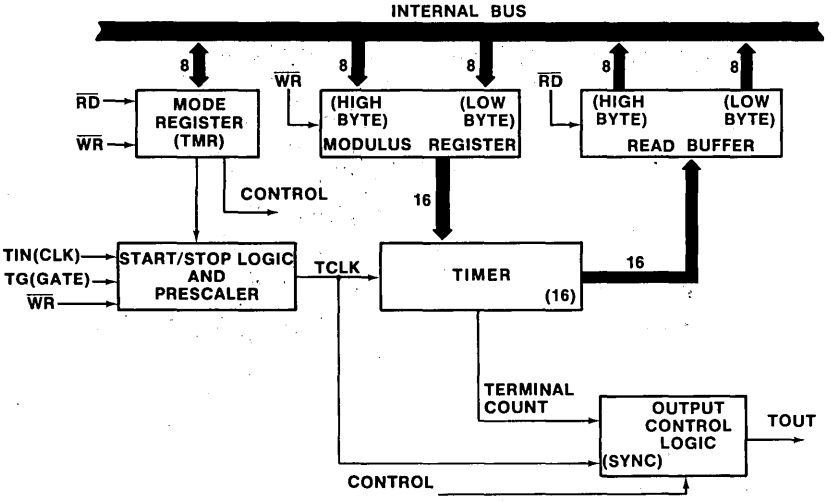


FIGURE 6-9. Timer Internal Block Diagram (One of Two) 20-9

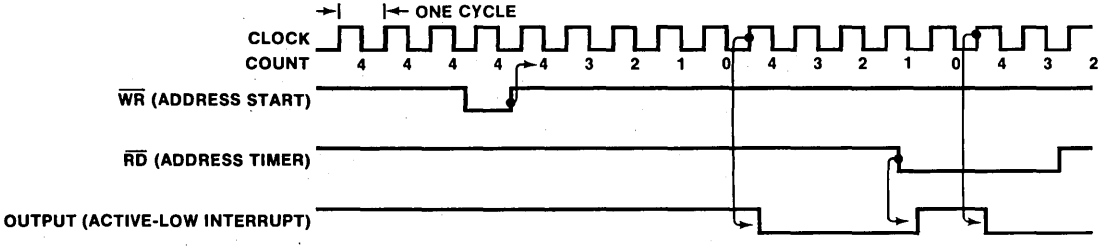


FIGURE 6-10. Mode 1: Counter 20-10

6.4.2.3 Restartable Timer (Mode 3, TMR Bits = 011). In this gated mode the counter will decrement only when the gate input is active. If the gate becomes inactive, the counter will reload the modulus and hold this value until the gate again becomes active (see *Figure 6-12*). The timer restarts at its modulus value. When the counter reaches terminal count, the output will become valid and remain valid until the count is read by the CPU or the timer is halted.

At the terminal count the timer is reloaded and the count continues as long as the gate is active.

6.4.2.4 One Shot (Mode 4, TMR Bits = 100). In this gated mode the timer holds the modulus count until the active gate edge (see *Figure 6-13*). The output immediately becomes valid and remains valid as the counter decrements. The gating signal may go inactive without affecting the count. If TG (the gate) becomes inactive and returns active prior to the terminal count, the modulus will be reloaded, retriggering the one shot period. When the timer reaches the terminal count, the output becomes inactive. The gate, in this mode, is edge sensitive; the active edge is defined in the TMR (see 6.4.5)

6.4.2.5 Square Wave (Mode 5, TMR Bits = 101). In this non-gated mode, the counter decrements for each clock period and complements its output when zero is reached (see *Figure 6-14*). The modulus is then reloaded and counting continues. Assuming a regular clock input, the output will then be a square wave with a period equal to twice the value loaded into the modulus.

6.4.2.6 Pulse Generator (Mode 6, TMR Bits = 110). In this non-gated mode the counter decrements for each clock period (see *Figure 6-15*). When the timer decrements to zero, the output becomes valid for one clock width. The modulus is reloaded and the sequence repeated.

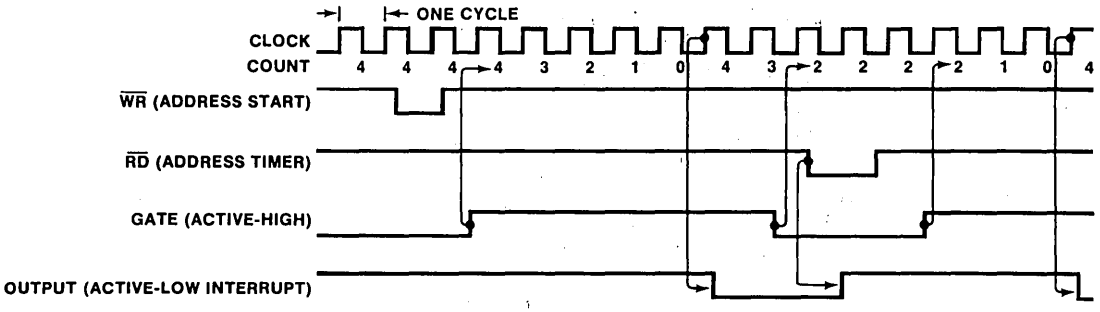


FIGURE 6-11. Mode 2: Timer 20-11

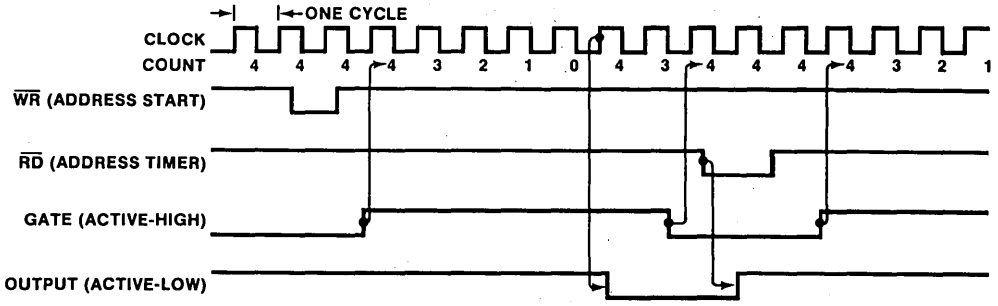


FIGURE 6-12. Mode 3: Timer 20-12

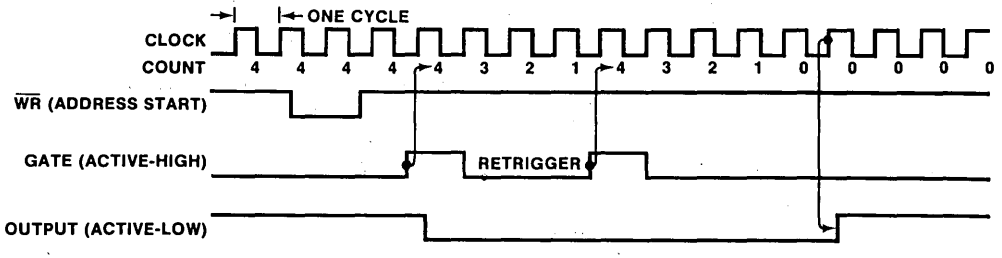


FIGURE 6-13. Mode 4: One Shot 20-13

6.4.3 Timer Prescaler

There is a prescale function associated with each timer. It serves as an additional divisor to lengthen the counts for each timer circuit. The value of the divisor is fixed and selectable in each TMR.

The timer output is affected by the prescale selection. The output responds to the timer clock, not the incoming clock (TIN); so, TOUT will be prescaled by the same value as the timer. Although the 16-bit prescaled count of the timer may be read, the internal value of the prescaler cannot be read by the user. A "00" for either timer represents ÷ 1 (no prescale). Timer 0 has the two possibilities of ÷ 2 or ÷ 64:

TIMER Bit	4	3	Prescale
	0	0	÷ 1
	0	1	÷ 2
	1	x	÷ 64

Timer 1 has only the ÷ 2 prescale available; TMR bit 4 is a "don't care".

TIMER Bit	4	3	Prescale
	x	0	÷ 1
	x	1	÷ 2

6.4.4 Timer Read/Write Mode

A two-byte word (or a single byte when one byte is a "don't care") may be read from or written to the timers. To program the timer buffers, TMR bit 5 must be set as follows:

0 - Double byte read or write low byte first, then high byte (see note)

1 - Single byte read or write low byte only - high byte "don't care" or high byte only with low byte "don't care".

NOTE: *The order of low byte first, high byte second must be maintained for proper Read/Write communications.*

6.4.5 Gate Input Polarity

The polarity of the gate input may be selected by the contents of bit 6 of the TMR. If bit 6 equals 0, the gate signal will be active high; if bit 6 equals 1, the gate polarity will be active low.

In order to use either timer in a gated mode (e.g., Accumulative Timer, Restartable Timer, or One Shot), it is necessary to use Port C, bit 3 as the gating signal, TG. Both timers share a common TG if they are programmed in a gated mode.

6.4.6 Output Polarity

Like the gating function, the polarity of the output signal is also programmable via bit 7 of the TMR. A zero will cause an active low output; a one will generate an active high output.

The output for T1 is multiplexed with Port C, bit 5. (Similarly T1IN is multiplexed with Port C, bit 4). When any timer mode other than 0 or 7 is specified for T1, or when Mode 2, Mode 3, or Mode 4 is specified for T0, the three Port C pins, bit 3, bit 4, and bit 5, become TG, T1IN and T1OUT, respectively.

6.5 PROGRAMMING

The various register addresses and bit assignments in the NSC810 are summarized in *Tables 6-3, 6-4 and 6-5.*

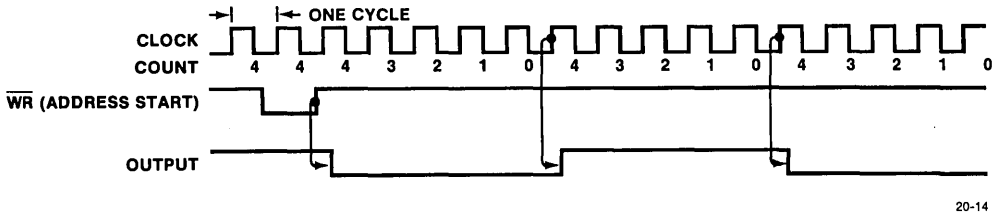


FIGURE 6-14. Mode 5: Square Wave

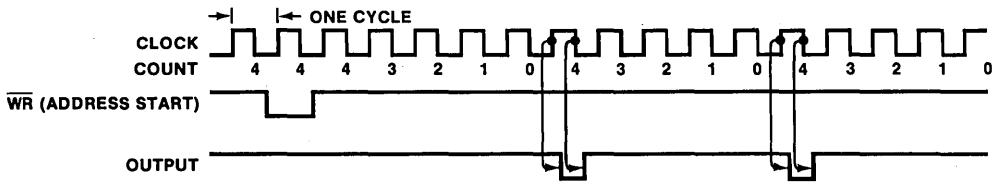


FIGURE 6-15. Mode 6: Pulse Generator

TABLE 6-3. I/O Port and Timer Addressing

I/O Address A_4	A_3	A_2	A_1	A_0	FUNCTION
0	0	0	0	0	READ OR WRITE I/O PORT A
0	0	0	0	1	READ OR WRITE I/O PORT B
0	0	0	1	0	READ OR WRITE I/O PORT C
0	0	1	0	0	WRITE DDR A
0	0	1	0	1	WRITE DDR B
0	0	1	1	0	WRITE DDR C
0	0	1	1	1	WRITE MODE DEFINITION REGISTER
0	1	0	0	0	CLEAR BIT PORT A
0	1	0	0	1	CLEAR BIT PORT B
0	1	0	1	0	CLEAR BIT PORT C
0	1	1	0	0	SET BIT PORT A
0	1	1	0	1	SET BIT PORT B
0	1	1	1	0	SET BIT PORT C
1	0	0	0	0	READ OR WRITE T0 LSB
1	0	0	0	1	READ OR WRITE T0 MSB
1	0	0	1	0	READ OR WRITE T1 LSB
1	0	0	1	1	READ OR WRITE T1 MSB
1	0	1	0	0	STOP T0
1	0	1	0	1	START T0
1	0	1	1	0	STOP T1
1	0	1	1	1	START T1
1	1	0	0	0	READ OR WRITE T0 COMMAND REGISTER
1	1	0	0	1	READ OR WRITE T1 COMMAND REGISTER

Table 6-4. Timer Mode Selection

BIT 2	1	0	Timer Function	Modes
0	0	1	Event Counter	1
0	1	0	Event Timer (Stop watch)	2
0	1	1	Event Timer (Resetting)	3
1	0	0	One Shot	4
1	0	1	Square Wave	5
1	1	0	Pulse Generator	6

NOTE: 000 and 111 are no op's (timer halted, output inactive).

6.6 NSC810 RAM MEMORY

The NSC810 memory is comprised of 1024 bits of static RAM organized as 128 x 8. The memory portion of the RAM-I/O-Timer is accessed by a 7-bit address input to pins AD0 through AD6. The IOT/$\overline{M}$ input must be low (RAM select) and the CE input must be high at the falling edge of ALE to address the RAM (refer to *Figures 6-4* and *6-5*). Address bit AD7 is a "don't care" for RAM addressing.

Table 6-5. Timer Programming Selection Examples

Mode Register Bit 7 6 5 4 3 2 1 0	Output Sense Active L/H	Timer Gate Polarity Active L/H	Mode Description Read/Write Mode Single/Double Byte	Prescale Value	Timing Mode	Port C DDR 5 4 3 2 1 0
	Bit 7	Bit 6	Bit 5	Bits 3&4	Bits 0-2	
TIMER 0						
x x x x x 0 0 0	x	x	x	x	0	x x x x x x
0 x 0 0 0 0 0 1	L	x	D	÷ 1	1	x x x x x x
1 x 0 1 x 1 1 0	H	x	D	÷ 64	6	x x x x x x
1 0 0 0 1 1 0 0	H	L	D	÷ 2	4	1 0 0 x x x
0 1 1 0 0 0 1 0	L	H	S	÷ 1	2	1 0 0 x x x
TIMER 1						
x x x x x 1 1 1	x	x	x	x	7	x x x x x x
0 x 0 x 0 0 0 1	L	x	D	÷ 1	1	1 0 0 x x x
1 0 1 x 1 1 0 1	H	L	S	÷ 2	5	1 0 0 x x x
0 1 0 x 0 0 1 1	L	H	D	÷ 1	3	1 0 0 x x x

x = "don't care"

Chapter 7

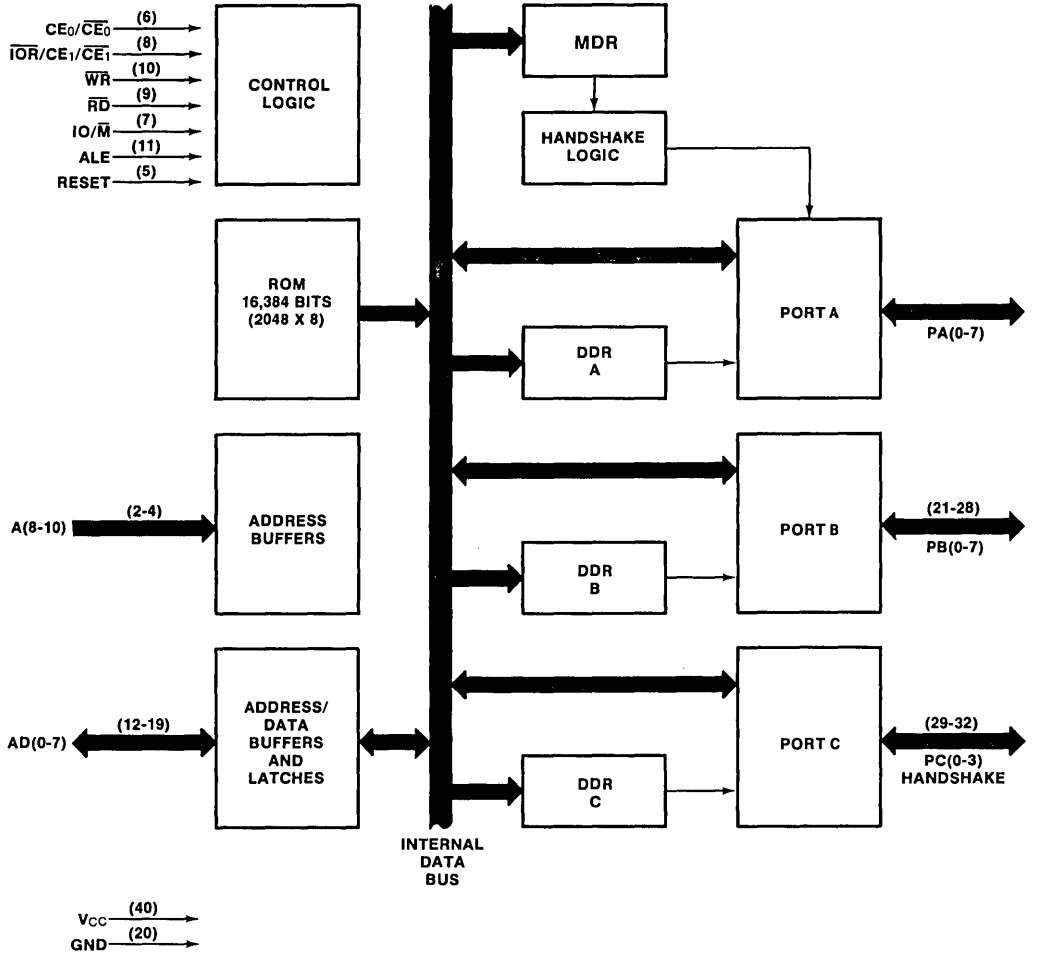
NSC830 ROM-I/O

7.1 INTRODUCTION

The NSC830 ROM-I/O is a supporting device to the NSC800 CPU. The NSC830 is a high-performance, low-power dedicated memory peripheral that provides ROM memory and flexible, programmable I/O capabilities.

7.1.1 Architecture

A detailed block diagram of the NSC830 architecture is shown in *Figure 7-1*.



NOTE: Applicable pinout numbers are included within parentheses.

46-5

FIGURE 7-1. NSC830 Block Diagram

7.1.2 ROM-I/O Functions

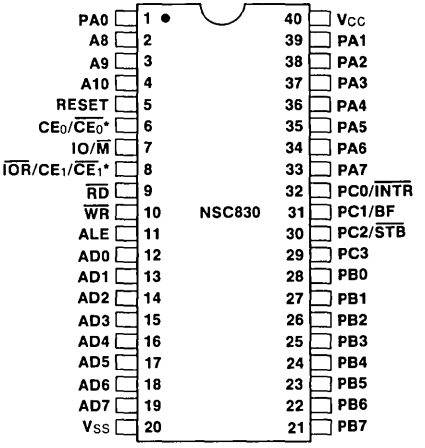
The following paragraphs detail the NSC830 functions as they appear to the external system. By understanding these functions, the designer can design efficient systems that meet his specific application requirements. The pin configuration is shown in *Figure 7-2* and the functional description for each pin is listed in *Table 7-1.*

7.2 INTERFACE

The NSC830 is designed to interface directly to the NSC800 CPU bus structure. All signals required to communicate with the NSC830 are generated by the NSC800. These control signals are ALE, IO/$\overline{M}$, $\overline{RD}$, $\overline{WR}$ and RESET. The NSC830 contains on-chip address latches for demultiplexing the address/data bus of the NSC800. These latches are enabled by the ALE control signal. Latching occurs on the high-to-low transition of ALE. The state of the chip selects and IO/$\overline{M}$ are also latched at this time. Data is then transferred by the $\overline{RD}$ or $\overline{WR}$ strobes as directed by the IO/$\overline{M}$ control signal. Seven control signals effect the functions of the NSC830. These control signals are furnished by the NSC800 CPU.

7.2.1 Reset Signal

The reset (RESET) input to the NSC830 will, when activated, cause the device to reset with the existing system. A logic 1 (high) input on the RESET pin causes the NSC830 to initialize, which results in the input/output ports (Ports A, B and C) reverting to the high impedance mode.



*NOTE: Pin 6 is mask programmable as $\overline{CE_0}$ or $\overline{CE_0}$.
Pin 8 is mask programmable as $\overline{IOR}$, CE_1, or $\overline{CE_1}$.

46-6

FIGURE 7-2. Pin Configuration

TABLE 7-1. NSC830 Pin Descriptions

INPUT SIGNALS

Master Reset (RESET): An active-high input on the RESET pin initializes the chip causing the three I/O ports (A, B and C) to revert to the input mode. The three ports, the three data direction registers and the mode definition register are reset low (0).

Input/Output/Memory Select (IO/$\overline{M}$): The IO/$\overline{M}$ pin is a latched, select input line. A high (1) input selects the I/O portion of the chip; a low (0) input selects the ROM portion of the chip. The select input is latched by the trailing edge (high to low transition) of the ALE signal.

Chip Enable ($CE_0/\overline{CE_0}$, $\overline{IOR}/CE_1/\overline{CE_1}$): The chip enable inputs are mask programmable at the factory. The CE inputs permit the use of multiple NSC830's in a system without using a chip select decoder. The CE inputs must be active at the falling edge of ALE. At ALE time, the CE inputs are latched to provide access to the NSC830. The $\overline{IOR}$ input performs the same function as the combination of IO/$\overline{M}$ input high and the $\overline{RD}$ input low.

Read ($\overline{RD}$): When the $\overline{RD}$ (or the $\overline{IOR}$, when mask programmed) input is an active low, data is read from the AD(0-7)bus. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD (0-7) bus is in the high impedance state.

Write (WR): When the CE inputs are active, and the IO/$\overline{M}$ input is high, an active low $\overline{WR}$ input causes the selected output port to be written with the data from the AD(0-7) bus.

Address Latch Enable (ALE): The trailing edge (high to low transition) of the ALE input signal latches the address/data present on the AD(0-7) bus, A(8-10) bus, plus the input control signals on IO/$\overline{M}$, $CE_0/\overline{CE_0}$, and $CE_1/\overline{CE_1}$.

Address Bus A (8-10): The high-order bits of the ROM address are input on this 3-bit bus and are latched by the high-to-low transition of the ALE input. These bits do not affect the I/O operations.

Power (V_{CC}): 3-12 volt supply.

Ground (V_{SS}): Ground reference.

INPUT/OUTPUT SIGNALS

Bidirectional Address/Data Bus AD (0-7): The lower 8 bits of the ROM or I/O address are applied to these pins, and latched by the trailing edge of ALE. During read operations, 8 bits are present on these pins, and are read when $\overline{RD}$ or IOR is low. During an I/O write cycle, Port A, B, or C is written with the data present on this bus at the trailing edge of the WR strobe.

Ports A, B, C [PA (0-7), PB (0-7), PC (0-3)]: These are general purpose I/O pins. Their input/output direction is determined by the contents of the Data Direction Registers (DDRs)

7.2.2 Chip Enable (CE_0, $\overline{IOR}$, $\overline{CE_1}$)

The two chip enable inputs, (pins 6 and 8 respectively,) provided on the NSC830 are mask programmable at the factory as indicated by the user. They can be either active high or low (when used as chip enables). The state of these control signals is latched at the falling edge of ALE and is held throughout the I/O or ROM cycle. The chip enables must be in the active programmed state to access either the memory or I/O sections of the device. Alternately, $\overline{CE_1}$ can be mask programmed as an I/O Read ($\overline{IOR}$) input. In this condition, $\overline{IOR}$ performs the same enabling function as IO/$\overline{M}$ high and $\overline{RD}$ low.

7.2.3 Address Latch Enable (ALE)

The high-to-low transition (falling edge) of the ALE input to the NSC830 latches the address from the AD(0-7) bidirectional bus. When combined with the CE and IO/$\overline{M}$ inputs, the information on AD(0-7) forms the address for the ROM, I/O.

7.2.4 Input/Output or ROM Select (IO/$\overline{M}$)

The IO/$\overline{M}$ input select line determines which portion of the NSC830 is to perform the manipulations requested by the CPU. A logic 1 (high) on the IO/$\overline{M}$ input selects the input/output portion of the device. A logic 0 (low) on the IO/$\overline{M}$ input selects the ROM.

7.2.5 Read Signal ($\overline{RD}$)

The $\overline{RD}$ is an active-low input that enables a read operation of the ROM or I/O at the preselected address location. The high-to-low transition (falling edge) of the $\overline{RD}$ input allows the NSC830 to drive onto the AD(0-7) bidirectional bus.

7.2.6 Write Signal ($\overline{WR}$)

The $\overline{WR}$ is an active-low input that enables a write operation to the I/O at the preselected address location. The low-to-high transition (rising edge) of the $\overline{WR}$ input latches the data from the AD(0-7) bidirectional bus.

7.3 MEMORY

The 2K bytes of ROM contained in the NSC830 are mask programmable by the factory. The ROM memory of the NSC830 is comprised of 16,384 bits of ROM organized as 2048 x 8. Access to the memory portion is accomplished by having the IO/$\overline{M}$ control signal in the low condition and the chip enables in their active state at the falling edge of ALE. The lower eleven bits of the NSC800 address bus will then select one of the 2048 bytes to be read by the CPU.

The NSC830 is full access-time compatible with the NSC800; therefore, no wait state insertion to hold the processor is necessary.

7.4 INPUT/OUTPUT (I/O) FUNCTIONS

The input/output section of the NSC830 consists of three ports that can be configured in a number of ways. Ports A and B are both 8 bits wide and Port C is 4 bits wide.

Each bit in any of the ports can be configured as either an input or an output. This directional information is set by the data written to the Data Direction Register (DDR). Any combination of inputs and outputs are valid on any of the port bits, except in the special case of Port C when Port A is in the strobed mode of operation.

Ports B and C can operate only in the standard mode, while Port A can operate in the standard mode, plus one of three strobed modes.

7.4.1 Standard I/O Operation

7.4.1.1 Port Addressing. Data is read from and written to each port in parallel. Each port is assigned an address in the I/O space of the NSC830. The addresses assigned are as follows:

Port A	X'00
Port B	X'01
Port C	X'02

The I/O portion is selected by a logic 1 (high) on the IO/$\overline{M}$ pin on the NSC830. In an NSC800 based system this pin can be hard wired to the IO/$\overline{M}$ control signal from the CPU and all the NSC830 registers can then be accessed via the set of NSC800 I/O instructions (see paragraph 5.2). For memory-mapped systems, by tying this pin on the NSC830 to a high-order address bit, the I/O portion of the NSC830 will be accessed by addressing a memory location. This allows addressing of the I/O registers as well as the ROM with any of the NSC800's memory reference instructions.

7.4.1.2 Data Direction. The direction of the data in each port (input or output) is defined by the Data Direction Registers (DDR). Each port has a DDR. By writing a pattern of data in the DDR, the port bit position corresponding to the DDR bit position is defined as an input if a "0" is written in the DDR, or an output if a "1" is written in the DDR. When the DDR for Port A is written with the data X'A5 (10100101) this results in PA0, PA2, PA5, and PA7 being defined as outputs, while PA1, PA3, PA4, and PA6 are defined as inputs. The DDR bits cannot be written to individually; the entire DDR byte is affected by a write to the DDR address. Thus, all data must be consistent with the direction desired for each port.

Any write or read operations on a port contradicting the DDR will not affect the port output or input. However, a read from a port bit, defined as an output, will cause a read from the output latch, and a write to a port bit, defined as an input, will modify the output latch, (see *Figure 7-3.*)

The addresses of the data direction registers are as follows:

Port A DDR	X'04
Port B DDR	X'05
Port C DDR	X'06

The NSC830 provides loss of data protection from operations that are contrary to the defined data direction. For example, a read of a bit defined as an output will read the output latch, returning the data previously written. Similarly, a write to a bit defined as an input will modify the output latch and not alter the input data.

7.4.1.3 Bit Set/Bit Clear. The NSC830 allows the user to selectively set or clear a single bit or group of bits in any of the ports without having to rewrite the entire port in parallel. This is accomplished by writing particular addresses associated with the set or clear functions. The data written to these addresses causes the corresponding port bit (or bits) to be set or cleared. The set and clear addresses are as follows:

Port A	Bit Clear	X'08
Port B	Bit Clear	X'09
Port C	Bit Clear	X'0A
Port A	Bit Set	X'0C
Port B	Bit Set	X'0D
Port C	Bit Set	X'0E

For example, if a data pattern of X'A0 (= 1010000) were written to I/O register X'0D (Port B, Bit Set), the results are the setting of PB7 and PB5 with all other port bits unaffected. If the data X'0C (= 00001100) were written to address X'0A (Port C, Bit Clear) the results are PC3 and PC2 reset to zero, and all other Port C bits remain unchanged.

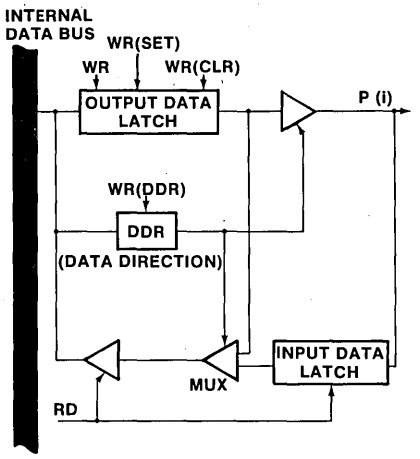


FIGURE 7-3. Block Diagram for Port Bit (i)

20-8

7.4.1.4 Mode Definition Register (MDR). The Mode Definition Register (MDR) defines the operating mode for Port A. While Ports B and C are always in the basic I/O mode, there are four operating modes for Port A:

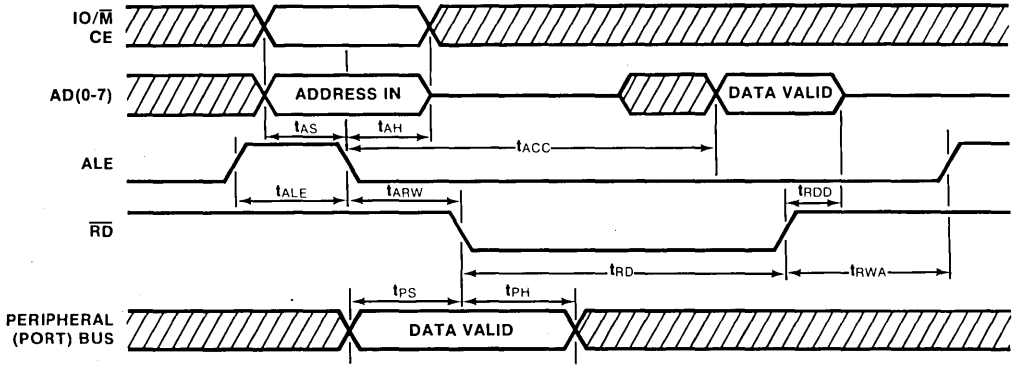
Mode 0 - Basic I/O (Input or Output)
Mode 1 - Strobed Mode Input
Mode 2 - Strobed Mode Output — Active Peripheral Bus
Mode 3 - Strobed Mode Output — TRI-STATE Peripheral Bus

The MDR has the I/O address assignment X'07 and is illustrated for the four modes in *Table 7-2.*

TABLE 7-2. Mode Definition Register Bit Assignments

MDR CONTENTS								MODE
7	6	5	4	3	2	1	0	
x	x	x	x	x	x	x	0	Basic I/O (non strobed)
x	x	x	x	x	x	0	1	Strobed Input
x	x	x	x	x	0	1	1	Strobed Output (active peripheral bus)
x	x	x	x	x	1	1	1	Strobed Output (TRI-STATE peripheral bus)

7.4.1.5 Port Functions - Basic I/O. Basic I/O is the mode of operation of Ports B and C and Mode 0 of Port A (defined by the MDR). Read and write byte operations, and bit operations can be executed in the basic I/O mode. The timing for basic input and basic output modes is shown in *Figures 7-4* and *7-5.* The AC characteristics for input and output timings are listed in the NSC830 Data Sheet in Appendix A.



NOTE: Diagonal lines indicate interval of invalid data.

(Read from ROM, or Port)

46-1

FIGURE 7-4. Read Cycle Timing

When a read occurs, the information is latched from the peripheral (port) bus during the leading (falling) edge of the $\overline{RD}$ strobe. When a write occurs, the port bus is modified after the trailing (rising) edge of the $\overline{WR}$ strobe with data from the AD bus. Port output data remains valid at the output pin from one trailing edge of $\overline{WR}$ strobe to the trailing edge of the next $\overline{WR}$ strobe, which then modifies that port.

7.4.2 Strobed Mode of Operation

The NSC830 provides strobed mode communication via Port A. By programming the mode definition register (MDR, address X'07), one of three strobed mode operations can be evoked, in addition to the basic standard I/O mode previously described.

The strobed modes allow handshake interfacing to asynchronous external devices. Handshake signals are provided by Port C (pins 0, 1, and 2) to facilitate interfacing between an external device and the CPU.

The output data latch of Port A is written to the trailing edge of the $\overline{WR}$ strobe (see *Figure 7-6*). If the MDR defines Port A to be in the strobed output mode, active bus, the contents of the output data latch fall through onto the output pin. If the MDR specifies TRI-STATE bus, the output data latch performs in a similar manner as the active bus except the output pin remains at high-impedance until the $\overline{STB}$ signal becomes valid. The contents of the output data latch is then transmitted to the port output pin.

7.4.2.1 Handshake Signals. In the strobed modes of operation, three signals provide the proper interfacing between the CPU and an external device. These signals are:

1. A $\overline{STB}$ strobe to signify when the external device has loaded data into Port A, or taken data from the Port A
2. A BF signal to indicate when valid data is present
3. An $\overline{INTR}$ signal to notify the CPU when action on its part is required (an interrupt)

The handshake signals are multiplexed with bits 0 through 2 of Port C, (i.e., PC0, PC1, and PC2). When Port A is in the standard I/O mode of operation (MDR bit 0 = 0), these pins operate as standard Port C bits, as described in paragraph 7.4.1. When any strobed mode is selected (MDR bit 0 = 1), these pins take on their handshake functions.

The Port C DDR for the bits used for the handshake signals must be initialized to the proper direction definition for the function performed, as follows:

Port C	Handshake Function	Direction
PC0	Interrupt ($\overline{INTR}$)	Output
PC1	Buffer Full (BF)	Output
PC2	Strobe ($\overline{STB}$)	Input

The Port C data direction register (address X'06) should be written with

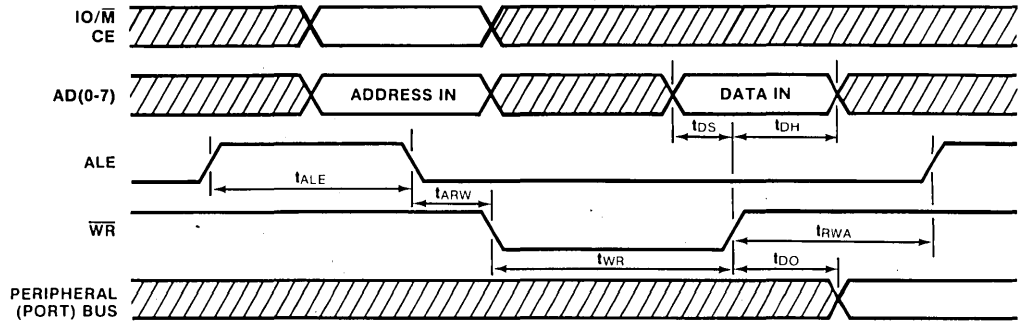
5	4	3	2	1	0
x	x	x	0	1	1

In addition, the data direction register for Port A must be consistent with the data direction selected by the MDR.

7.4.2.2 Interrupt/PC0. The interrupt handshake signal $\overline{INTR}$ is multiplexed on pin PC0. $\overline{INTR}$ is an active low interrupt signal to the CPU. The $\overline{INTR}$ is true (active low) when the CPU is requested to take some action. In the strobed input mode, this indicates data is available to be read. In the strobed output mode, the $\overline{INTR}$ signifies data is being requested by the external device. The CPU places the requested data on Port A.

By setting or clearing the output latch PC2, the interrupt can be enabled or disabled.

7.4.2.3 Buffer Full/PC1. The buffer full handshake signal BF is multiplexed on pin PC1. BF indicates that the Port A buffer contains valid data. In the input mode, BF signifies that the external device has loaded data and the CPU has not yet retrieved the data. In output mode, BF signifies that the CPU has written to Port A and the external device has not accepted the data.



NOTE: Diagonal lines indicate interval of invalid data.

(Write to Port) 46-2

FIGURE 7-5. Write Cycle Timing

7.4.2.4 Strobed/PC2. PC2 shares a pin with the handshake strobe, $\overline{STB}$. $\overline{STB}$ is an active low signal, that is generated by the external device that is interfacing to the CPU via the NSC830. It is used to asynchronously signal when the device performs a transfer from the Port A.

7.4.3 Strobed Input

During strobed input operations, an external device can load data into Port A with the $\overline{STB}$ signal. Data is input to the PA 0-7 input latches on the leading (negative) edge of $\overline{STB}$ (see *Figure 7-7*), causing BF to go high (true). On the trailing (positive) edge of $\overline{STB}$ the interrupt signal, $\overline{INTR}$, becomes valid indicating to the CPU that data is available for reading. $\overline{INTR}$ will become valid only if the interrupt is enabled, that is the DDR for PC2 is true.

When the CPU reads Port A, address X'00, the trailing edge of the $\overline{RD}$ strobe causes BF and $\overline{INTR}$ to become inactive, indicating that the strobed input cycle was completed.

7.4.4 Strobed Output

During strobed output operations, an external device can read data from Port A with the $\overline{STB}$ signal (see *Figure 7-8*). Data is initially loaded into Port A by the CPU writing to I/O address X'00. On the trailing edge of $\overline{WR}$, $\overline{INTR}$ is set inactive and BF becomes valid indicating data is available for the external device. When the external device is ready to accept the data in Port A, it pulses the $\overline{STB}$ signal. $\overline{STB}$ will reset BF with its rising edge and also activates the $\overline{INTR}$ signal.

$\overline{INTR}$, in this mode, indicates a condition that requires CPU intervention, which is the output of the next byte of data.

7.4.5 Strobed Output - TRI-STATE Mode (Mode 3)

The strobed output-TRI-STATE mode and the strobed output-active (peripheral) bus mode function in a similar manner, with one exception. The exception is the data signals on PA 0-7 assume the high impedance state at all times except when accessed by the $\overline{STB}$ signal. Thus, in addition to its timing function, $\overline{STB}$ activates Port A outputs to active logic levels. This Mode 3 operation allows other data sources, in addition to the NSC830, to feed a common external device.

7.5 PROGRAMMING

The various register addresses and bit assignments in the NSC830 are summarized in *Table 7-3*.

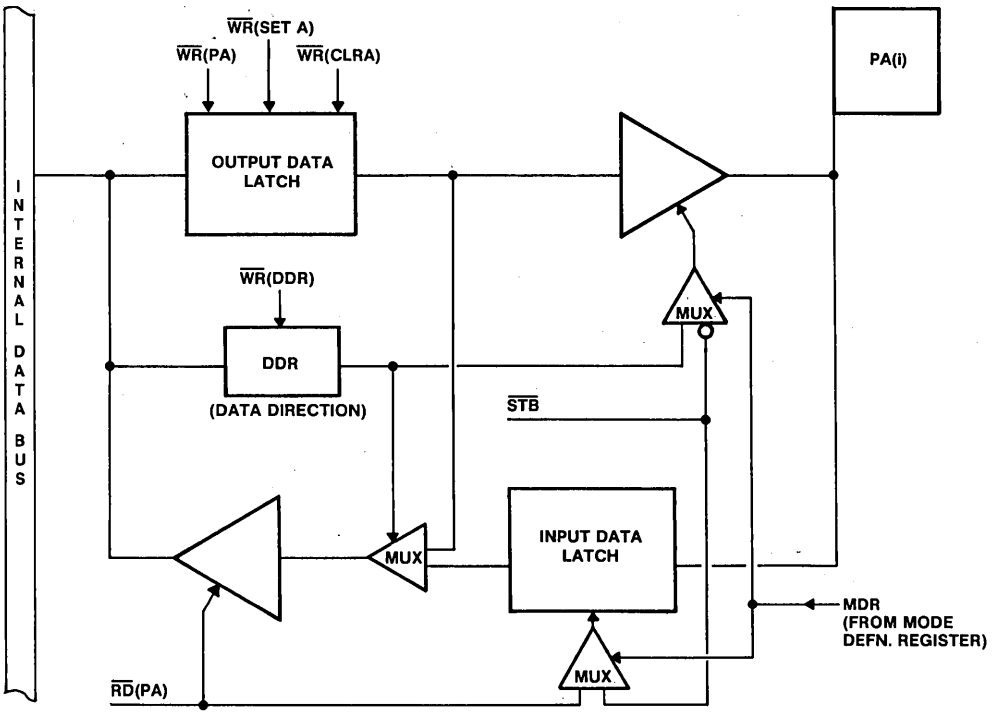
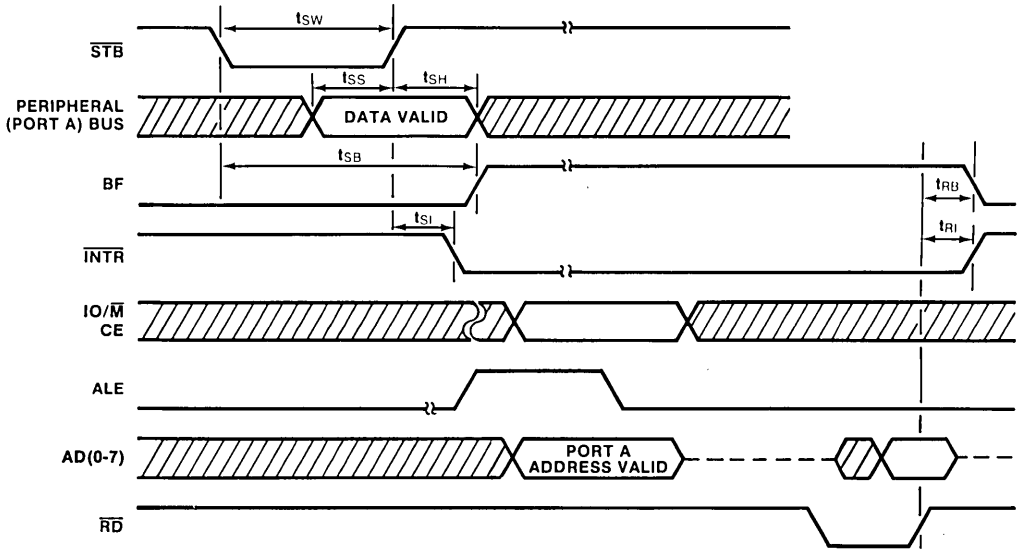


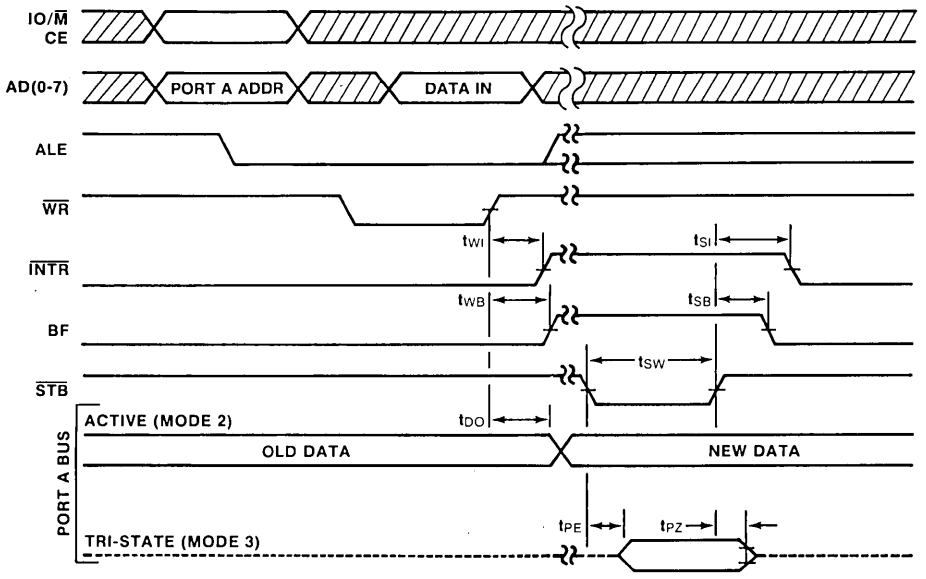
FIGURE 7-6. NSC830 Port A Block Diagram 72-93



NOTE: Diagonal lines indicate interval of invalid data.

Strobed Mode Input 46-3

FIGURE 7-7. Strobed Mode Input



NOTE: Diagonal lines indicate interval of invalid data.

Strobed Mode Output 46-4

FIGURE 7-8. Strobed Mode Output

TABLE 7-3. I/O PORT ADDRESSING

I/O Address					
A_4	A_3	A_2	A_1	A_0	FUNCTION
x	0	0	0	0	READ OR WRITE I/O PORT A
x	0	0	0	1	READ OR WRITE I/O PORT B
x	0	0	1	0	READ OR WRITE I/O PORT C
x	0	1	0	0	WRITE DDR A
x	0	1	0	1	WRITE DDR B
x	0	1	1	0	WRITE DDR C
x	0	1	1	1	WRITE MODE DEFINITION REGISTER
x	1	0	0	0	CLEAR BIT PORT A
x	1	0	0	1	CLEAR BIT PORT B
x	1	0	1	0	CLEAR BIT PORT C
x	1	1	0	0	SET BIT PORT A
x	1	1	0	1	SET BIT PORT B
x	1	1	1	0	SET BIT PORT C

x = "don't care"

Chapter 8

Hardware Support and System Design

8.1 INTRODUCTION

The NSC800 is an 8-bit, single-chip, microprocessor fabricated using double polysilicon, oxide-isolated CMOS technology. This process yields a processor with all the advantages of CMOS (e.g., low-power consumption, high degree of noise immunity, and wide power supply voltage range) that approaches the speed of standard NMOS. The NSC800 has a powerful set of 158 instruction types including 8- and 16-bit arithmetic, block moves and searches, and bit set, test, and reset. Included on-chip are five levels of vectored, prioritized interrupts to increase system flexibility and decrease interrupt response time for most applications.

This chapter describes the basic requirements to design and implement an NSC800-based system. Included are descriptions of the various components available to the user for designing different types of systems, from minimum chip systems to large, expanded systems for complex applications. Also included are some specific applications examples.

8.2 NSC800 ARCHITECTURE - OVERVIEW

The NSC800 architecture (see block diagram, *Figure 8-1*) features a number of on-chip functions that reduce external hardware requirements and increase relative system throughput. A total of 22 programmer accessible registers reduce the number of external memory locations required for temporary storage. The register complement is depicted in *Figure 8-1*. One advantage of this register arrangement is fast context switching in interrupt service routines. This aspect is thoroughly covered in chapter 4.

Other functions include on-chip clock generation, DMA control, dynamic RAM refresh address generation, and multiplexed address/data buses.

All control signals required for system operation are provided by the NSC800. All data communication between the processor, external memory, and peripherals takes place via the 8-bit data bus. The architecture of the NSC800 provides for separate data memory, program memory, and I/O peripherals. A total of 256 unique peripheral locations and 65,536 memory locations can be directly addressed by the processor via the 16-bit address bus.

8.3 SYSTEM TIMING

All NSC800 system timing is generated on-chip by the internal clock generator. This clock has a frequency of one half that of the clock present at the XIN (pin 11) input.

8.3.1 Clock Generation

The system clock can be generated in one of two ways: via the internal clock generator or an external source. *Figure 8-2* demonstrates the use of the internal clock generator.

A crystal can be used (see *Figure 8-2A*) to generate the system clock. In this manner, the frequency will be one half of the fundamental frequency of the crystal in use.

The second method of generating the system clock is to drive the XIN input from an external clock source. *Figure 8-2B* shows one method of implementing this. Note that when connecting an external clock source to the XIN input, the XOUT output should be left floating (open). The system operating frequency will be, again, one half of the input frequency at XIN.

A clock output is provided by the NSC800 and is available at the CLK (pin 9) output. This system clock will be one half of the input frequency and is therefore the basic operating frequency of the NSC800.

8.3.2 Control Strobes

All system timing and data transfers are implemented by the system control strobes. These are: ALE, IO/$\overline{M}$, $\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, and $\overline{RFSH}$. A detailed description of each of these control strobes follows:

ALE Address Latch Enable: This signal indicates, on the high-to-low transition, that a valid address is available on the AD(0-7) bus lines. This is the low-order 8 bits of the 16-bit address for a memory data transfer, the 8-bit address for a peripheral data transfer, or the 8-bit refresh address for dynamic RAM refreshing. In system applications where other components have on-chip address latches, ALE is normally tied to the control strobe input to perform address/data bus separation. If non-multiplexed bus compatible components are used, then ALE can be used to enable a latch to perform bus separation. Refer to section 8.5 for additional details.

IO/$\overline{M}$ Input/Output or Memory Control: This signal indicates to external hardware the nature of the ensuing data transfer. If IO/$\overline{M}$ is high, then the transfer will involve an I/O-mapped device; if it is low, the transfer will be to memory.

$\overline{RD}$ Read: The $\overline{RD}$ signal is an active low control strobe used to gate memory or peripheral data into the NSC800. On the trailing (low-to-high) edge, data present on the AD(0-7) bus lines are expected to be valid and are accepted by the CPU.

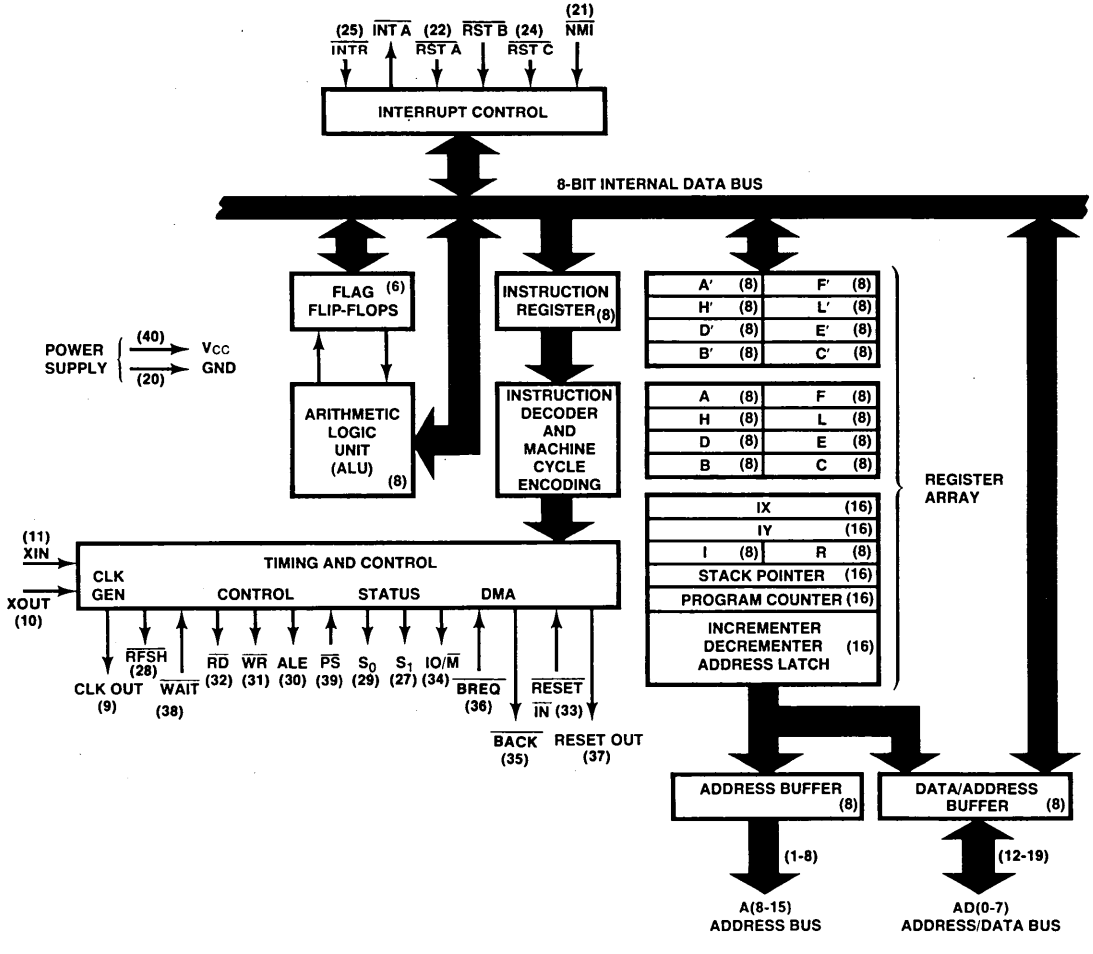


FIGURE 8-1. NSC800 CPU Functional Block Diagram

72-1

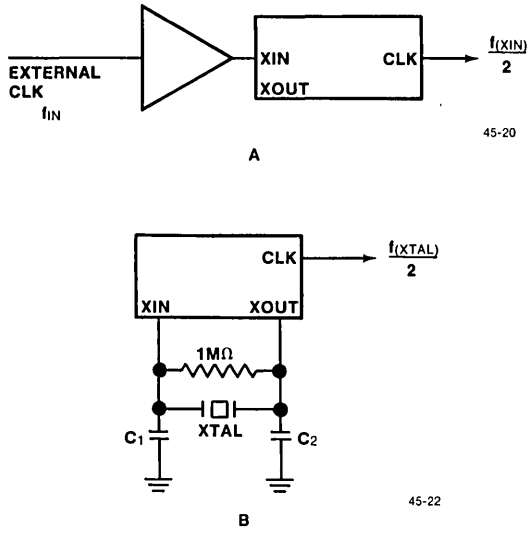


FIGURE 8-2. External Clock Circuits

$\overline{WR}$ Write: The $\overline{WR}$ signal is an active low control strobe and indicates to the external devices that valid data are present on the AD(0-7) bus. Data are guaranteed to be valid on both the leading (high-to-low) and trailing (low-to-high) edges of $\overline{WR}$.

$\overline{WAIT}$: This control input can be used to extend the I/O times of read and write cycles to accomodate slow memories or peripherals. When the $\overline{WAIT}$ input is brought low during a read or write operation, the NSC800 extends the I/O cycle by increments of one T-state. There is no maximum time limit for the $\overline{WAIT}$ signal to be active (low). In order for $\overline{WAIT}$ to be recognized by the NSC800, it must become active (low) within one T-state of the falling edge of ALE and remain valid for the duration of the desired hold time, but not less than (t/2 + 50nsec). Note that in I/O operations, the NSC800 automatically inserts a one T-state wait cycle. This is necessary because many peripherals cannot maintain access time compatibility with the NSC800 (except for peripherals that are members of the NSC800 family).

$\overline{RFSH}$ Refresh: The $\overline{RFSH}$ signal indicates that on the next falling edge of ALE a valid refresh address is available on the AD(0-7) bus. It can be used to initiate a dynamic RAM refresh cycle. However, the falling edge of $\overline{RFSH}$ does not guarantee that the refresh address is still valid. The NSC800 has an on-chip 8-bit wide counter that performs RAM refresh. After each opcode (instruction) fetch cycle, this counter is incremented.

8.3.3 System Initialization

System initialization is performed by two pins on the NSC800, $\overline{RESET\ IN}$ (pin 33) and RESET OUT (pin 37). $\overline{RESET\ IN}$, an active low input, has a Schmitt trigger circuit that produces glitch-free reset signals to the NSC800. RESET OUT is active high and indicates to other devices in the system that the CPU is being initialized and they should also be initialized, if necessary. For power-on reset, a simple R-C network can be connected to $\overline{RESET\ IN}$ as shown in *Figure 8-3*. The time constant of the reset circuit is approximately 100ms to keep $\overline{RESET\ IN}$ active long enough to allow the oscillator to stabilize. $\overline{RESET\ IN}$ must remain active for at least 3 T-states after the oscillator has stabilized to ensure proper initialization of the CPU. Following this stabilization period, the RESET OUT output becomes active (high), and can be used to initialize peripherals in the system. Resetting the NSC800 causes the PC, I, and R registers to be cleared. All maskable interrupts are disabled and the master interrupt enable is reset. Interrupt mode 0 is set to maintain 8080/8085/Z80 compatibility. When RESET OUT becomes low, the NSC800 has completed all internal initialization and the CPU initiates its first opcode fetch from location 0.

In order to reset the CPU manually, a switch can be connected to $\overline{RESET\ IN}$ (see *Figure 8-3.*) This allows a reset to be performed by an operator. The resultant sequence will be the same as previously described for power-on reset.

While $\overline{RESET\ IN}$ is active, the AD(0-7) and A(8-15) bus lines are in the high impedance (TRI-STATE™) condition and all control strobes are in the inactive state.

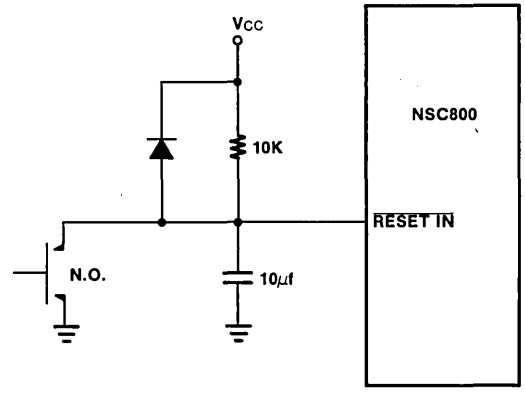


FIGURE 8-3. NSC800 Initialization

72-4

8.4 POWER SUPPLY CONSIDERATIONS

The NSC800, being CMOS, is not sensitive to minor power supply fluctuations. Therefore, tight voltage regulation is not necessary. However, some specific considerations must be taken into account when designing, or selecting the power supply for an NSC800 system. First, with CMOS components, the higher the supply voltage is, the higher the power consumption. This is due to an increase in the drain-to-source current drawn while the complementary pair of transistors are changing logic states. Note that in a

quiescent CMOS system, where no logic transitions are being made, no effective power is consumed (except for leakage currents). The trade-off is that higher supply voltage allows for higher operating frequencies. Also, the power consumption of a CMOS system increases directly in proportion to the frequency of operation. (See AN-77 for a complete analysis of CMOS operation). In light of the above, it is obvious that when deciding on the power supply design, the designer should also consider operating the NSC800 system at the lowest voltage and frequency to accomplish the intended task. This will in turn reduce power dissipation to the absolute minimum, thus reducing the size of the power supply required.

One other point to remember when designing a CMOS system is that all unused inputs should be terminated, tied either high or low, but never allowed to float since a floating input will cause CMOS to draw current.

The low power of the NSC800 makes it ideal for many applications that were previously not suited to microprocessors because a high-performance, low-power microprocessor simply was not available. The system designer was forced to compromise and accept a low-performance processor or attempt to make a high-power device work. These applications are typically those that require battery operation, either due to being remotely located, or required to operate in a hostile environment where hermetic sealing is necessary. Either of these applications require a processor to operate without the bulk or heat generating aspects of a standard linear or switching power supply and fan assembly, if induction cooling is inadequate. Since the power drawn by the NSC800 is extremely low (approximately 50mw), it can be powered by a small battery for relatively long durations. The $\overline{PS}$ (Power-Save) function (to be explained later), can extend battery life even further.

Another advantage of the low-power operation of the NSC800 is the capability of designing a system with battery backup for power failure. In a system that uses a power supply, power fail detection circuits can activate the Power-Save input to the CPU, causing it to stop operating, yet maintain all internal status. At the same time, battery backup would come on-line to sustain power to the critical components of the system. With the CPU held in a static condition, power consumption is reduced by approximately 50% throughout the system. When main power returns, the NSC800 starts operating exactly where it was stopped. *Figure 8-4* shows a circuit that will implement this capability plus the added feature of a trickle charge for the NiCad batteries used during backup.

8.5 SYSTEM BUS STRUCTURE

The NSC800 CPU uses a multiplexed bus that consists of 16 bits of address and 8 bits of data. The upper 8 bits of the address bus are present on the A(8-15) lines. The lower 8 bits of address are time multiplexed on the 8-bit data bus, AD(0-7). The 16-bit address allows the NSC800 to directly address 65,536 bytes of memory and 256 I/O locations. Two basic types of systems can be implemented with the bus structure of the NSC800. First, a multiplexed bus system, and second a separated bus system that divides the multiplexed bus into a separate 16-bit address bus and 8-bit data bus.

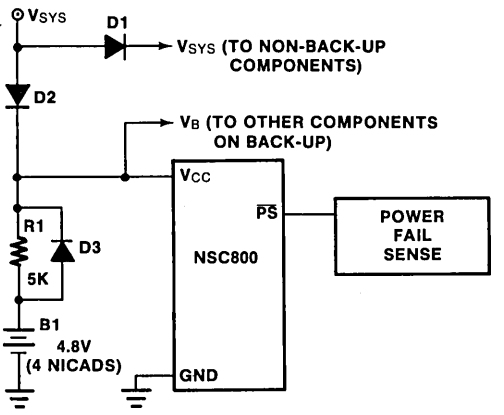


FIGURE 8-4. Battery Back-Up In Power-Save 72-5

In a multiplexed bus system, all components that use the address and data buses must have the ability to separate the buses internally. All bus-oriented components in the NSC800 family (NSC810, NSC830, NMC6504, NMC6514) have this capability. As previously stated, the ALE control strobe is used in performing the function of bus separation within these parts. If no other types of components are used in the user's system, or if the additional components used are capable of operating on a multiplexed bus (i.e., have internal address latches), then no expansion of the system bus is necessary. The advantage of not expanding, or separating, the bus is that fewer interconnects are required, reducing the complexity of PC board layout.

In a system that requires the use of components that do not have the capability of interfacing directly to a multiplexed bus, some type of bus separation is needed to produce a separate address and data bus. One method of achieving this separation utilizing the MM82PC12 is shown in *Figure 8-5*. The MM82PC12 is pin-for-pin and function-for-function compatible with the industry standard INS8212. In this example, the MM82PC12 is operated in the "fall-through" mode to reduce the propagation delay through the latch. The maximum delay for this latch is 60nsec, which will not adversely affect access times. By using this, or some similar method, the system now has a full 16-bit address bus and 8-bit data bus that will enable direct interfacing to most standard microprocessor peripheral and memory devices.

8.6 BUFFERING

For many applications, the minimum system configuration of the NSC800 will be adequate. It is, therefore, possible to design a system based on the NSC800 family that does not require any buffering of the address, data, and control buses. However, many applications that are suited to the high-performance aspects of the NSC800 family will need to expand the memory and peripheral sections of the system to a point that exceeds the interface specifications of the components. Two dedicated devices, the MM82PC08 and MM82PC12, in the family of

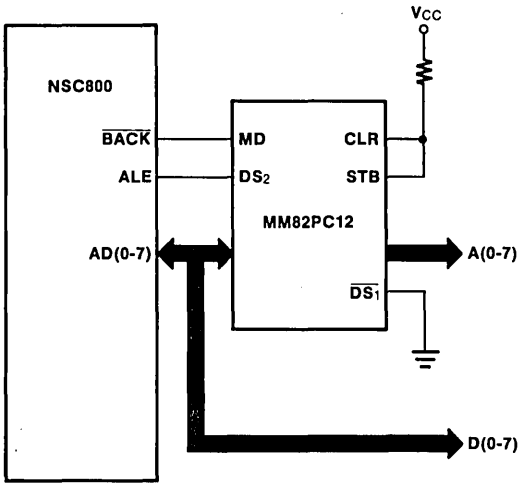


FIGURE 8-5. System Bus Separation 72-6

compatible interface components provide the required buffering.

In large systems based on the NSC800 family where the load exceeds the drive or capacitance specifications, buffering will be necessary. For this purpose, the MM82PC08 buffer/transceiver is ideally suited. The MM82PC08 can sink up to 5mA, which should be adequate for most medium to large systems. *Figure 8-6* is a schematic for one method of using this device for buffering both the high-order address bus and the low-order address/data bus. With this method, the multiplexed bus structure of the NSC800 is mainatined. In this example, the T/R signal of the buffer for the AD(0-7) bus is controlled by the $\overline{RD}$ control signal from the NSC800. During an opcode fetch, or data read, either from memory or peripheral, the MM82PC08 is gated into the receive mode. At all other times, the MM82PC08 is in the transmit mode. The buffer on the A(8-15) address lines is hard-wired in the transmit mode since no data is accepted as input on that bus. An additional signal, derived from the Bus Acknowledge ($\overline{BACK}$) signal, is used to place the bus drivers in the high impedance state. This is optional and intended for systems that will be operated in a DMA mode. When the NSC800 recognizes a Bus Request ($\overline{BREQ}$) and responds with a $\overline{BACK}$, all buses are tri-stated.

If, on the other hand, the buses are separated for the system, then the MM82PC12, which demultiplexes the buses, is adequate for buffering the low-order address bus (5mA sink), and two MM82PC08s are used to buffer the high-order addresses and data. This configuration is shown in *Figure 8-7*. Again, in this example, the $\overline{BACK}$ signal is shown as an option for tri-stating the buses for DMA purposes.

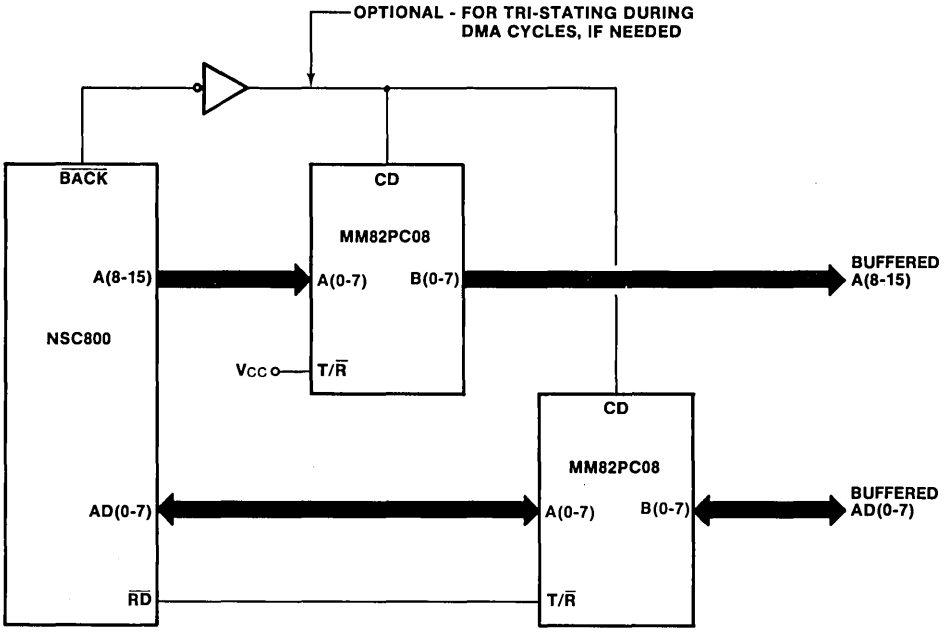


FIGURE 8-6. Buffering 72-7

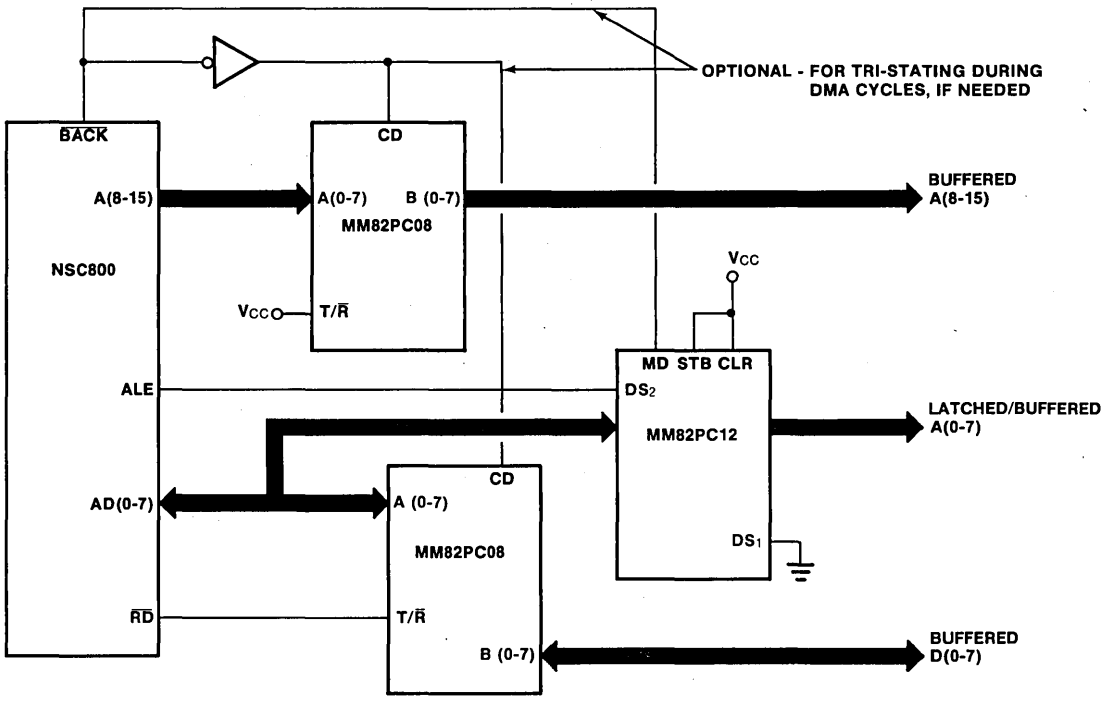


FIGURE 8-7. Buffering/Demultiplexing

72-8

8.7 INTERFACING

The NSC800 uses the $\overline{RD}$ and $\overline{WR}$ control strobes to communicate with both memory and I/O devices. The CPU issues an address followed by either the $\overline{RD}$ or $\overline{WR}$ control strobe, depending on whether the machine cycle is a data input or data output. Read and write cycles are similar in timing, except for the special case of an opcode fetch cycle. The access time for an opcode fetch cycle is shorter than a memory read cycle by the amount t/2, where t is one half the duration of the clock at the XIN input. During a read or write cycle, the NSC800 could be reading memory, or I/O, but no distinction is made internally, other than in the logic state of the IO/$\overline{M}$ control strobe (logic 1 for I/O and logic 0 for memory).

If special functions are required, that cannot be fulfilled with the components presently available to the NSC800 family, then interfacing to various types of memory and peripherals will be necessary. This situation would arise, for example, if the I/O ports of the NSC810 and/or NSC830 were not adequate for the system function desired. The following section gives the designer some basic guidelines for developing the circuits required to perform the interface.

8.8 ADDRESS DECODING

When interfacing to memory and peripherals, control strobes must be generated to enable the device to either receive or transmit data between it and the CPU. This generally takes the form of a chip select. Many methods are available to produce the chip select signal, derived from the address bus. The 16-bit address word output by the NSC800 during any data input/output operation provides a total addressing capability of 65,536 (64K) locations. For special I/O instructions, this address space is 256 locations, since these particular instructions have an 8-bit address, duplicated on both the low-order 8 bits and high-order 8 bits of the 16-bit address bus. The least-significant address bits are normally used to directly address the internal locations of the memory or peripheral device. For example, a 1K byte memory would require the least-significant 10 address A(0-9) bits to fully access all internal data. This leaves the most-significant 6 address A(10-15) bits for selecting any one particular memory chip.

The actual number of bits required to select any individual device depends on the total number of devices in the system. Allocation of address space for each device is accomplished by decoding address bits to derive the required control signals.

In small systems, the use of a decoding device may not be necessary; the high-order address bits themselves can be used as the chip select signals. This is demonstrated in the schematic diagram of a minimum NSC800 system (see *Figure 8-8*). Note that address bits A12 and A13 alone are used as the chip-select signals. Using just address bits to

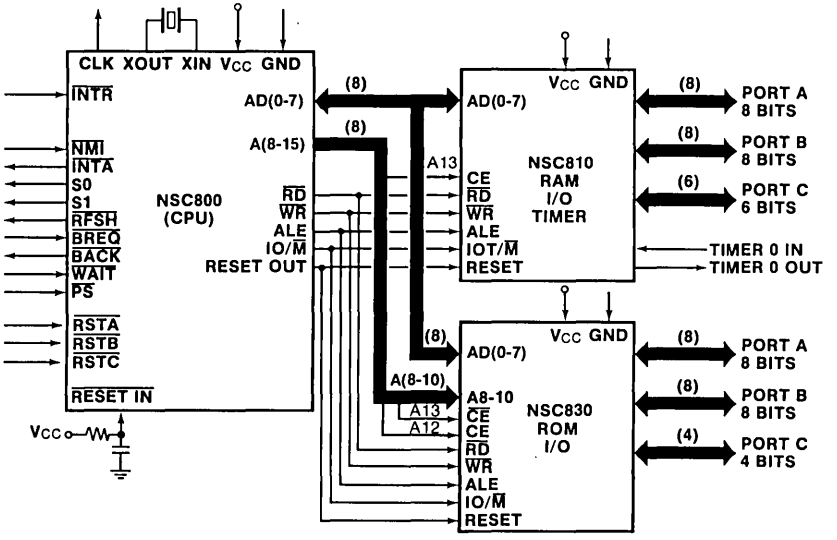


FIGURE 8-8. Minimum NSC800 System 72-9

perform device enable eliminates the need for extra components in the system, thus reducing PC board area and power consumption. However, this method, called "linear selection," has some drawbacks. First, it limits the number of devices that can be addressed because it makes inefficient use of the address space. Second, it makes future expansion of the system tedious, if not impossible. This can be seen, again from the minimum system. The NSC830 will respond to addresses in the range of X'0000 to X'7FF, as is intended, but it will also respond to addresses in the range of X'4000 to X'47FF, X'8000 to X'87FF, and X'C000 to X'C7FF since in all of these cases, address bits 12 and 13 are low. In this small system, where no other components are intended, this is no probelm. In order to expand the amount of ROM with additional NSC830s, care would have to be taken that the additional NSC830 was not selected simultaneously with other components in the system.

For larger systems, a preferred method of generating chip selects is to make use of an address decoder. In the NSC800 family, this takes the form of the MM74PC138. *Figure 8-9* depicts one way of connecting this device to produce chip selects for 8 1K devices. Note that the IO/$\overline{M}$ control signal is used to enable the address decoder for memory devices only. When an I/O operation is in progress, the MM74PC138 is disabled. This is necessary since the I/O address is duplicated on both the upper and lower 8 bits of the 16-bit address bus, and it is conceivable that the bit pattern might cause both the addressed I/O device and a memory device to be selected simultaneously. If the decoder was intended to decode chip selects for I/O devices, then the IO/$\overline{M}$ signal could be connected to the G2B input with the G1 control input grounded. Various combinations of address lines can be used as inputs for the MM74PC138 to produce chip selects for other devices of differing sizes.

8.9 MEMORY INTERFACING

The amount of memory, either ROM or RAM, used in an NSC800 system is determined by a variety of application dependent factors that include program size, data storage requirements, speed considerations, and system power constraints. In many applications, the minimum system configuration provides an adequate amount of memory. This system (refer to *Figure 8-8*), consists of one each NSC800, NSC810, and NSC830, providing 2K bytes of ROM and 128 bytes of RAM plus I/O. Additional NSC810s and NSC830s can be added to expand the memory and I/O capabilities of the system, within certain limits. Some large systems may require more than 1K of RAM, but it could be undesirable to have as many as 8 NSC810s in the system. For this situation, the 1K x 4 (NMC6514) and the 4K x 1 (NMC6504) have been developed. These memory components are also fabricated with the P^{2}CMOS

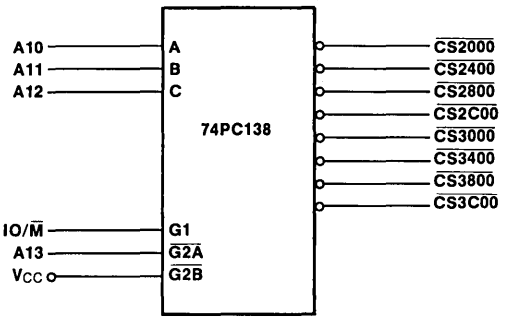


FIGURE 8-9. Address Decoding 72-10

process, and are speed and bus compatible with the NSC800. *Figure 8-10* shows an expanded system using the NMC6514 to provide 4K of R/W memory. Since the NMC6514 has on-chip address latches, it is not necessary to demultiplex the address/data bus for interfacing. An MM74PC138 is used to decode the address space for the RAM.

This memory interface makes use of the on-chip address latches to eliminate the need for external bus separation. The E input is a combination of ALE and the chip select, produced by the MM74PC138. The lower 8 address bits are latched at the falling edge of the E input. The lower 8 addresses, in conjunction with the two least-significant address bits of the upper 8-bit address bus, select one location out of 1024. The chip enable input allows the RAM to place this data on the output, for a read operation, or take the data on the inputs and write it into the location for a write operation. The $\overline{WR}$ signal is connected to the W input to determine if the data transfer is a read or write. In operation, the RAM chips are always in the read mode unless a write data transfer is initiated by the CPU. When the E input is removed by the address decode for the next I/O cycle, the NMC6514 sets its bus connection in the high impedance state.

For interface to other types of R/W memory devices, output data are valid on both the leading and trailing edge of $\overline{WR}$, so either edge can be used to perform a write operation. For the standard-speed version of the NSC800, the data setup and hold times are 125nsec.

Not all of the possible memory components that can be interfaced to the NSC800 have on-chip address latches and, therefore, the address bus must be demultiplexed. For a description of methods for accomplishing demultiplexing, refer to section 8.5.

8.10 I/O INTERACING

The NSC800 provides two basic ways of addressing I/O devices: memory-mapped I/O, and standard or I/O-mapped I/O. The distinguishing characteristic between these two methods is the use of the IO/$\overline{M}$ signal generated by the CPU during a data transfer cycle. If the peripheral uses the IO/$\overline{M}$ signal to determine whether the read or write cycle is directed towards I/O or memory, then the system would be defined as having I/O-mapped I/O. If, on the other hand, an address bit is used in place of the IO/$\overline{M}$ signal, then the system is considered to have memory-mapped I/O. The advantages and disadvantages of both methods are discussed later. Interfacing to the NSC830 ROM-I/O is used in demonstrating these two methods.

8.10.1 Standard I/O Mapped Interface

In *Figure 8-8* the NSC830 is connected to the NSC800 in a standard I/O-mapped configuration. The IO/$\overline{M}$ signal from the CPU is connected to the IO/$\overline{M}$ input of the NSC830. Consequently, when the CPU makes a memory reference to the address space occupied by the NSC830, the IO/$\overline{M}$ signal will be low and the device will respond to the access by decoding the address supplied and output data from the internal ROM. When the CPU performs an I/O reference to the NSC830, the IO/$\overline{M}$ signal will be high and it will respond with the internal I/O section. In this configuration, the same chip selects are used for both memory and I/O transactions. The chip selects are user programmable; thus, if this type of I/O technique is used, proper selection of the chip select states is required so that the NSC830 will respond when it is addressed in either mode. The I/O address generated by the CPU is composed of 8 bits which are duplicated on the upper and lower 8 bits of the 16-bit address bus. In the example shown where both chip selects have been programmed active low, the ROM portion will respond to addresses in the range of X'0 to X'7FF. The I/O section will respond to addresses in the range of X'0 to X'E (see discussion of the NSC830 for details of internal I/O addressing).

The disadvantage of using I/O-mapped addressing in this case is that only the INPUT and OUTPUT related instructions can be used for communicating with the I/O ports of the NSC830. While this is not as great a disadvantage with the NSC800 as it is with the 8080 or 8085, due to the richer instruction set, there may be applications where the use of some of the memory reference instructions might be desirable.

8.10.2 Memory Mapped I/O

Figure 8-11, demonstrates the same interface to the NSC830 using memory-mapped I/O techniques. In this example, a high-order address bit is used to determine what is addressed-memory or I/O. Address bit 15 is connected to the IO/$\overline{M}$ input of the NSC830 and, therefore, only addresses above X'8000 will communicate with the I/O portion, but any of the memory reference instructions can be used. This allows such instructions as ANDing or ORing, to and from the ports, which would not be available had standard I/O-mapped interfacing been used.

8.10.3 Memory and Peripheral I/O Cycle Extend

When interfacing to memories or peripherals that cannot meet the access time requirements of the NSC800, the I/O cycles of the CPU must be extended. A dedicated input to the NSC800 is provided for this purpose. This signal is the $\overline{WAIT}$ input. The $\overline{WAIT}$ input is sampled by the CPU during the falling edge of the clock during T2 of any I/O cycle. If $\overline{WAIT}$ is active (low) at this time, the NSC800 will insert a one-clock cycle (T-state) hold in the I/O operation. This occurs after the $\overline{RD}$ or $\overline{WR}$ strobe is active and gives the addressed location additional time to make output data ready (if read) or accept input data (if write). The wait cycle (Tw) is merely an additional T2 cycle, and if $\overline{WAIT}$ is no longer active (high) at the falling edge of CLK during Tw, the wait will be removed and the I/O cycle will be completed. If, however, $\overline{WAIT}$ is still active at this time, an additional wait cycle will be generated. This can go on indefinitely until the access times of the addressed memory or peripheral device have been met.

On any data transfer between the NSC800 and a peripheral device (one that is accessed via the special I/O instructions), the CPU automatically inserts one wait state. As stated previously, this is necessary for two reasons. First, to maintain Z80™ compatibility. Secondly, and perhaps as important, many of the potential peripheral controllers that will be interfaced to the NSC800 cannot maintain access time compatibility with it due to its high performance specifications.

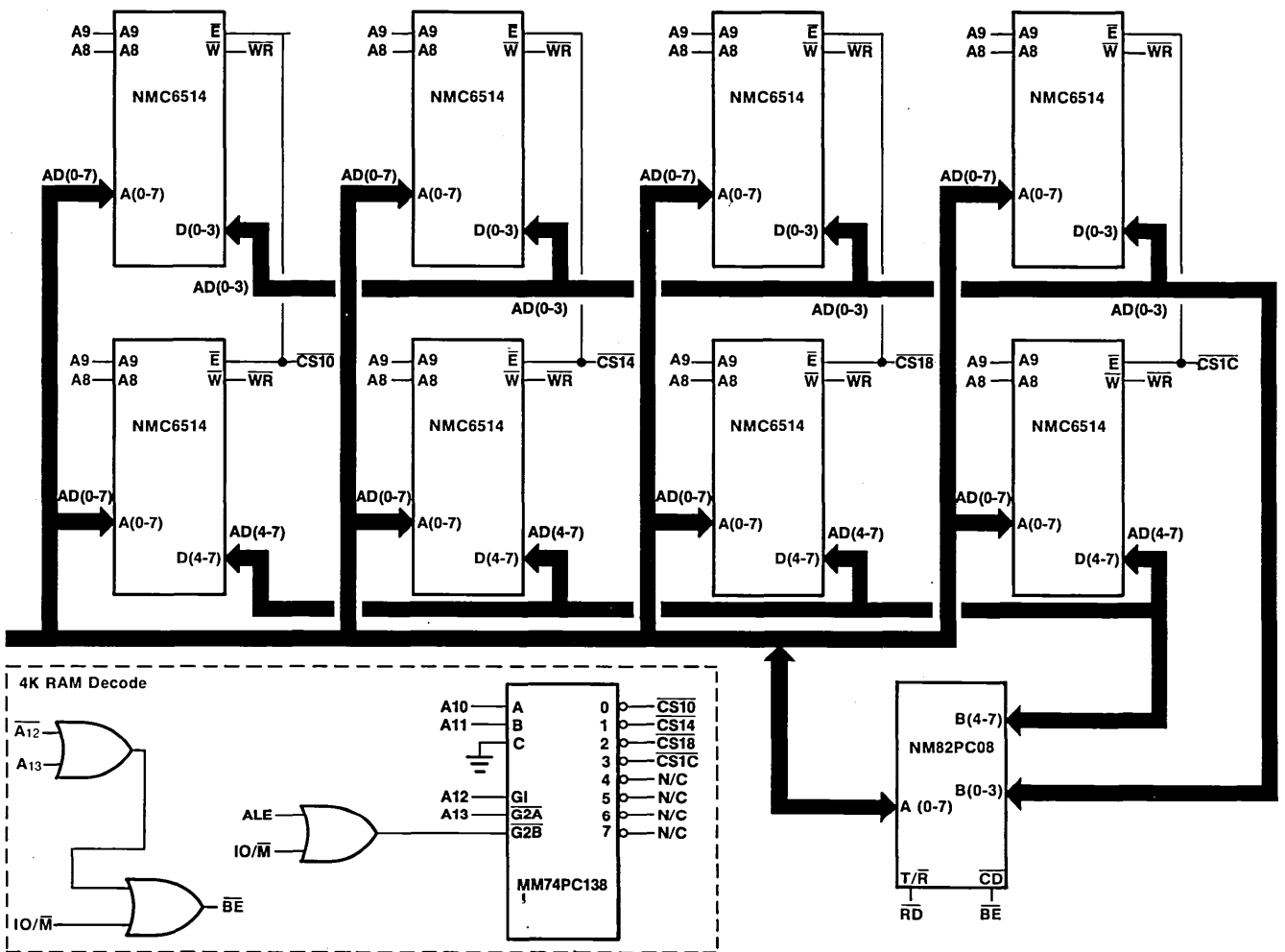


FIGURE 8-10. Expanded R/W Memory System

72-11

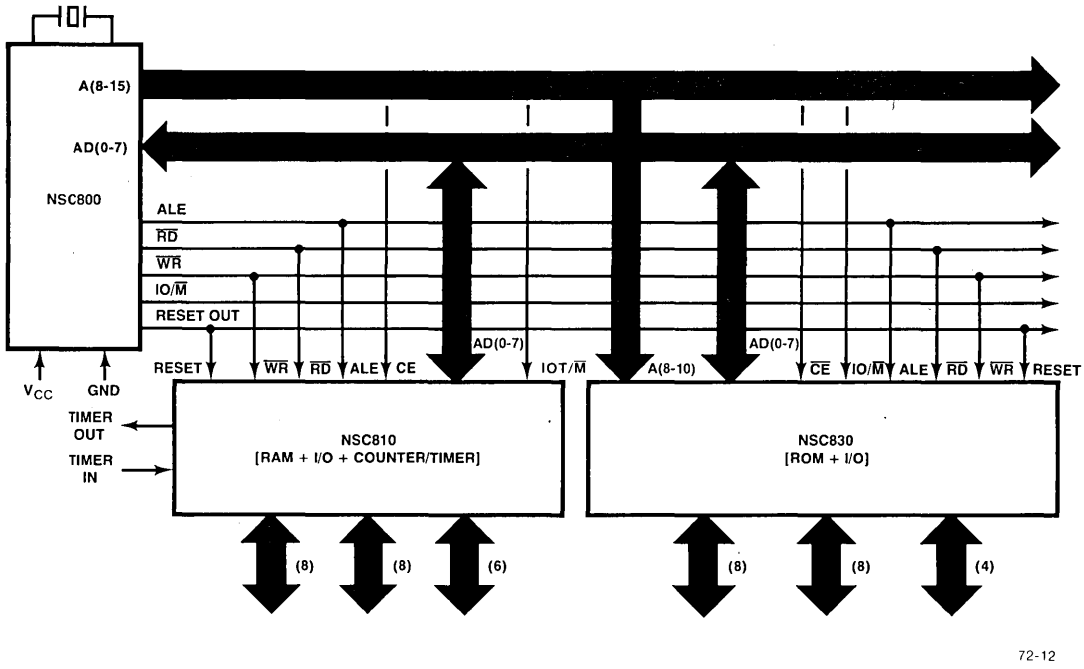


FIGURE 8-11. Minimum NSC800 System (Memory-Mapped I/O)

Figure 8-12 shows a method of inserting one wait state on any memory transaction. Note that on this implementation, no additional wait states will be generated for peripheral I/O operations. Thus, one wait state will be generated for each memory operation and one for each I/O operation (automatic). *Figure 8-13* demonstrates a method of generating multiple wait states through the use of a shift register. The H input is always grounded and if all other inputs (A-G) are logic one, no wait states will be generated. For each additional input grounded, one wait state will be produced. For example, if inputs F and G are grounded, two wait states will be inserted by the NSC800 on any I/O operation. It should be noted also that the shift register used, the MM74C165, is not a P2CMOS device and therefore will not be fast enough to generate wait state insertion for the NSC800A. It is compatible, however, with the standard speed version, NSC800. To utilize this method with the NSC800A, the use of a bipolar version, DM74LS165, will be necessary. Any unused input to the shift register should be tied high along with the Serial In (SI) input. For the example shown, inputs SI and A-E would be tied high so that the shift register will produce only two wait states.

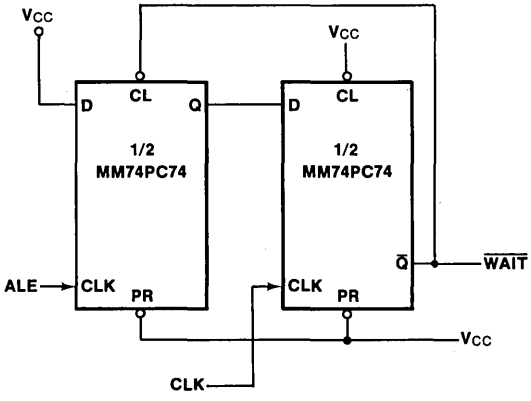


FIGURE 8-12. Single $\overline{\text{WAIT}}$ State Generation 72-13

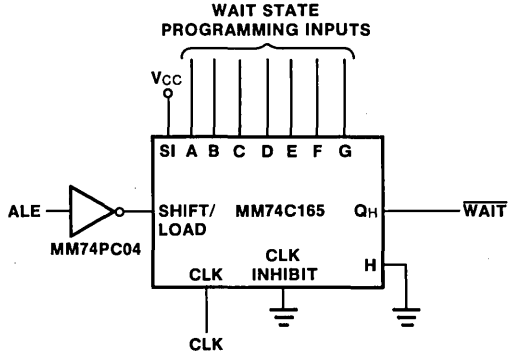


FIGURE 8-13. Multiple $\overline{\text{WAIT}}$ State Generation 72-14

It may not be desirable to produce wait states for all I/O transactions, as this would degrade the overall throughput of an NSC800 system. Since all memory and peripheral devices in the NSC800 family are speed compatible with the NSC800 CPU, only accesses to other types of devices (those that aren't speed compatible) would need the addition of wait states. Therefore, it would be more efficient to include a qualifier to make a decision whether to produce wait states. This could take the form of logically including the chip select for the slow peripheral or memory with the wait generating circuitry.

8.11 INTERFACING TO OTHER FAMILIES

In many applications, the NSC800 family of components must be interfaced to components that are operating at incompatible voltage levels. This could occur, for example, if the NSC800 is operating at 12V and must interface with a TTL device operating at 5V. Voltage level translation would be necessary to produce compatible signals. Two components in the standard CMOS family are designed for this purpose. The MM74C902 accepts input from CMOS (operating, in this example, at 12V) and outputs TTL compatible voltage levels with a fan-out of 2 TTL loads. *Figure 8-14* depicts how this direction of level translation can be implemented using the MM74C902. For translation in the other direction, from TTL at 5V to CMOS at a higher voltage, use of the MM74C906 is indicated. This connection is shown in *Figure 8-15.* Pull-up resistors are required on the CMOS side of the MM74C906 because the outputs are open drain to facilitate device connections to various system supply voltages. The output of the translator is compatible with the input levels of the P²CMOS component.

8.12 APPLICATIONS OF THE NSC800 FAMILY

This section discusses several applications of the NSC800 family of components. *Figure 8-8* shows an example of a minimum system based on the three primary chips in this family. The following sections describe expanded systems based on the minimum system, with extended capabilities, using some of the other compatible components that can be interfaced, either directly or indirectly, to the NSC800.

8.12.1 Expanded NSC800 System

Figure 8-16 shows how the designer might connect additional NSC830s and NSC810s to produce an end system with 4K of ROM, 256 bytes of RAM, 4 timers, and a large number of I/O ports. Standard I/O-mapped addressing is used. To reduce the component count, the chip selects of the NSC830s have been programmed to make the most efficient use of address selection using only address bits (linear selection).

8.12.2 Serial I/O Using Port Bits of NSC830

Figure 8-17 shows methods of using the ports of the NSC830 as a serial I/O interface. The interface is to a TTY via a 20ma current loop. The outputs drive discrete transistors which produce the 20ma signal required to transmit to the TTY and the current required to enable a reader relay. The reader relay is the standard type supplied with National Semiconductor TTYs. This allows the CPU to control the tape reader, turning it on and off as needed. A current loop to voltage filter is provided on the input from the TTY to produce recognizable signals for the NSC830. Use of the bit set and clear functions of the

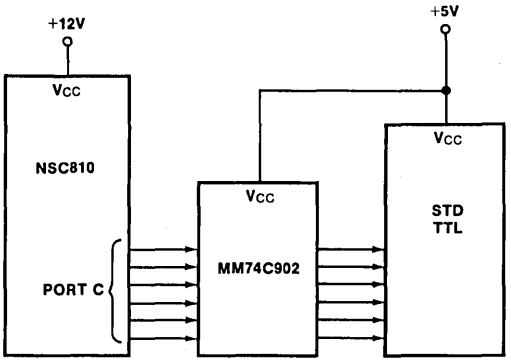
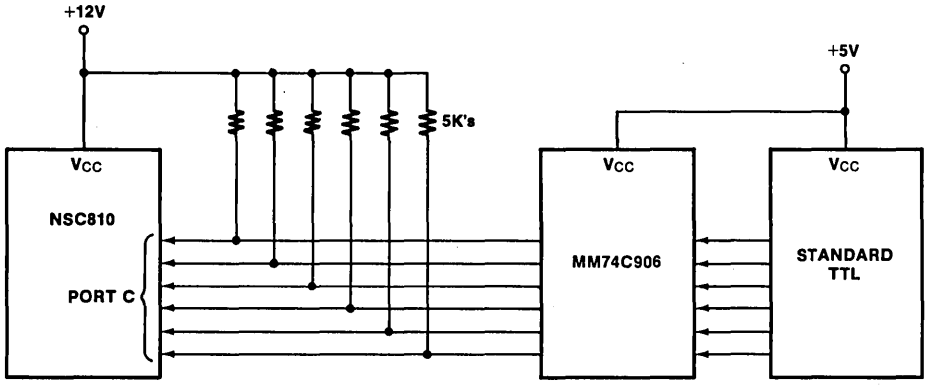


FIGURE 8-14. P²CMOS at High Voltage to Standard TTL

72-15



72-16

FIGURE 8-15. Standard TTL to P²CMOS at Higher Voltage

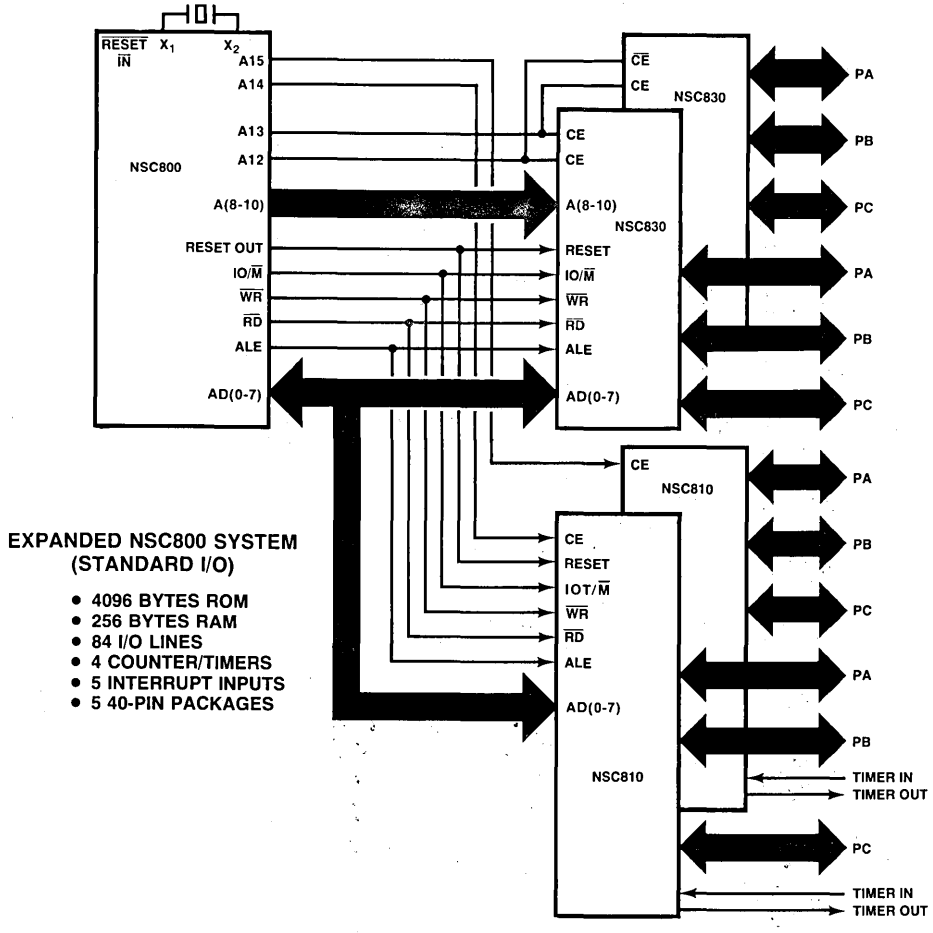


FIGURE 8-16. Expanded NSC800 System

72-17

NSC830 can also be used, or if the NSC830 is configured in the memory-mapped I/O technique, the bit set and reset instructions of the NSC800 could also be used.

The serial data stream transmitted to the TTY is produced by first sending a start bit followed by 8 data bits and ending with two stop bits. The port bit is manipulated to produce this data stream by using output instructions from the CPU after it has determined whether the next bit should be a "1" or "0". Software delay is used to produce the timing for each data bit. If an NSC810 is also included in the system, one of the timers could be set up to produce an interrupt output at the proper timing for each bit, thereby removing this task from the CPU.

The input stream from the TTY comes into the port bit (PB7) of the NSC830. The port is scanned by the CPU for the occurrence of a start bit. After receiving a start bit, the CPU times to the center of the bit to synchronize and then samples the input at each whole bit duration until all data bits have been received.

8.12.3 Emulating the NSC830

During the prototyping of hardware/software integration, programs being written often require many changes. If the end system is to use the NSC830 ROM-I/O, great benefit would be realized if the program could be checked out resident in the normal location in which it will be situated, i.e., in the NSC830 ROM. However, since an EPROM version of the NSC830 is not currently availabe, this is not possible using only one component. This application describes a method of accomplishing this by emulating the NSC830 using a ROMless NSC830 and an MM2716 EPROM.

In order to emulate the NSC830, the emulating device must have the same complement of I/O pins, with the

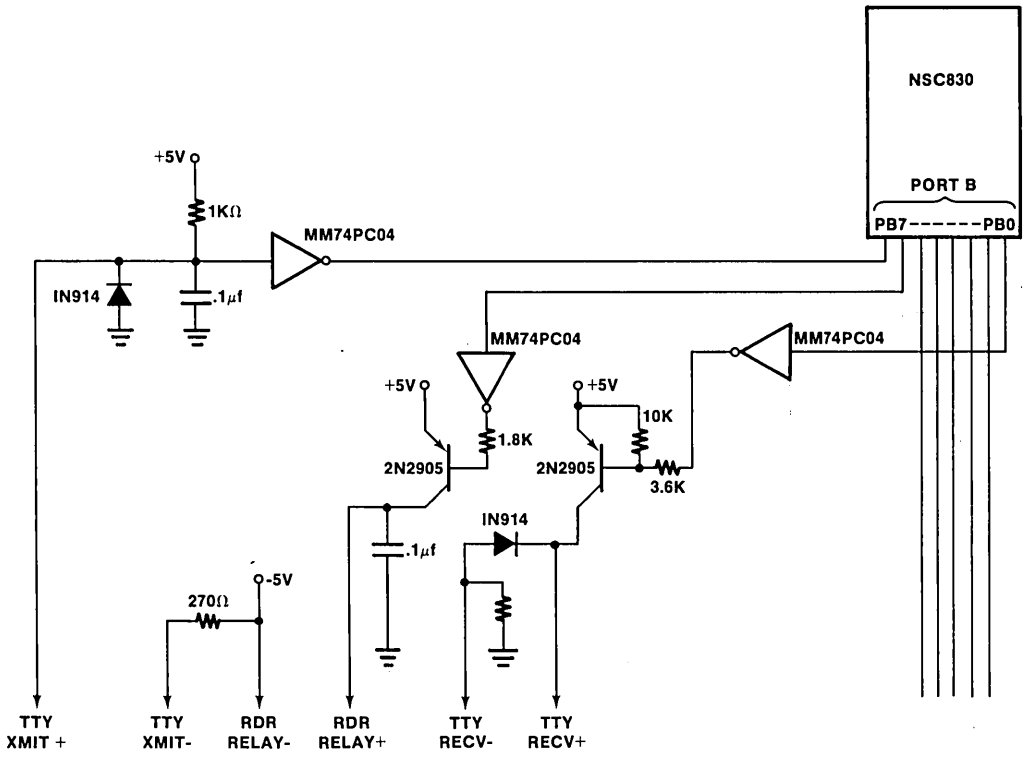


FIGURE 8-17. Serial I/O Via Port I/O Pins

72-18

same drive capabilities as the NSC830, and have a 2K x 8 EPROM to store the program that will eventually be contained in the NSC830. The easiest way to accomplish this is to use a ROMless NSC830 (NSC831) which is identical to the 830 in every detail except for the ROM. In addition to this, a MM2716 type EPROM is used for program storage. Since the MM2716 does not have on-chip address demultiplexers, an MM82PC12 8-bit latch is used to perform the function of address/data bus separation. The MM82PC12 is used in the fall-through mode to keep access time to a minimum. Added to these three parts are a couple of P²CMOS logic elements to perform the function of switching between the NSC831 (for all I/O operations) and the MM2716 (for any memory operation). *Figure 8-18* depicts the actual schematic diagram of the NSC830 emulator.

If the CPU in the system is the NSC800 (as opposed to the NSC800A), no timing restrictions occur. The MM2716 EPROM can operate at the 2.5MHz rate of the NSC800. If, however, the CPU is an NSC800A, the user prototyping system must insert one wait state during any memory operation involving the NSC830 Emulator. The worst case access time of the MM2716 is 300nsec, and the required access time for IFETCH with the NSC800A is 250nsec. Therefore, during program debug, the system must produce this wait state insertion. When the final end system is operating, the wait state will not be needed as the access time of the NSC830 is compatible with the NSC800A.

The NSC830 Emulator can be constructed using standard wire-wrap techniques on a small wire-wrap board. A cable is used to connect the emulator to the prototyping system. The cable should be terminated in a 40-pin DIP plug which occupies the socket intended for the NSC830. This cable should be kept as short as is convenient, but in no case should it be longer than eight inches.

Jumpers on the emulator card are provided for proper selection of the chip-select polarities of the prototyping system. Since both chip selects on the NSC831 are programmed active low, the inverters are necessary for those applications where the user supplies one or both chip selects active high.

8.12.4 Data Acquisition System

A natural application for the NSC800 is one that requires remote operation. Since power consumption is low if the system consists of only P²CMOS or CMOS components, the entire package can conceivably operate from only a battery power source. In the application described herein, the only source of power will be from a battery pack composed of a stacked array of NiCad batteries (see *Figure 8-19*).

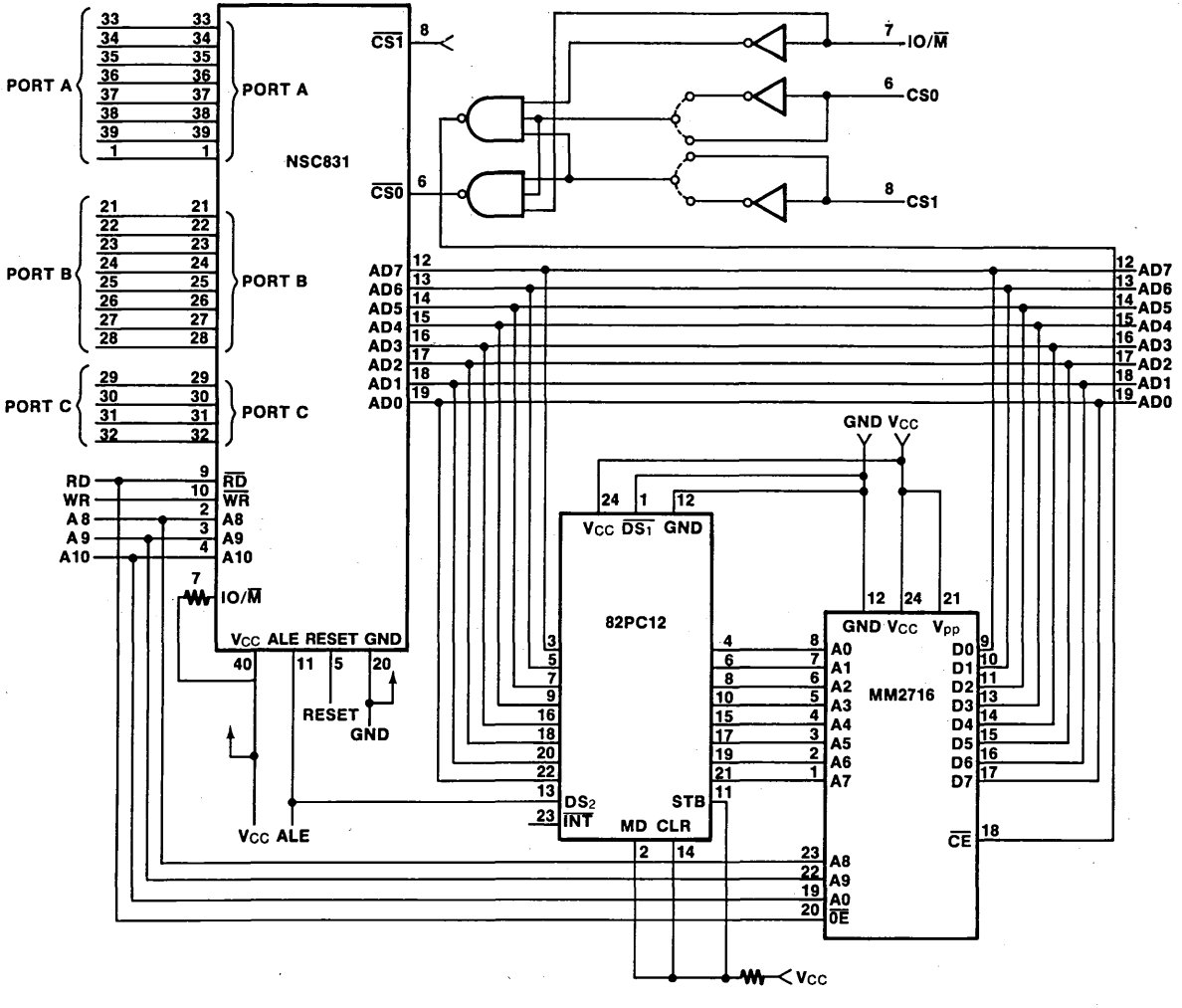


FIGURE 8-18. NSC830 Emulator

72-19

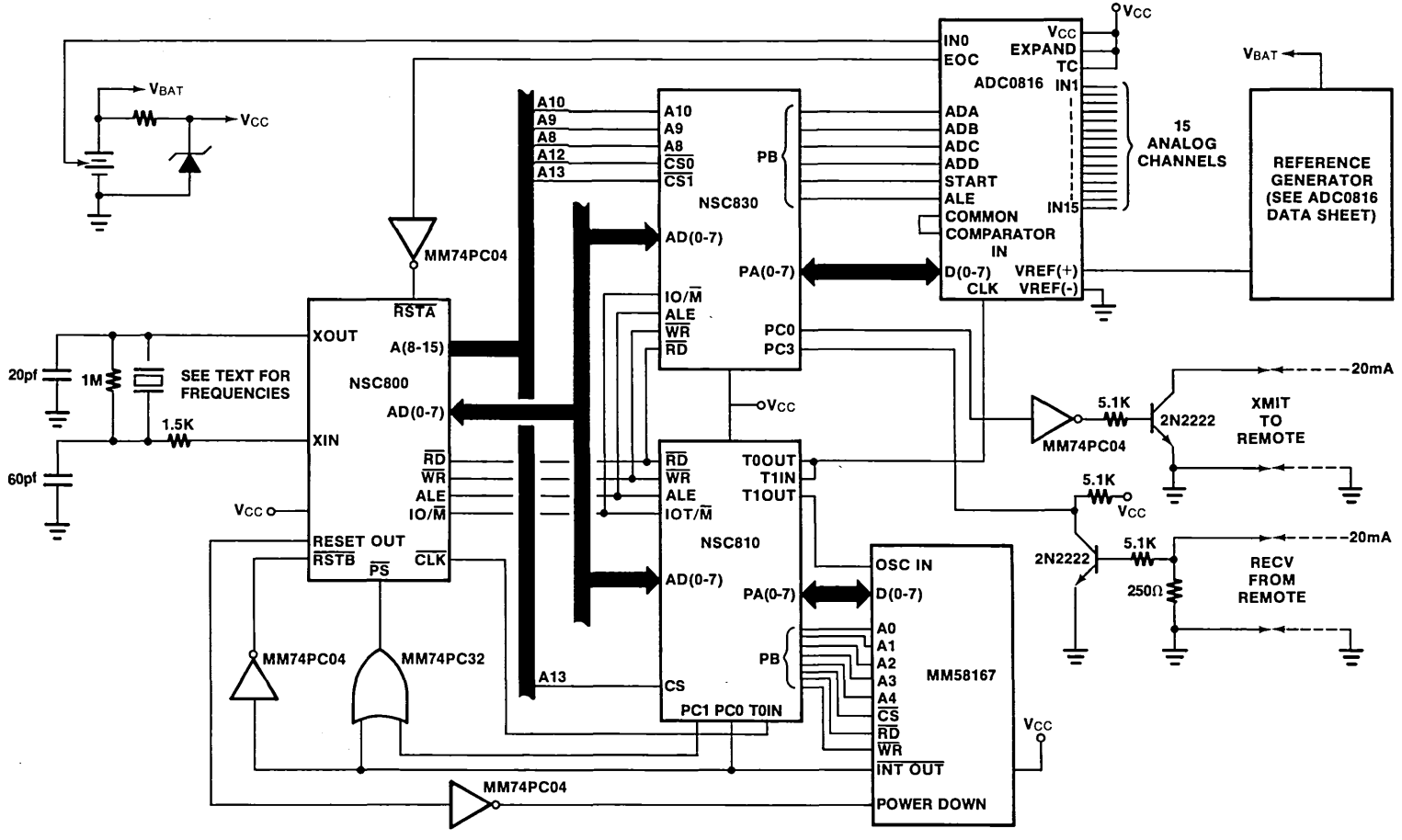


FIGURE 8-19. Remote Data Acquisition

72-20

The application is that of a remote data acquisition system. Extensive use is made of some of the other LSI CMOS components manufactured by National: notably the ADC0816 and MM58167. The ADC0816 is a 16-channel analog-to-digital converter which operates from a 5V source. The MM58167 is a microprocessor-compatible real-time clock (RTC). The schematic for this system is shown in *Figure 8-19*. All the necessary features of the system are contained in five integrated circuits: NSC800, NSC810, NSC830, ADC0816, and MM58167. Some other small scale integration P²CMOS components are used for normal interface requirements. To reduce component count, linear selection techniques are used to generate chip selects for the NSC810 and NSC830. Included also is a current loop communication link to enable the remote system to transfer data collected to a host system.

In order to keep component count low and maximize effectiveness, many of the features of the NSC800 family have been utilized. The RAM section of the NSC810 is used as a data buffer to store intermediate measurements and as scratch pad memory for calculations. Both timers contained in the NSC810 are used to produce the clocks required by the A/D converter and the RTC. The Power-Save feature of the NSC800 makes it possible to reduce system power consumption when it is not necessary to collect any data. One of the analog input channels of the A/D is connected to the battery pack to enable the CPU to monitor its own voltage supply and notify the host that a battery change is needed.

In operation, the NSC800 makes readings on various input conditions through the ADC0816. The type of devices connected to the A/D input depends on the nature of the remote environment. For example, the duties of the remote system might be to monitor temperature variations in a large building. In this case, the analog inputs would be connected to temperature transducers. If the system is situated in a process control environment, it might be monitoring fluid flow, temperatures, fluid levels, etc. In either case, operation would be necessary even if a power failure occurred, thus the need for battery operation or at least battery backup. At some fixed times or at some particular time durations, the system takes readings by selecting one of the analog input channels, commands the A/D to perform a conversion, reads the data, and then formats it for transmission; or, the system checks the readings against set points and transmits a warning if the set points are exceeded. With the addition of the RTC, the host need not command the remote system to take these readings each time it is necessary. The NSC800 could simply set up the RTC to interrupt it at a previously defined time and when the interrupt occurs, make the readings. The resultant values could be stored in the NSC810 for later correlation. In the example of temperature monitoring in a building, it might be desired to know the high and low temperatures for a 12-hour period. After compiling the information, the system could dump the data to the host over the communications link. Note from the schematic that the current for the communication link is supplied by the host to remove the constant current drain from the battery supply.

The required clocks for the two peripheral devices are generated by the two timers in the NSC810. Through the use of various divisors, the master clock generated by the NSC800 is divided down to produce the clocks. Four examples are shown in *Table 8-1*.

All the crystal frequencies are standard frequencies available from CTS Knights, Inc. The various divisors listed are selected to produce, from the master clock frequency of the NSC800, an exact 32,768 Hz clock for the MM58167 and a clock within the operating range of the A/D converter.

The MM58167 is a programmable real-time clock that is microprocessor compatible. Its data format is BCD. It allows the system to program its interrupt register to produce an interrupt output either on a time of day match (which includes the day of the week, the date and month) and/or every month, week, day, hour, minute, second, or tenth of a second. With this capability added to the system, precise time of day measurements are possible without having the CPU do timekeeping. The interrupt output can be connected, through the use of one port bit of the NSC810, to put the CPU in the power-save mode and reenable it at a preset time. The interrupt output is also connected to one of the hardware restart inputs ($\overline{\text{RSTB}}$) to enable time duration measurements. This power-down mode of operation would not be possible if the NSC800 had the duties of timekeeping. When in the power-save mode, the system power requirements are decreased by about 50%, thus extending battery life.

Communication with the peripheral devices (MM58167 and ADC0816) is accomplished through the I/O ports of

TABLE 8-1. Typical Timer Output Frequencies

Crystal Frequency	CPU Clock Output	Timer 0 Output	Timer 1 Output
2.097152 MHz	1.048576 MHz	262.144 KHz divisor = 4	32.768 KHz divisor = 8
3.276800 MHz	1.638400 MHz	327.680 KHz divisor = 5	32.768 KHz divisor = 10
4.194304 MHz	2.097152 MHz	262.144 KHz divisor = 8	32.768 KHz divisor = 8
4.915200 MHz	2.457600 MHz	491.520 KHz divisor = 5	32.768 KHz divisor = 15

the NSC810 and NSC830. The peripheral devices are not connected to the bus of the NSC800 as they are not directly compatible with a multiplexed bus structure. Therefore, additional components would be required to place them on the microprocessor bus. Writing data into the MM58167 is performed by first putting the desired data on Port A, followed by selecting the address of the internal register and applying the chip select through the use of Port B. A bit set and clear operation is performed to emulate a pulse on the bit of Port B connected to the $\overline{WR}$ input of the MM58167. For a read operation, the same sequence of operations is performed except that Port A is set for the input mode of operation and the $\overline{RD}$ line is pulsed. Similar techniques are used to read converted data from the A/D converter. When a conversion is desired, the CPU selects a channel and commands the ADC0816 to start a conversion. When the conversion is complete, the converter will produce an End-of-Conversion signal which is connected to the $\overline{RSTA}$ interrupt input of the NSC800.

When operating, the system shown consumes about 125mw. When in the power-save mode, power consumption is decreased to about 70mw. If, as is likely, the system is in the power-save mode most of the time, battery life can be quite long depending on the amp-hour rating of the batteries incorporated into the system. For example, if the battery pack is rated at 5 amp-hours, the system should be able to operate for about 400-500 hours before a battery charge or change is required.

As shown in the schematic (refer to *Figure 8-19*), analog input IN0 is connected to the battery source. In this way, the CPU can monitor its own power source and notify the host that it needs a battery replacement or charge. Since the battery source shown is a stacked array of 7 NiCads producing 8.4V, the converter input is connected in the middle so that it can take a reading on two or three of the cells. Since NiCad batteries have a relatively constant voltage output until very nearly discharged, the CPU can sense that the "knee" of the discharge curve has been reached and notify the host.

Chapter 9

Development Support

9.1 INTRODUCTION

National Semiconductor supports its microprocessors and microcomputers with a full range of publications, technical support, and products.

National's publications contain detailed component or system information; National's technical support consists of full-time training, technical support specialists, and Field Applications Engineers; and National's product support provides an interactive, versatile, and easy-to-use development system.

9.2 PUBLICATIONS

Publications are available covering the various devices manufactured by National Semiconductor. The available literature is grouped in the following categories:

- Literature Index
- Handbooks
- Manuals
- Linear Applications, Vol I and II
- Databooks
- Guides
- Product Selection Guides
- Briefs
- Individual Application Notes
- Individual Data Sheets

See appendix B for more detailed reference material. For list of currently available literature, refer to the Literature Index.

9.3 TRAINING

National Semiconductor operates a microprocessor training center in Santa Clara, Calif. The training center is fully equipped and professionally staffed to provide students with an effective mixture of hardware/software theory and hands-on laboratory experience. Courses covering our microprocessor related products are available at the National Semiconductor Training Center. To obtain information on current courses being offered and schedules, please contact:

Western Training Center
1333 Lawrence Expressway
Santa Clara, CA. 95051
(408) 737-6453

9.4 TECHNICAL SUPPORT PROGRAMS

National Semiconductor has the strongest on-the-scene technical support team - in the U.S. and abroad - of any semiconductor manufacturer. Our large network of independent sales representatives and franchised distributors is backed by our Field Application Engineers (FAE's) and microprocessor Application Engineers. The FAE's are available domestically and internationally to offer on-site technical assistance, and are equipped technically to help analyze your application, translate your needs into a viable hardware/software configuration, and then follow it through to system delivery. The microprocessor application engineers are National's home-base technical support specialists who support the FAE's in the field, and who help you use your microprocessor most effectively; they are always available to answer specific technical questions regarding the use of National's microprocessor and peripheral components.

9.5 STARPLEX DEVELOPMENT SYSTEM

National Semiconductor provides everything you need for efficient NSC800 system development. The line of available support tools is complete and has been carefully designed to make the development process as simple and speedy as possible.

The STARPLEX™ Development System (see *Figure 9-1*), provides an ideal software development environment. The powerful operating system offers a level of convenience that simply has not been available in this type of system. Comprehensive system prompts guide inexperienced users through complex tasks. Menus are presented to the user for command and function selection. System functions can be selected with single keystrokes. Also, STARPLEX offers a powerful CRT oriented editor, which eliminates the need to remember the complex commands used in standard editors. High level languages (BASIC, FORTRAN, PLM and Pascal) as well as cross assemblers are available.

STARPLEX hardware is complete. All necessary elements are included in the STARPLEX package - central processor, 64K of RAM, floppy disc drives, CRT and keyboard are standard. Options to further expand the system's capability such as double density disc or high speed line printer are available, but the basic system has everything you need to begin work.

9.6 IN-SYSTEM EMULATOR

National Semiconductor has an In-System-Emulator to support the NSC800 microprocessor. These products feature high performance as well as the flexibility required for successful debug at the hardware/software integration level. The emulator consists of three cards: trace, 32K map memory, and an NSC800 target card. This set plugs directly into STARPLEX for real-time emulation of a single NSC800 or two sets can be installed in an emulator chassis for multiprocessor emulation.

The In-System Emulator software follows the STARPLEX ease-of-use philosophy. Commands are easy to learn and can be listed in files (IN-FILE) for automatic execution of predetermined sequences. Breakpoint facilities are extremely flexible, offering 35 possible combinations including a coast feature which allows the processor to run for up to 256 additional cycles after break conditions have been met. The In-System Emulator features an NSC800 in-line assembler and disassembler for interactive patching as well as symbolic debug capability.

STARPLEX, with the In-System Emulator, provides the capability you need for timely development for your NSC800 based product. This saves you money in the long run but you don't have to wait to realize this economic advantage. Although STARPLEX with In-System Emulator offers features not found in other development system products, it costs less to own and operate than any competitive system.

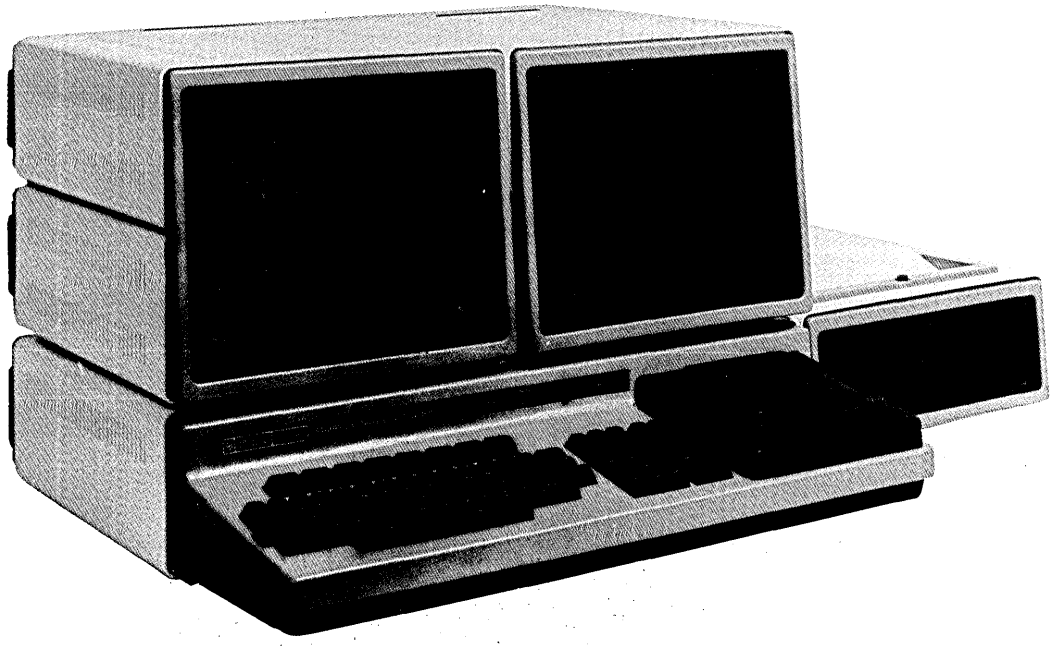


FIGURE 9-1. STARPLEX Development System

Chapter 10

General Information

10.1 INTRODUCTION

This chapter provides information relative to National Semiconductor's requirements for programming the NSC830 ROM-I/O, and, packaging protection for shipments of NSC800 Family devices.

10.2 NSC830 ORDERING INFORMATION

The information contained in *Table 10-1* must be supplied when ordering NSC830 ROM-I/O devices.

10.3 PACKAGING AND PROTECTION

The input circuits for all NSC800 Family devices are protected from static charge with diode clamps to both V_{CC} and Ground. However, the normal precautions exercised with MOS devices are recommended.

10.4 PHYSICAL DIMENSIONS

Dimensions for the NSC800 Family, and compatible devices, are included in the Data Sheets in Appendix A.

Ordering Information for Custom Programmed Parts

The following information must be submitted with each customer microcomputer program. An order will not be processed unless it is accompanied by this information. This form acts as a Traveler from Customer through Customer Service to ROM programming. Please retain a copy of this form to compare against the verification listing. The form will be sent back to the customer by Customer Service.

	National Microcomputer Part Number
	ROM Letter Code (National Use Only)
Name	Date
Address	Customer Print or I.D. No.
City State Zip	Purchase Order No.
Telephone ( )	Name of person National can contact (Print)
Authorized Signature	Date

INPUT MEDIUM

See following page for approved formats. Please check the medium you are using.

☐ **Paper Tape**

☐ **2716 EPROM**

☐ **2708 EPROM**

____**Total number of EPROMs**

OPTIONS FOR NSC830 ROM - I/O

Option 1 = ☐

CE_0 Select, enter: 0 for CE_0
1 for $\overline{CE_0}$

Option 2 = ☐

CE_1/IOR Select, enter: 0 for $\overline{IOR}$
1for CE_1
2 for $\overline{CE_1}$

TABLE 10-1. Required Ordering Information

APPROVED FORMATS FOR CUSTOM PROGRAMMED PARTS

INPUT MEDIUM:

2716 EPROM
2708 EPROM
PAPER TAPE

IMPORTANT - EPROM LABELLING

Only one customer program may be included in a single order. The following method must be used to identify the EPROMs comprising a program.

a) The EPROMs used for storing a custom program are designated as shown:

2716:	Block A	0-2047
2708:	Block A	0-1023
	Block B	1024-2047

b) All EPROMs must be labelled (stickers, paint, etc.) with this block designation plus a customer assigned print or identification number.

Example:

1) Customer Data
   - Custom Program Length - 2K
   - Medium - Two 2708's
   - Customer Print or I.D. No. C123-45

2) EPROM Labels

C123-45 A 0-1023	C123-45 B 1024-2047

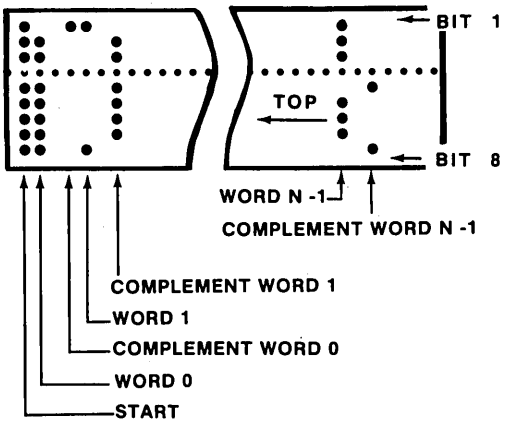
Paper Tape

Tapes may only be submitted in binary complement format. The following information should be written on the paper tape.

Company Name
Customer Print or I.D. No.
NSC Part No.
A Punch = ("1" or "0")
This is _____ logic (POS or NEG)

NOTE: *This information is provided in the NSC830 Data Sheet in Appendix A.*

BINARY COMPLEMENT FORMAT



NOTE 1: Tape must be blank except for the data words.
NOTE 2: Tape must start with a rubout character.
NOTE 3: Data is comprised of two words, the first being the actual data and the second being the complement of the data.

Verification

You will receive a listing of the options ordered and the input data. If you also wish to receive EPROMs for verification, please send additional blank EPROMs as necessary for this purpose. You can use software (the listing) or hardware (EPROMs) to verify the program.

You will be asked for a GO/NO GO response within one week after you receive the listing.

VERIFICATION LISTING

The verification listing has six sections:

1. A cover sheet with provision for "STOP, DO NOT PROCEED" or "VERIFICATION CERTIFIED" signatures.
2. Description of the options you have chosen.
3. A description of the log designations and assumptions used to process the data.
4. A listing of the data you have submitted.
5. An error summary.
6. A definition of the standard logic definitions for the ROM and the reduced form of the data. This list shows the output word corresponding to each address coded in binary.

Appendix A

Data Sheets

National Semiconductor

Preliminary
January 1981

NSC800 High-Performance Low-Power Microprocessor

General Description

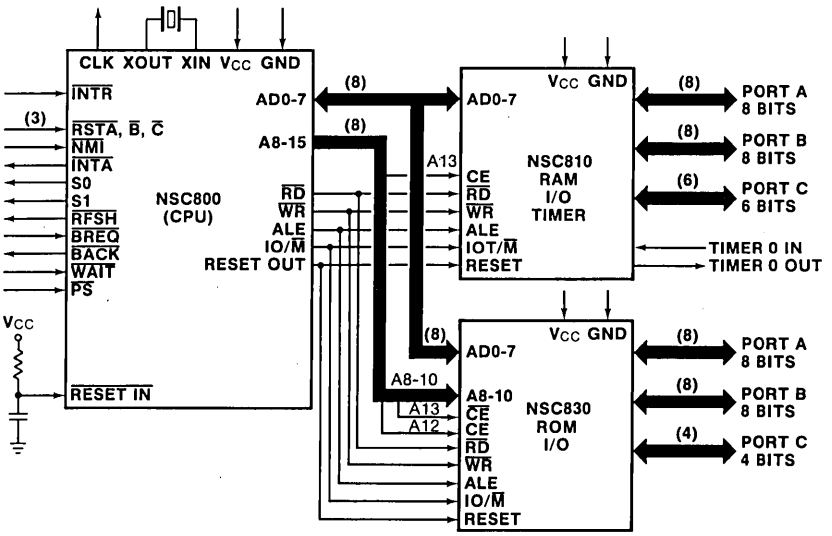
The NSC800 is an 8-bit microprocessor that functions as the central processing unit (CPU) in National Semiconductor's NSC800 microcomputer family. The device is fabricated using National's P²CMOS technology. This technology provides the system designer with devices that equal the performance levels of comparable NMOS products, combined with the low-power advantages of CMOS. Many system functions are incorporated on the device, such as: vectored priority interrupts, refresh control, power-save feature and interrupt acknowledge. The NSC800 is housed in a 40 pin, dual-in-line package.

Dedicated memories (NSC810 RAM-I/O Timer and NSC830 ROM-I/O) have on-chip logic for direct interface to the NSC800. In addition, National also offers a full line of P²CMOS and CMOS components to allow a full low-power solution to system designs.

Features

- Single 5V Power Supply
- Fully Compatible with Z80™ Instruction Set
- Powerful Set of 158 Instructions
- 10 Addressing Modes
- 22 Internal Registers
- Low Power: 50mW at 5V V_{CC}
- Multiplexed Bus Structure
- On-Chip Bus Controller and Clock Generator
- On-Chip 8-Bit Dynamic RAM Refresh Circuitry
- Standard Speed: 1.6 μs Instruction Cycle at 2.5 MHz
- Fast Version (NSC800A): 1 μs Instruction Cycle at 4 MHz (Availability to be Announced)
- Capable of Addressing 64K Bytes of Memory and 256 I/O devices
- Five Interrupt Request Lines On-Chip
- Schmitt Trigger Input on Reset
- Unique Standby-Current (Power Save) Feature

NSC800 Microcomputer Family Block Diagram



45-1

rμPG - FL5M10/Printed in U.S.A.

Absolute Maximum Ratings

Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... -0.3 to V_{CC} + 0.3V
Lead Temperature (Soldering 10 Seconds) .....300°C
Power Dissipation ..... 1 W

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40°C to +85°C	5V
Commercial	0°C to +70°C	5V

NOTE: *Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC Electrical Characteristics.*

DC Electrical Characteristics

T_A = 0 to 70°C, V_{CC} = 5V ± 10%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}	Logical 1 Input Voltage	0.7 V_{CC}		V_{CC}	V	
V_{IL}	Logical 0 Input Voltage	0		0.3 V_{CC}	V	
V_{HY}	Hysteresis at RESET IN input		0.5		V	V_{CC} = 5V
V_{OH1}	Logical 1 Output Voltage	2.4			V	I_{OUT} = -1.0 mA
V_{OH2}	Logical 1 Output Voltage	V_{CC}-0.5			V	I_{OUT} = -10 μA
V_{OL1}	Logical 0 Output Voltage	0		0.4	V	I_{OL} = 2 mA
V_{OL2}	Logical 0 Output Voltage	0		0.1	V	I_{OUT} = 10 μA
I_{IL}	Input Leakage Current	-1.0		1.0	μA	$0 \le V_{IN} \le V_{CC}$
I_{OL}	Output Leakage Current	-1.0		1.0	μA	$0 \le V_{IN} \le V_{CC}$
I_{CC}	Active Supply Current		10		mA	I_{OUT} = 0; $f_{(XIN)}$ = 5MHz

*Availability of Military temperature range components to be announced.

**Availability of extended operating voltage range to be announced.

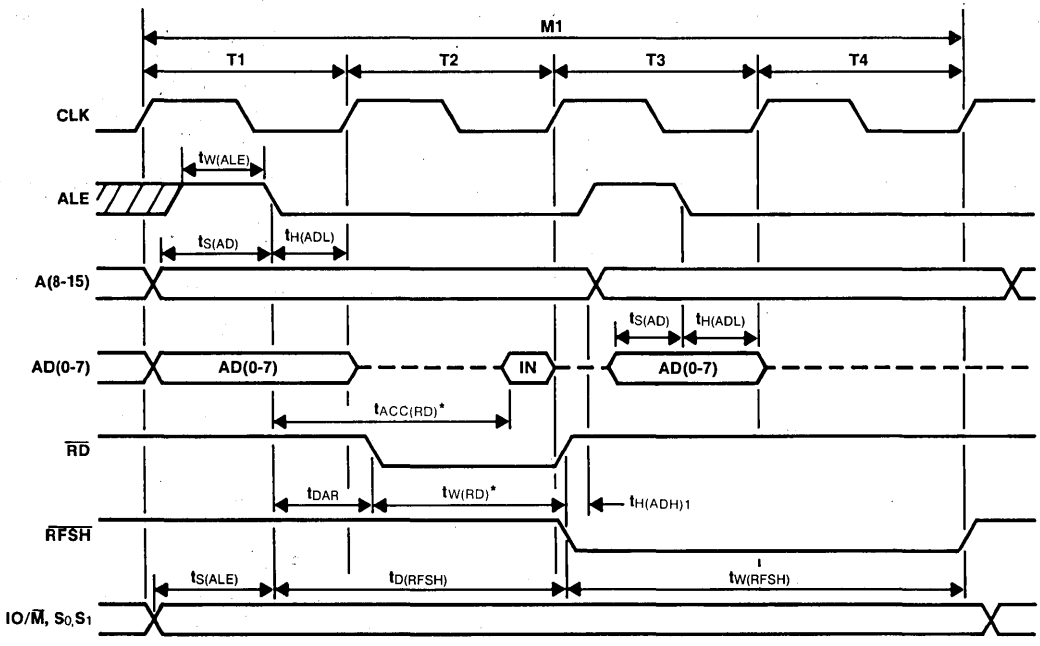
AC Electrical Characteristics

T_A = 0 to 70°C, V_{CC} = 5V ± 10%, GND = 0V

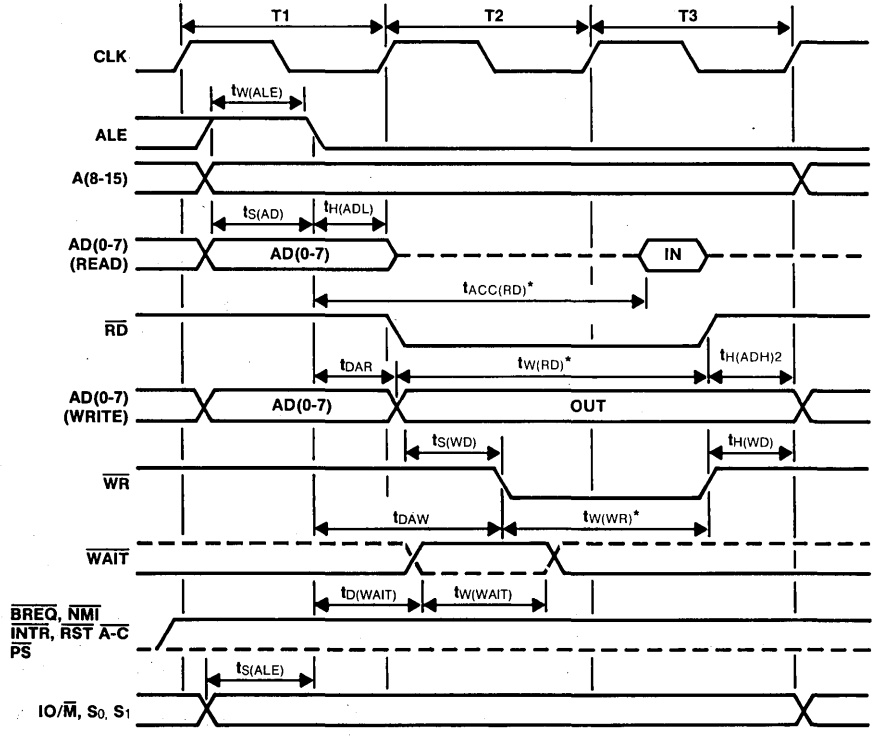
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions (Note 1)
t_X	Period at XIN and XOUT pins		200		ns	
t	Period at "CLK" output (= 2 t_X)		400		ns	
$t_{S(AD)}$	Address Set Up Time		$\frac{t}{2}$ - 75		ns	
$t_{H(ADL)}$	Address (0-7) Hold Time		$\frac{t}{2}$ - 75		ns	
$t_{W(ALE)}$	ALE Width		$\frac{t}{2}$ - 50		ns	
t_{DAR}	Falling Edge of ALE to Leading Edge of Read Strobe		$\frac{t}{2}$ - 50		ns	
$t_{ACC(RD)}$	Read Access Time (Note 2)		$\frac{3t}{2}$ - 125		ns	Add t/2 for Memory Read Cycles
$t_{W(RD)}$	Read Strobe Width During Op-code Fetch. (Note 2)		t - 50		ns	Add t/2 for Memory Read Cycles
$t_{H(ADH)1}$	Address (8-15) Hold Time During Opcode Fetch.		0		ns	
$t_{S(WD)}$	Write Data Setup Time		$\frac{t}{2}$ - 75		ns	
$t_{H(WD)}$	Write Data Hold Time		$\frac{t}{2}$ - 75		ns	
t_{DAW}	Falling Edge of ALE to Leading Edge of WR		t - 100		ns	
$t_{W(WR)}$	Write Strobe Width (Note 2)		t - 50		ns	
$t_{H(ADH)2}$	Address (8-15) Hold Time during Memory or I-O Read and Write		$\frac{t}{2}$ - 75		ns	
$t_{D(RFSH)}$	Falling Edge of ALE to Leading Edge of Refresh Strobe (Note 2)		$\frac{3t}{2}$ + 50		ns	
$t_{W(RFSH)}$	Refresh Strobe Width		2t - 75		ns	
$t_{D(WAIT)}$	Falling Edge of ALE to Wait Input Valid		t		ns	
$t_{W(WAIT)}$	Wait Input Width		$\frac{t}{2}$ + 50		ns	
$t_{D(I)}$	Falling Edge of ALE to $\overline{INTR}$, $\overline{NMI}$, $\overline{RST}$ A-C, $\overline{PS}$ & $\overline{BREQ}$ Inputs Valid (Note 2)		t			Add t for Opcode Fetch Cycles
$t_{W(I)}$	Width of $\overline{INTR}$, $\overline{NMI}$, $\overline{RST}$ A-C, $\overline{PS}$ & $\overline{BREQ}$ Inputs		$\frac{t}{2}$ + 50		ns	
t_{DAI}	Falling Edge of ALE to Interrupt Acknowledge Strobe		$\frac{t}{2}$ - 50		ns	
$t_{W(INTA)}$	Interrupt Acknowledge Strobe Width (Note 2)		3t - 50		ns	(*Figure 10*)
$t_{D(BACK)1}$	Falling Edge of ALE to Falling Edge of BUS Acknowledge Output (Note 2)		$\frac{5t}{2}$ + 50		ns	Add t for Opcode Fetch Cycles
$t_{D(BACK)2}$	Rising Edge of Bus Request to Rising Edge of BUS Acknowledge	$\frac{t}{2}$		$\frac{3t}{2}$ + 50	ns	(*Figure 6*)
t_{DPA}	Rising Edge of Power Save to Falling Edge of ALE	$\frac{t}{2}$		$\frac{3t}{2}$ + 50	ns	(*Figure 12*)
t_S (ALE)	Setup Time		$\frac{t}{2}$ - 75		ns	

Notes: **1.** Output Capacitive loading = 100pf; **2.** Add t for each WAIT state

Timing Waveforms



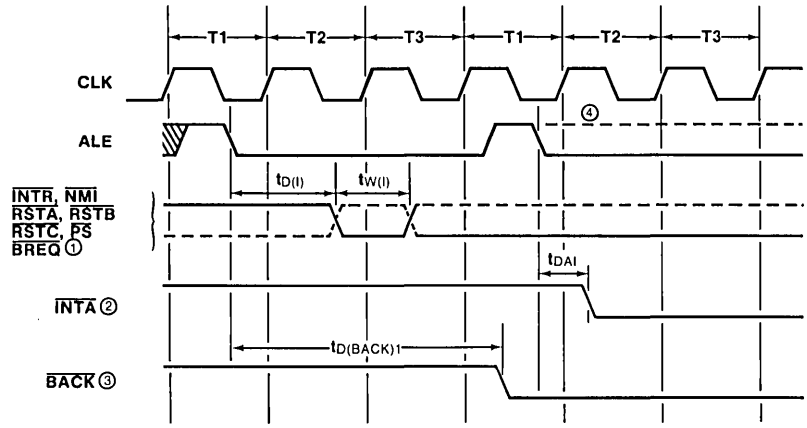
Opcode Fetch Cycle 45-18



***NOTES:** 1. Output capacitive loading = 100pf
2. Add t for each WAIT state

Memory Read and Write Cycle 45-19

Timing Reference



Timing reference diagram used for specifying parameters related to $\overline{\text{INTR}}$, $\overline{\text{NMI}}$, $\overline{\text{RSTA}}$, $\overline{\text{RSTB}}$, $\overline{\text{RSTC}}$, $\overline{\text{BREQ}}$, PS inputs and $\overline{\text{BACK}}$, $\overline{\text{INTA}}$ outputs

Notes: 1. $\overline{\text{BREQ}}$ input recognized during the last T-State of any M-cycle
2. Response to $\overline{\text{INTR}}$ input. See *Figure 10.*
3. Response to $\overline{\text{BREQ}}$ input. See *Figure 6.*
4. Response to $\overline{\text{PS}}$ input. See *Figure 12.*

NSC800 Functional Pin Descriptions

The following describes the function of all NSC800 input/output pins. Some of these descriptions reference internal circuits.

INPUT SIGNALS

Reset Input ($\overline{\text{RESET IN}}$): Active low. Sets A (8-15) and AD (0-7) to TRI-STATE® (high impedance). Clears the contents of PC, I and R registers, disables interrupts, and causes a reset output to be activated.

Bus Request ($\overline{\text{BREQ}}$): Active low. Used when another device is requesting the system bus. $\overline{\text{BREQ}}$ is recognized at the end of the current machine cycle, then A(8-15), AD (0-7), IO/$\overline{\text{M}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ are set to the high impedance mode and the request is acknowledged via the $\overline{\text{BACK}}$ output signal.

Non-Maskable Interrupt ($\overline{\text{NMI}}$): Active low. The non-maskable interrupt, generated by the peripheral device(s), is the highest priority interrupt request line. Input is only recognized at the end of the current instruction. Its execution is independent of the interrupt enable flip-flop. $\overline{\text{NMI}}$ execution involves saving the PC on the stack and automatic branching to restart address X'0066 in memory.

Restart Interrupts A, B, C, ($\overline{\text{RSTA}}$, $\overline{\text{RSTB}}$, $\overline{\text{RSTC}}$): Active low. Restarts generated by the peripherals are recognized at the end of the current instruction if their respective interrupt enable bits (and IFF1/IFF2) are set. Execution is identical to $\overline{\text{NMI}}$ except interrupts are enabled for the following restart addresses:

NAME	RESTART ADDRESS (X')
$\overline{\text{RSTA}}$	003C
$\overline{\text{RSTB}}$	0034
$\overline{\text{RSTC}}$	002C

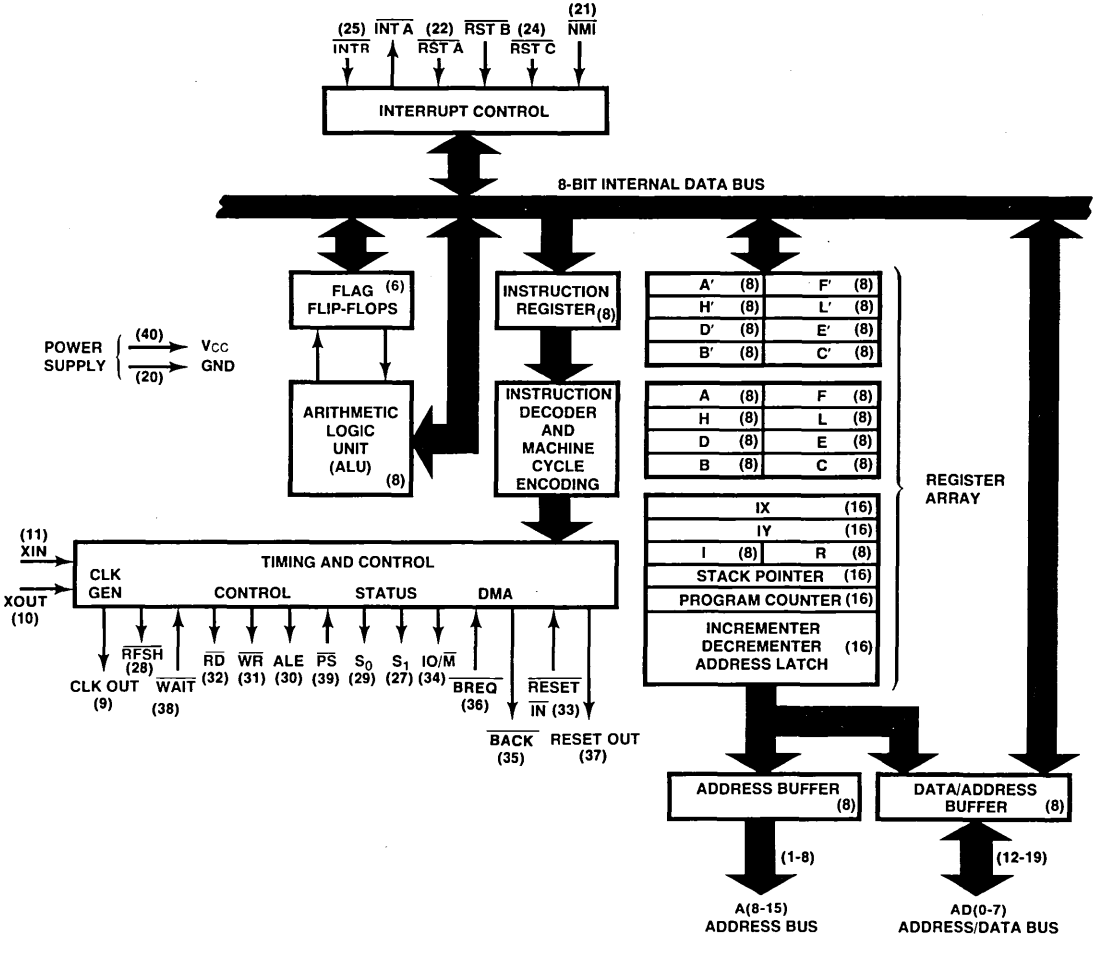
The order of priority is fixed (highest first) as follows:

1) $\overline{\text{RSTA}}$, 2) $\overline{\text{RSTB}}$, 3) $\overline{\text{RSTC}}$

Interrupt Request ($\overline{\text{INTR}}$): Active low. An interrupt request input generated by a peripheral device is recognized at the end of the current instruction provided that the interrupt enable and master interrupt enable bits are set. $\overline{\text{INTR}}$ is the lowest priority interrupt request input. Under program control, $\overline{\text{INTR}}$ can be executed in three distinct modes in conjunction with the $\overline{\text{INTA}}$ output.

Wait ($\overline{\text{WAIT}}$): Active low. When set low during $\overline{\text{RD}}$ or $\overline{\text{WR}}$, the CPU extends its machine cycle in increments of t (wait) states. The wait machine cycle continues until the $\overline{\text{WAIT}}$ input returns high.

Power Save ($\overline{\text{PS}}$): Active low. $\overline{\text{PS}}$ is sampled at the end of the current instruction cycle. When $\overline{\text{PS}}$ is low, the CPU stops executing at the end of current instruction and keeps itself in the low-power mode. Normal operation resumes when $\overline{\text{PS}}$ is returned high.



NSC800 CPU Functional Block Diagram

72-1

OUTPUT SIGNALS

Bus Acknowledge ($\overline{BACK}$): Active low. $\overline{BACK}$ indicates to the bus requesting device that the CPU bus and its control signals are in the TRI-STATE mode. The requesting device may then take control of the bus and its control signals.

Address Bits 8-15 [A(8-15)]: Active high. These are the most significant 8 bits of the memory address bus, or of the input/output address. During a $\overline{BREQ}/\overline{BACK}$ cycle, the A (8-15) bus is in the TRI-STATE mode.

Reset Out (RESET OUT): Active high. When RESET OUT is high, it indicates the CPU is being reset. The signal is normally used to reset the peripheral devices.

Input/Output/Memory (IO/$\overline{M}$): An active high on the IO/$\overline{M}$ output signifies that the current machine cycle is relative to an input/output device. An active low on the IO/$\overline{M}$ output signifies that the current machine cycle is relative to memory. It is TRI-STATE during $\overline{BREQ}/\overline{BACK}$ cycles.

Refresh ($\overline{RFSH}$): Active low. The refresh output indicates that the dynamic RAM refresh cycle is in progress. $\overline{RFSH}$ goes low during T3 and T4 states of all M1 cycles.

Address Latch Enable (ALE): ALE is active only during the T1 state of M cycles and T3 state of M1 cycles. The high to low transition of ALE indicates that a valid memory /I-O/refresh address is available on the AD (0-7) lines.

Read Strobe ($\overline{RD}$): Active low. On the trailing edge of the $\overline{RD}$ strobe, data are input to the CPU via the AD (0-7) lines. The $\overline{RD}$ line is in the TRI-STATE mode during $\overline{BREQ}/\overline{BACK}$ cycles.

Write Strobe ($\overline{WR}$): While the $\overline{WR}$ line is low, valid data are output by the CPU on the AD (0-7) lines. The $\overline{WR}$ line is in the TRI-STATE mode during $\overline{BREQ}/\overline{BACK}$ cycles.

Clock (CLK): CLK is an output provided for use as a system clock. The CLK output is a square wave at one half the input frequency.

Interrupt Acknowledge ($\overline{INTA}$): Active low. The interrupt acknowledge output is activated in the M1 cycle (S) immediately following the t state in which the $\overline{INTR}$ input is recognized. (Output is normally used to gate the interrupt response vector from the peripheral controller onto the AD (0-7) lines). It is used in two of the three interrupt modes. In mode 0, an instruction is gated onto the AD (0-7) line during $\overline{INTA}$. In mode 2, a single interrupt response vector is gated onto the data bus.

Status (S0, S1): Bus status outputs indicate encoded information regarding the ensuing M cycle as follows:

S1	S0	STATE
0	0	HALT
0	1	WRITE
1	0	READ
1	1	(OPCODE) FETCH

INPUT/OUTPUT SIGNALS

Power (V_{CC}): +5-volt supply.

Ground (GND): 0-volt reference.

Crystal or R-C (XIN, XOUT): XIN may be used as an external clock input.

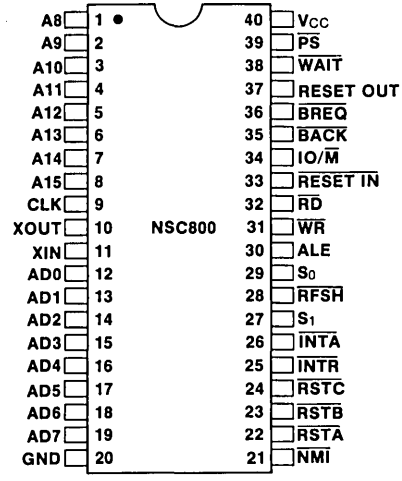
Multiplexed Address/Data (AD 0-7): Active High

At $\overline{RD}$ Time:	Input data to CPU
At $\overline{WR}$ Time:	Output data from CPU
At Falling Edge of ALE Time:	Least significant byte of address during memory reference cycle. 8-bit port address during I/O reference cycle.
During $\overline{BREQ}/\overline{BACK}$ Cycle:	High Impedance

INPUT PROTECTION

All inputs are protected from static charge with diode clamps to both V_{CC} and GND. Normal precautions taken with MOS devices are recommended.

PIN CONFIGURATION



45-6

TIMING CONTROL

All necessary timing signals are provided by a single state inverter oscillator contained on the NSC800 chip. The chip operation frequency is equal to one half of the frequency of this oscillator. The oscillator frequency can be controlled by one of the following methods:

1. Leaving the XOUT pin unterminated and driving the XIN pin with an externally generated clock as shown in *Figure 1A.*

2. Connecting a resistor capacitor feedback network between the XIN and XOUT pins as shown in *Figure 1B.*

3. Connecting a crystal with the proper biasing network between XIN and XOUT as shown in *Figure 1C.*

FUNCTIONAL DESCRIPTION

The NSC800 is an 8-bit general-purpose microprocessor designed for stand-alone and DMA (direct memory access) applications. A minimum system can be constructed with an NSC800, an NSC810 (RAM/IO/Timer) and an NSC830 (ROM/IO).

NSC800 uses a multiplexed bus for data and addresses. The 16-bit address bus is divided into a high-order 8-bit address bus that handles bits 8-15 of the address, and a low-order 8-bit multiplexed address/data bus that handles bits 0-7 of the address and bits 0-7 of the data. Strobe outputs from the NSC800 (ALE, $\overline{RD}$ and $\overline{WR}$) indicate when a valid address or data is present on the bus. IO/$\overline{M}$ indicates whether the ensuing cycle accesses memory or I/O.

Figure 2 illustrates the timing relationship for OP code fetch cycles with and without a wait state. *Figure 3* illustrates the timing relationship for memory read and write cycles with and without wait state. Input/output cycles with and without a wait state are shown in *Figure 4.*

INITIALIZATION

The NSC800 and its peripheral components are initialized by $\overline{RESET\ IN}$ and RESET OUT. $\overline{RESET\ IN}$ input is associated with an on-chip Schmitt trigger that facilitates using an R-C network power-on reset scheme (*Figure 5*).

To ensure proper power-up conditions for the NSC800, the following power up and initialization procedure is recommended:

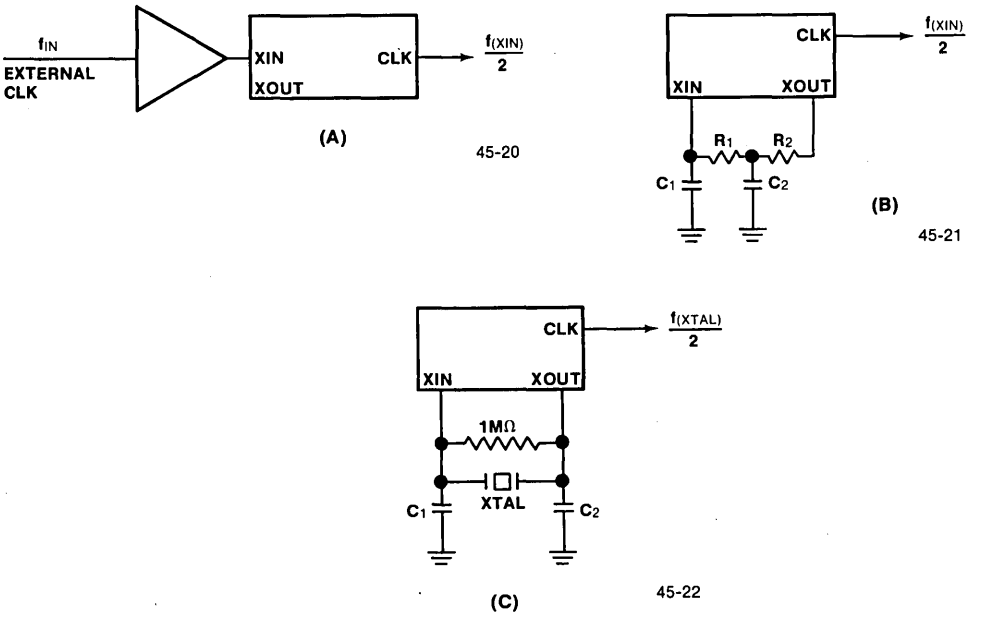


FIGURE 1. Timing Control Configurations

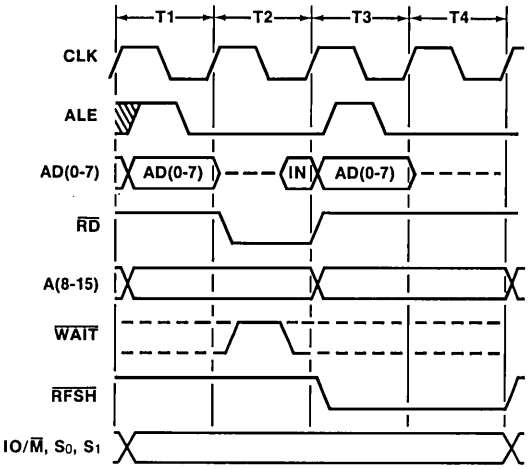


FIGURE 2A. Op Code Fetch Cycles Without Wait States

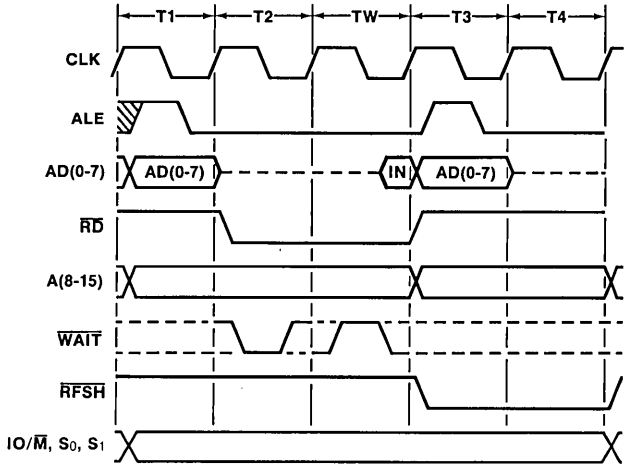


FIGURE 2B. Op Code Fetch Cycles With Wait States

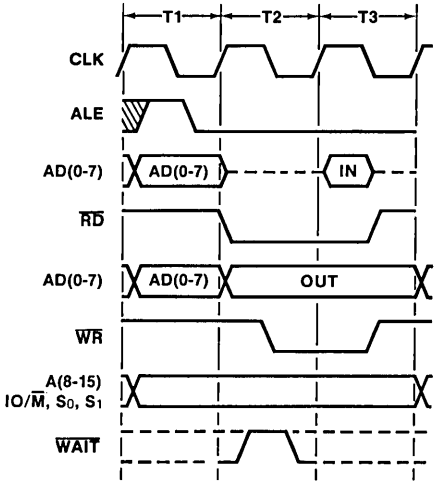


FIGURE 3A. Memory Read/Write Cycles Without Wait States

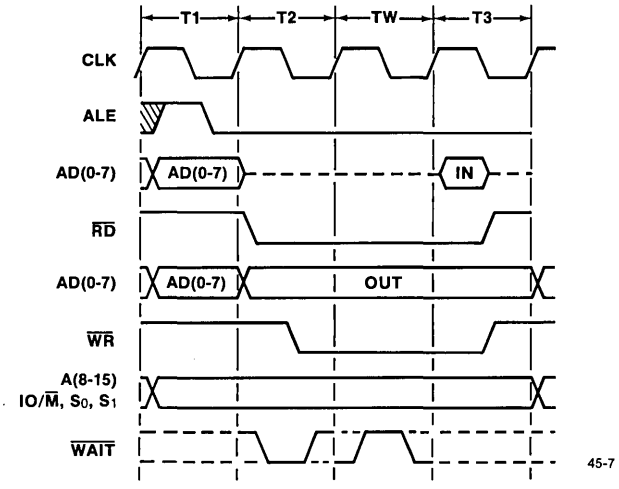


FIGURE 3B. Memory Read and Write With Wait States

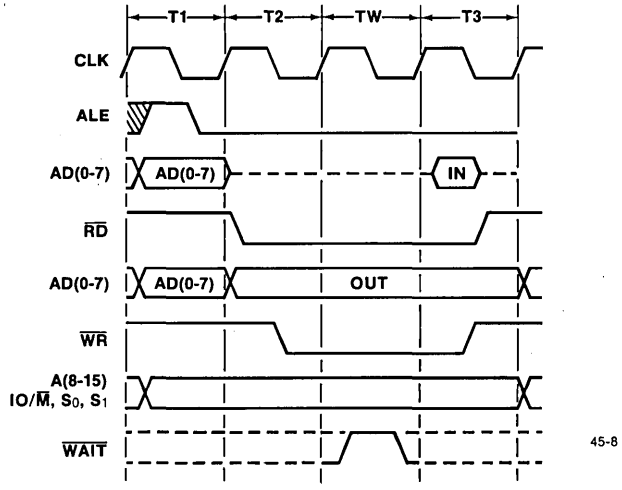
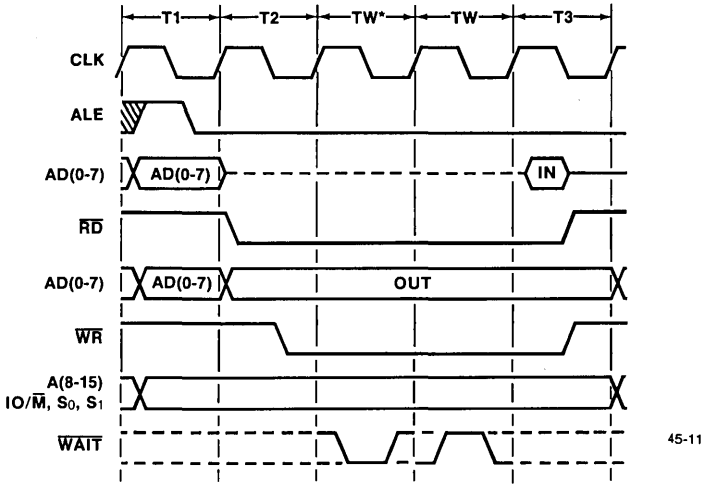


FIGURE 4A. Input and output Cycles Without Wait States



*WAIT State User Generated

FIGURE 4B. Input and Output Cycles With Wait States

1. Apply power (V_{CC} and GND) and set $\overline{\text{RESET}}$ IN active (low). Allow sufficient time (approximately 100 ms if crystal used) for the oscillator and internal clocks to stabilize. $\overline{\text{RESET}}$ $\overline{\text{IN}}$ must remain low for at least 3 t state (CLK) times. RESET OUT, following the clock stabilization period, responds by going high, indicating to the system that the NSC800 is being reset. RESET OUT signal becomes available to reset the peripherals.

2. Set $\overline{\text{RESET}}$ $\overline{\text{IN}}$ high following which the RESET OUT goes low and the CPU initiates the first opcode fetch cycle.

NOTE: *The NSC800 initialization includes: Clear PC to X'0000 (the first opcode fetch, therefore, is from memory location X'0000). Clear registers I (Interrupt Vector Base) and R (Refresh Counter) to X'00. Clear interrupt control register bits IEA, IEB and IEC. The interrupt control bit IEI is set to 1 to maintain INS8080A/Z80A™ compatability (see interrupts for more details). Maskable Interrupts are disabled and the CPU enters Interrupt Mode 0. While $\overline{\text{RESET}}$ $\overline{\text{IN}}$ is active (low), the A (8-15), and AD (0-7) lines go to high impedance (TRI-STATE) and all CPU strobes go to the inactive state.*

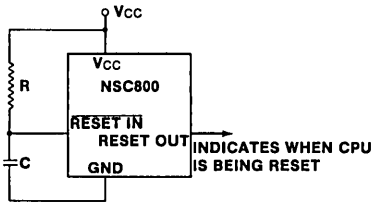
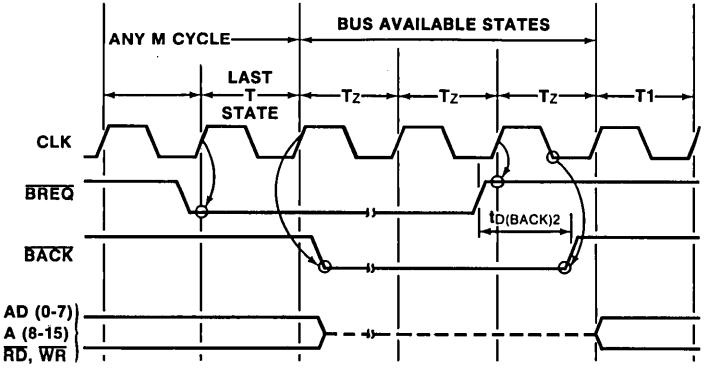


FIGURE 5. Power On Reset

45-5

BUS ACCESS CONTROL

Figure 6 illustrates Bus access control in the NSC800. The external device controller produces an active $\overline{\text{BREQ}}$ signal that requests the bus. When the CPU responds with $\overline{\text{BACK}}$ then the bus and related control strobes go to high impedance (TRI-STATE). It should be noted that (1) $\overline{\text{BREQ}}$ is sampled at the last t state of any M machine cycle only. (2) the NSC800 will not acknowledge any interrupt/restart requests, and will not perform any dynamic RAM refresh functions until after $\overline{\text{BREQ}}$ input signal is inactive high. (3) $\overline{\text{BREQ}}$ signal has priority over all interrupt request signals, should $\overline{\text{BREQ}}$ and interrupt request occur active simultaneously.



45-10

FIGURE 6. Bus Acknowledge Cycle

T_Z = Time states Bus and control signals are in high impedance mode.

REGISTER CONFIGURATION

The NSC800 contains 22 programmable registers as shown in *Figure 7*. The CPU working registers are arranged in two, 8-register configurations, each of which includes an 8-bit accumulator, a flag register, and six general purpose 8-bit registers. Only one, 8-bit register set may be active at any given moment in time. However, simple instructions exist that allow the programmer to exchange the active and alternate register sets.

It should also be noted that the six, 8-bit general purpose registers (B, C, D, E, H, and L) can be accessed as 16-bit registers (BC, DE, and HL). The functions of these become apparent in the instruction set description.

CPU Main Working Register Set

Accumulator A	(8)	Flags F	(8)
Register B	(8)	Register C	(8)
Register D	(8)	Register E	(8)
Register H	(8)	Register L	(8)

CPU Alternate Working Register Set

Accumulator A'	(8)	Flags F'	(8)
Register B'	(8)	Register C'	(8)
Register D	(8)	Register E'	(8)
Register H'	(8)	Register L'	(8)

CPU Dedicated Registers

Index Register IX	(16)
Index Register IY	(16)
Interrupt Vector Register I	(8)
Memory Refresh Register R	(8)
Stack Pointer SP	(16)
Program Counter PC	(16)

FIGURE 7. Register Configuration

DEDICATED REGISTERS

1. **Program Counter (PC)**. The program counter contains the 16-bit address of the current instruction being fetched from memory. The PC is incremented after its contents have been transferred to the address lines. When a program jump occurs, the new address is placed in the PC, overriding the incrementer.

2. **Stack Pointer (SP)**. The stack pointer contains the 16-bit address of the current top of a stack located in external system RAM memory. The external stack memory is organized as a last-in, first-out (LIFO) file. The stack allows simple implementation of multiple level interrupts, virtually unlimited subroutine nesting and simplification of many types of data manipulation.

3. **Index Registers (IX & IY)**. The two,16-bit index registers hold a 16-bit base address used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's complement signed integer.

4. **Interrupt Page Address Register (I)**. The NSC800 CPU can indirectly call any memory location in response to a mode 2 interrupt. The I Register is used to store the high order 8 bits of the address. The low order 8 bits are supplied by the interrupting peripheral. This feature allows interrupt routines to be dynamically located anywhere in memory with minimal access time to the routine.

5. **Memory Refresh Register (R)**. The NSC800 CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. This 8-bit register is automatically incremented after each instruction fetch. The data in the refresh counter is sent out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer.

ACCUMULATORS AND FLAG REGISTERS

The CPU includes two, 8-bit accumulators and two associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operation. The flag register indicates specific conditions for 8-bit or 16-bit operations.

FLAG REGISTERS (F,F')

The two NSC800 Flag registers each contain six status bits that are set or reset (cleared) by various CPU operations (see *Figure 8*). Four of these bits, the Carry, Zero, Sign, and Parity/Overflow Flags, can be tested by the programmer. The descriptions of the flags follow.

1. **Carry Flag (C)** - This flag is set by the carry from the highest order bit of the accumulator during an add instruction or a borrow generated during a subtraction instruction. Specific shift and rotate instructions also affect this bit.

2. **Zero Flag (Z)** - This flag is set when a zero is loaded into the accumulator as a result of an operation. Otherwise it remains clear.

3. **Sign Flag (S)** - This flag stores the state of bit 7 (the sign bit) in the accumulator after an arithmetic operation. This flag is intended to be used with signed numbers.

4. **Parity/Overflow Flag (P/V)** - During logical operations this flag is set when the parity of the result is even and reset when it is odd. It represents overflow when signed two's complement arithmetic operations are performed. An overflow occurs when the resultant of a two's complement operation (in the accumulator) is out of range.

The two non-testable flag register bits used for BCD arithmetic are:

1. **Half carry (H)** - This flag indicates a BCD carry or borrow result from the least significant four bits of an operation; when using the DAA (Decimal Adjust Instruction), it is used to correct the result of a previously packed decimal add or subtract.

2. **Add/Subtract Flag (N)** - Since the algorithm for correcting BCD operations is different for addition or subtraction, this flag specifies what type of instruction was executed last in order that the DAA operation will be correct for either operation.

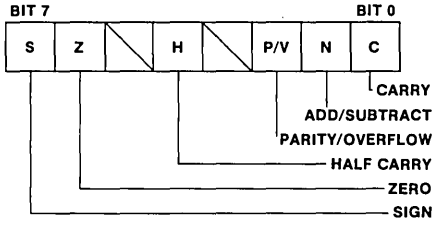


FIGURE 8. Flag Register

INTERRUPT

The NSC800 has five interrupt/restart inputs, four are maskable ($\overline{RSTA}$, $\overline{RSTB}$, $\overline{RSTC}$, and $\overline{INTR}$) and one is non-maskable ($\overline{NMI}$). $\overline{NMI}$ (see *Figure 9*), having the highest priority of all interrupts, is always serviced and cannot be disabled by the user. After recognizing an active input on $\overline{NMI}$, the CPU ignores the next instruction, pushes the PC onto the stack, and jumps to address X'0066, where the users interrupt sevice routine is located (i.e. restart to memory location X'0066). $\overline{NMI}$ is intended for interrupts requiring immediate attention, such as power down, control panel, etc.

$\overline{RSTA}$, $\overline{RSTB}$ and $\overline{RSTC}$ are restart inputs, which, if enabled, execute a restart to memory location X'003C, X'0034, and X'002C, respectively. Note that the CPU response to the $\overline{NMI}$ and $\overline{RST}$ ($\overline{A}$,$\overline{B}$,$\overline{C}$) request input is basically identical. Unlike $\overline{NMI}$, however, restart request inputs must be enabled. The NSC800 also provides one more general-purpose interrupt request input, $\overline{INTR}$. When enabled, the CPU responds to $\overline{INTR}$ in one of the three modes defined by instruction IM0, IM1, and IM2 for modes, 0, 1, and 2, respectively. Following reset, the CPU automatically sets itself in mode 0.

Interrupt ($\overline{INTR}$) mode 0: Identical to INS8080A mode. The CPU responds to an interrupt request by providing an $\overline{INTA}$ (interrupt acknowledge) strobe, which can be used to gate an instruction from a peripheral onto the data bus. The CPU follows by executing the instruction. Normally this instruction is a one byte call (Restart instruction) or a three-byte call (Call nn instruction). Two wait states are automatically inserted by the CPU during $\overline{INTA}$ cycle to allow the interrupting device (or its controller) ample time to gate the instruction and determine external priorities. (*Figure 10*).

Interrupt ($\overline{INTR}$) mode 1: Similar to an NMI interrupt except the restart location is X'0038.

Interrupt ($\overline{INTR}$) mode 2: With this mode, the programmer maintains a table that contains the 16-bit starting address of every interrupt service routine. This table may be located anywhere in memory. When the mode 2 interrupt is accepted (see *Figure 11*), a 16-bit pointer must be formed to obtain the desired interrupt service routine starting address from the table. The upper 8 bits of this pointer are from the contents of the I register, which has been previously loaded with the desired value by the programmer. The lower 8 bits of the pointer are supplied by the interrupting device with the low order bit forced to zero. The pointer is used to get two adjacent bytes from the interrupt service routine starting address table to complete 16-bit service routine starting address. The first byte of each entry in the table is the least significant (low order) portion of the address. The programmer must obviously fill this table with the desired addresses before any interrupts are to be accepted.

Note that this table can be changed at any time to allow peripherals to be serviced by different service routines. Once the interrupting device supplies the lower portion of the pointer, the CPU automatically pushes the program counter onto the stack, obtains the starting address from the table and does a jump to this address.

$\overline{NMI}$, $\overline{RST}$, $\overline{A}$, $\overline{B}$,$\overline{C}$, and $\overline{INTR}$ have fixed priorities built into the NSC800 as:

$\overline{NMI}$	**(Highest Priority)**
$\overline{RSTA}$	
$\overline{RSTB}$	
$\overline{RSTC}$	
$\overline{INTR}$	**(Lowest Priority)**

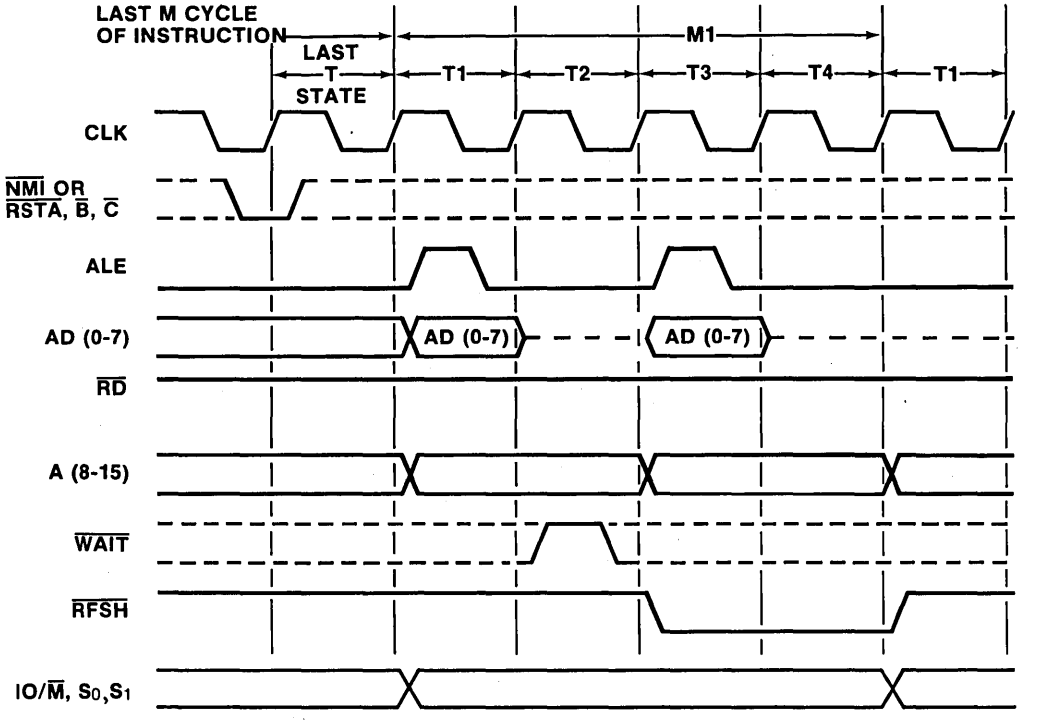


FIGURE 9. Non-Maskable Restart Interrupt Cycle

45-13

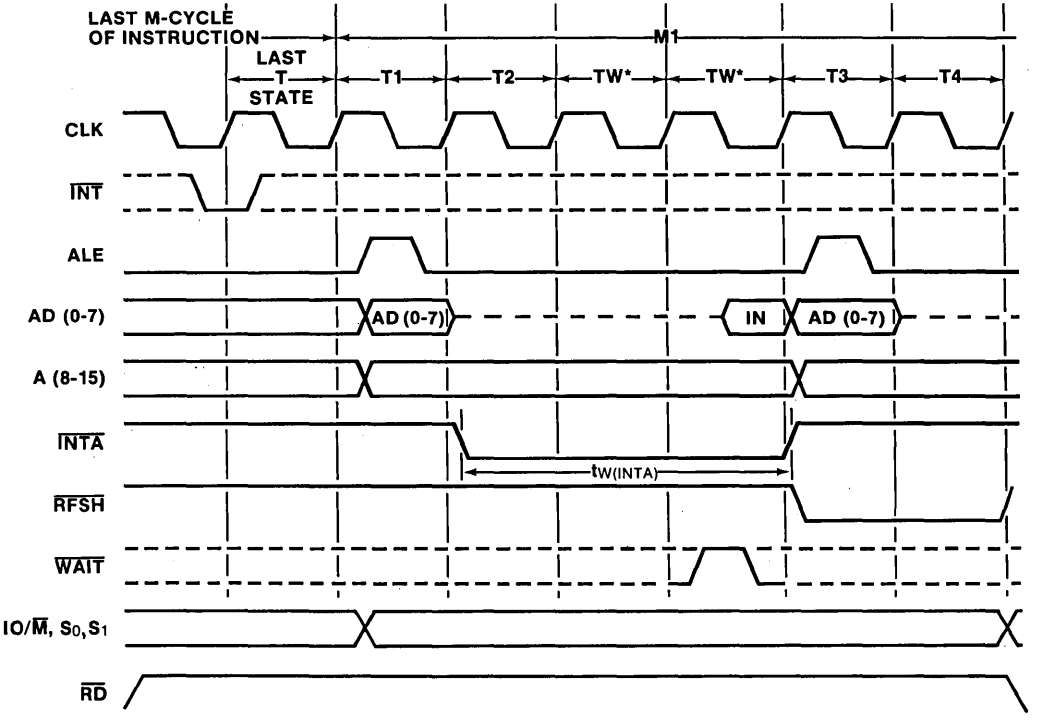


FIGURE 10. Interrupt Acknowledge Cycle

45-16

Tw is the CPU generated wait state in response to an interrupt request.

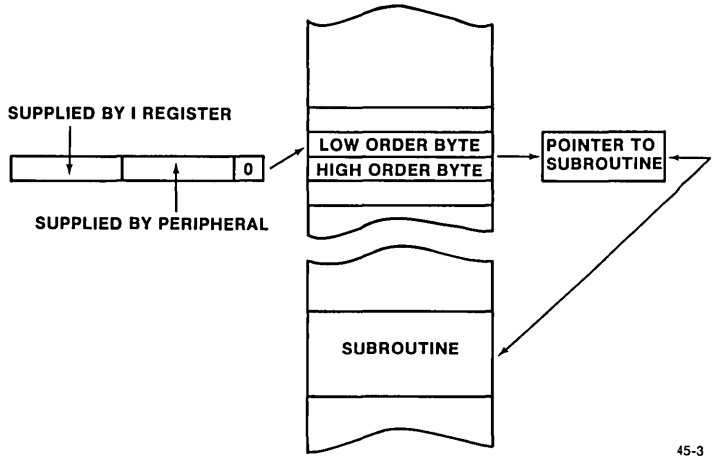


FIGURE 11. Interrupt Mode 2

ENABLING INTERRUPTS

1. $\overline{\text{NMI}}$, being a non-maskable interrupt request, is executed as it occurs and can never be disabled.

2. The maskable interrupt inputs ($\overline{\text{RSTA}}$, $\overline{\text{RSTB}}$, $\overline{\text{RSTC}}$, and $\overline{\text{INTR}}$) are enabled under program control through the use of the interrupt control register and enable/disable interrupt instruction.

The appropriate interrupt control bits in 4-bit interrupt control register (IEA, IEB, IEC, and IEI) must be enabled in conjunction with IFF1 and IFF2, before the maskable $\overline{\text{INTR}}$ and $\overline{\text{RST}}$ A, $\overline{\text{B}}$, $\overline{\text{C}}$ can be accepted by the CPU.

3	2	1	0
IEA	IEB	IEC	IEI

The interrupt control register is an on-chip write only output port located at port address X'BB. Its contents are:

BIT	NAME	FUNCTION
0	IEI	Interrupt Enable for $\overline{\text{INTR}}$
1	IEC	" " " $\overline{\text{RSTC}}$
2	IEB	" " " $\overline{\text{RSTB}}$
3	IEA	" " " $\overline{\text{RSTA}}$

For Example: In order to enable $\overline{\text{RSTB}}$, CPU interrupts must be enabled and IEB must be set.

At reset, IEI bit is set and other mask bits, IEA, IEB, IEC are cleared. This maintains the software compatibility between NSC800 and INS8080A (or Z80A).

POWER SAVE FEATURE

The NSC800 provides a unique power save mode by the means of the $\overline{\text{PS}}$ pin. $\overline{\text{PS}}$ input is sampled at the last t state of the last M cycle of an instruction. After recognizing an active (low) level on $\overline{\text{PS}}$, the NSC800 stops its internal clocks, thereby reducing its power dissipation, yet maintaining all register values and internal control status. A small amount of power is still consumed as the NSC800 keeps its oscillator running, and makes the CLK signal available to the system. When $\overline{\text{PS}}$ is returned high, the opcode fetch (or M1 cycle) of the CPU begins in a normal manner. Note this M1 cycle could also be an interrupt acknowledge cycle if the NSC800 was interrupted simultaneously with $\overline{\text{PS}}$. *Figure 12* illustrates the power save feature.

In the event $\overline{\text{BREQ}}$ is asserted (low) at the end of an instruction cycle and $\overline{\text{PS}}$ is active simultaneously, the following occurs:

1. The NSC800 will go into $\overline{\text{BACK}}$ cycle

2. Upon completion of $\overline{\text{BACK}}$ cycle and $\overline{\text{PS}}$ is still active then the CPU will go into power save mode.

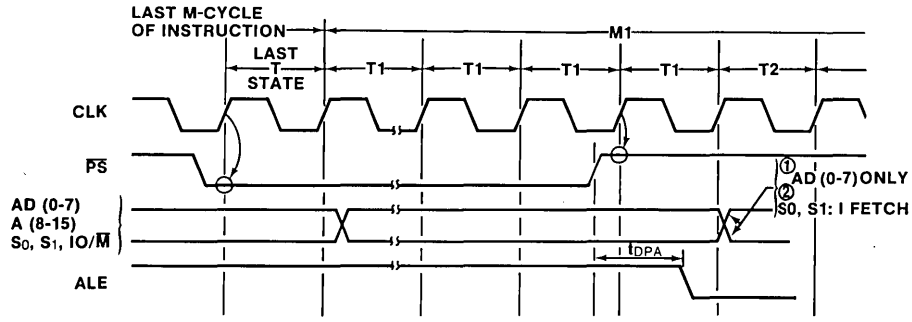


FIGURE 12. NSC800 Power Save

INSTRUCTION SET

In the following instruction set listing, the notation used is shown below.

b: Used in instructions employing bit mode addressing to designate one bit in a register or memory location.

cc: Designates condition codes used in conditional Jumps, Calls, and Return instructions; may be

NZ = Non Zero (Z Flag = 0)
Z = Zero (Z Flag = 1)
NC = Non Carry (C Flag = 0)
C = Carry (C Flag = 1)

PO = Parity Odd or No Overflow (P/V = 0)
PE = Parity Even or Overflow (P/V = 1)
P = Positive (S = O)
M = Negative (S = 1)

d: Used in instructions employing relative or indexed modes of addressing to designate 8-bit signed 2's complement displacement.

kk: Subset of cc condition codes used in conjunction with conditional relative jumps; may be NZ, Z, NC or C.

m_1: Used in instructions employing register indirect or indexed modes of addressing; may be (HL), (IX + d), or (IY + d).

m_2: Used in instructions employing register indirect or direct modes of addressing; may be (BC), (DE), or (nn).

n: Any 8-bit binary number

nn: Any 16-bit binary number

pp: Used in 16-bit arithmetic instructions employing register modes of addressing; may be BC, DE, SP, or register designated as destination operand.

qq: Used in instructions employing register modes of addressing; may be BC, DE, HL, AF, IX, or IY.

r: Used in instructions employing register mode of addressing; may be A, B, C, D, E, H, or L.

rr: Used in instructions employing register mode of addressing; may be BC, DE, HL, SP, IX, or IY.

ss: Used in instructions employing register mode of addressing; may be HL, IX, or IY.

T: Used in restart instructions employing modified page zero addressing mode; may take on hex values of 0, 8, 10, 18, 20, 28, 30, or 38.

X_L: Subscript L indicates the low order byte of a 16-bit register.

X_H: Subscript H indicates the high order byte of a 16-bit register.

( ): Parentheses indicate the contents are considered a pointer to a memory or I/O location.

8-BIT LOADS

- Register to Register -

Mnemonic	Description	Operation
LD r_d, r_s	Load register r_d with r_s	$r_d \leftarrow r_s$
LD A, I	Load ACC with register I	A ← I
LD I, A	Load register I with ACC	I ← A
LD A, r	Load ACC with register R	A ← r
LD r, A	Load register R with ACC	r ← A
LD r, n	Load register r with immediate data n	r ← n

- Register to Memory -

Mnemonic	Description	Operation
LD m_1, r	Load memory from register r	$m_1 \leftarrow$ r
LD m_2, A	Load memory from ACC	$m_2 \leftarrow$ A
LD m_1, n	Load memory with immediate data n	$m_1 \leftarrow$ n

- Memory to Register -

Mnemonic	Description	Operation
LD r, m_1	Load register r from memory	r ← m_1
LD A, m_2	Load ACC from memory	A ← m_2

16-BIT LOADS

- Register to Register -

Mnemonic	Description	Operation
LD rr, nn	Load register rr with immediate data nn	rr ← nn
LD SP, ss	Load SP register with register ss	SP ← ss

- Register to Memory -

Mnemonic	Description	Operation
LD (nn), rr	Load memory location nn with 16-bit register rr	(nn) ← rr_L (nn+1) ← rr_H
PUSH qq	Push contents of 16-bit register qq onto memory stack	(SP-1) ← qq_H (SP-2) ← qq_L SP ← SP-2

- Memory to Register -

Mnemonic	Description	Operation
LD rr, (nn)	Load 16-bit register rr from memory location nn	$rr_L \leftarrow (nn)$ $rr_H \leftarrow (nn+1)$
POP qq	Pop contents of stack to register qq	$qq_L \leftarrow (SP)$ $qq_H \leftarrow (SP+1)$ $SP \leftarrow SP+2$

8-BIT ARITHMETIC

- Register Addressed Arithmetic -

Mnemonic	Description	Operation
ADD A, r	Add contents of register r to ACC	$A \leftarrow A+r$
ADC A, r	Add with carry contents of register r to ACC	$A \leftarrow A+r+CY$
SUB r	Subtract contents of register r from ACC	$A \leftarrow A-r$
SBC A, r	Subtract with carry contents of register r from ACC	$A \leftarrow A-r-CY$
AND r	Logically AND contents of register r with ACC	$A \leftarrow A \wedge r$
OR r	Logically OR contents of register r with ACC	$A \leftarrow A \vee r$
XOR r	Exclusive OR contents of register r with ACC	$A \leftarrow A \veebar r$
CP r	Compare contents of register r to ACC	A:r Z flag ← 1 if A=r Otherwise Z Flag ← 0
INC r	Increment contents of register r	$r \leftarrow r+1$
DEC r	Decrement contents of register r	$r \leftarrow r-1$
DAA	Decimal adjust ACC	(ACC adjust for BCD)
CPL	Complement ACC (1's complement)	$A \leftarrow \overline{A}$
NEG	Negate ACC (2's complement)	$A \leftarrow 0-A$
CCF	Complement carry flag	$CY \leftarrow \overline{CY}$
SCF	Set carry flag	$CY \leftarrow 1$

- Immediate Addressing Mode Arithmetic -

Mnemonic	Description	Operation
ADD A, n	Add number n to ACC	$A \leftarrow A+n$
ADC A, n	Add with carry number n to ACC	$A \leftarrow A+n+CY$
SUB n	Subtract number n from ACC	$A \leftarrow A-n$
SBC A, n	Subtract with carry number n from ACC	$A \leftarrow A-n-CY$
AND n	AND number n with ACC	$A \leftarrow A \wedge n$
OR n	OR number n with ACC	$A \leftarrow A \vee n$
XOR n	Exclusive OR number n with ACC	$A \leftarrow A \veebar n$
CP n	Compare number n to ACC	A:n Z flag ← 1 if A=n Otherwise Z Flag ← 0

- Memory Addressed Arithmetic -

Mnemonic	Description	Operation
ADD A,m_1	Add memory to ACC	$A \leftarrow A+m_1$
ADC A,m_1	Add with carry memory to ACC	$A \leftarrow A+m_1+CY$
SUB m_1	Subtract memory from the ACC	$A \leftarrow A-m_1$
SBC A, m_1	Subtract with carry memory from ACC	$A \leftarrow A-m_1-CY$
AND m_1	AND memory with ACC	$A \leftarrow A \wedge m_1$
OR m_1	OR memory with ACC	$A \leftarrow A \vee m_1$
XOR m_1	Exclusive OR memory with ACC	$A \leftarrow A \veebar m_1$
CP m_1	Compare memory with ACC	A:m_1 Z flag ← 1 if A=r Otherwise Z Flag ← 0 INC
m_1	Increment memory	$m_1 \leftarrow m_1+1$
DEC m_1	Decrement memory	$m_1 \leftarrow m_1-1$

16-BIT ARITHMETIC

- Register Addressed Arithmetic -

Mnemonic	Description	Operation
ADD ss pp	Add 16-bit register pp to 16-bit register ss	ss ← ss+pp
ADC HL, pp	Add with carry 16-bit register pp to HL 1	HL ← HL +pp+CY
SBC HL, pp	Subtract with carry 16-bit register pp from HL	HL ← HL -pp-CY
INC rr	Increment 16-bit register rr	rr ← rr+1
DEC rr	Decrement 16-bit register rr	rr ← rr-1

BIT SET, RESET, and TEST

- Register -

Mnemonic	Description	Operation
SET b, r	Set bit b in register r	r_b ← 1
RES b, r	Reset bit b in register r	r_b ← 0
BIT b, r	Test bit b in register r	Z ← r_b

- Memory -

Mnemonic	Description	Operation
SET b, m_1	Set bit b in memory location m_1	m_{1b} ← 1b
RES b, m_1	Reset bit b in memory location m_1	m_{1b} ← 0
BIT b, m_1	Test bit b in memory location m_1	Z ← m_{1b}

EXCHANGES

- Register/Register

Mnemonic	Description	Operation
EX DE, HL	Exchange contents of DE and HL register	DE ↔ HL
EX AF, AF1	Exchange contents of A and F registers with A1 and F1 registers	AF ↔ AF'
EXX	Exchange contents of BC, DE and HL registers with corresponding alternate registers	BC ↔ BC' DE ↔ DE' HL ↔ HL'

- Register/Memory -

Mnemonic	Description	Operation
EX (SP), ss	Exchange top of stack with 16-bit register ss	(SP) ↔ ss_L (SP+1) ↔ ss_H

MEMORY BLOCK MOVES and SEARCHES

- Single Operations -

Mnemonic	Description	Operation
LDI	Move data from memory location (HL) to memory location (DE), increment memory pointers, and decrement byte counter BC.	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1
LDD	Move data from memory location (HL) to memory location (DE), and decrement memory pointer, and byte counter BC.	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← HL-1
CPI	Compare data in memory location (HL) to ACC, increment memory pointer and decrement byte counter BC.	A– (HL) HL ← HL+1 BC ← BC-1
CPD	Compare data in memory location (HL) to ACC and decrement memory pointer and byte counter BC.	A– (HL) HL ← HL-1 BC ← BC-1

- Repeat Operations -

Mnemonic	Description	Operation
LDIR	Move data from memory location (HL) to memory location (DE), increment memory pointers, decrement byte counter BC, repeat until BC=0	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC=0
LDDR	Move data from memory location (HL) to memory location (DE), decrement memory pointers and byte counter BC, repeat until BC=0	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC=0
CPIR	Compare data in memory location (HL) to ACC, increment memory pointer, decrement byte counter BC, repeat until BC=0 or (HL) = A	A-(HL) HL ← HL+1 BC ← BC-1 Repeat until BC=0 or (HL)=A

- Repeat Operations -

Mnemonic	Description	Operation
CPDR	Compare data in memory location (HL) to ACC, decrement memory pointer and byte counter BC, repeat until BC=0 or (HL)=A	A-(HL) HL ← HL-1 BC ← BC-1 Repeat until BC=0 or (HL)=A

INPUT/OUTPUT

Mnemonic	Description	Operation
IN A, (n)	Input from I/O device at address n to ACC	A ← (n)
OUT (n), A	Output to I/O device at address n from ACC	(n) ← A
IN r, (C)	Input from I/O device at address (C) to register	r ← (C)
OUT (C), r	Output to I/O device at address (C) from register	(C) ← r
INI	Input from I/O device at address (C) to memory location (HL), increment pointer, and decrement B counter	(HL) ← (C) HL ← HL+1 B ← B-1
OUTI	Output to I/O at address (C) from memory location (HL), increment pointer, and decrement B counter	(C) ← (HL) HL ← HL+1 B ← B-1
IND	Input from I/O device at address (C) to memory location (HL) and decrement pointer, and B counter	(HL) ← (C) HL ← HL-1 B ← B-1
OUTD	Output to I/O device at address (C) from memory location (HL) and decrement pointer and B counter	(C) ← (HL) HL ← HL-1 B ← B-1
INIR	Input from I/O device at address (C) to memory location (HL), increment pointer, decrement B counter, and repeat until B=0	(HL) ← C HL ← HL+1 B ← B-1 Repeat until B=0
OUTIR	Output to I/O device at address (C) from memory location (HL), increment pointer, decrement B counter, and repeat until B=0	(C) ← (HL) HL ← HL+1 B ← B-1 Repeat until B=0
INDR	Input from I/O device at address (C) to memory location (HL), decrement pointer and B counter, and repeat until B=0	(HL) ← (C) HL ← HL-1 B ← B-1 Repeat until B=0
OUTDR	Output to I/O device at address (C) from memory location (HL), decrement pointer and B counter, and repeat until B=0	(C) ← (HL) HL ← HL-1 B ← B-1 Repeat until B=0

CPU CONTROL

Mnemonic	Description	
NOP	No operation	
HALT	Halt processor	
DI	Disable Interrupts	
EI	Enable Interrupts	
IM 0	Set Interrupt Mode 0	
IM 1	Set Interrupt Mode 1	Sets $\overline{\text{INTR}}$ mode 0, 1, 2
IM 2	Set Interrupt Mode 2	

PROGRAM CONTROL

- Jumps -

Mnemonic	Description	Operation
JP nn	Unconditional jump direct to nn	PC ← nn
JP (ss)	Unconditional jump indirect via ss register	PC ← ss
JP cc, nn	Conditionally jump direct to nn if cc is true	If cc true PC ← nn, else continue
JR d	Unconditional jump to PC + d	PC ← PC+d
JR kk, d	Conditionally jump PC + d if kk is true	If kk true, PC ← PC+d,
DJNZ,d	Decrement B register and jump to PC + d if B ≠ 0, otherwise continue	B ← B-1 If B=0 PC ← PC+d

PROGRAM CONTROL

- Calls -

Mnemonic	Description	Operation
CALL nn	Unconditional call to subroutine at location nn	(SP-1) ← PC_H (SP-2) ← PC_L PC ← *nn*
CALL cc, nn	Conditional call to subroutine at location nn if cc true	If cc true: (SP-1) ← PC_H (SP-2) ← PC_L PC ← nn, else continue

- Returns -

Mnemonic	Description	Operation
RET	Unconditional return from subroutine	PC_L ← (SP) PC_H ← (SP+1)
RET cc	Conditional return from subroutine	If cc true: PC_L ← (SP) PC_H ← (SP+1) else continue
RETI	Return from interrupt	PC_L ← (SP) PC_H ← (SP+1)

- Returns -

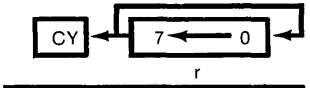
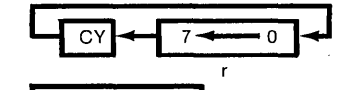
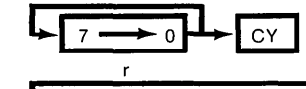
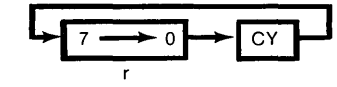
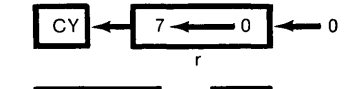
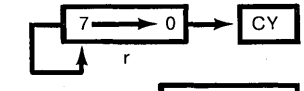
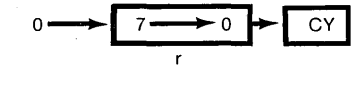
Mnemonic	Description	Operation
RETN	Return from non-maskable interrupt	PC_L ← (SP) PC_H ← (SP+1) Restore interrupt enable status

- Restarts -

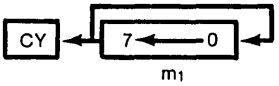
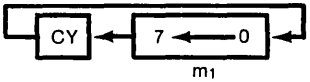
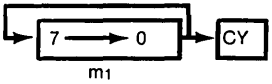
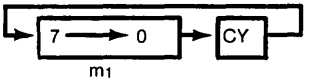
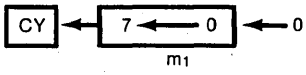
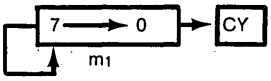
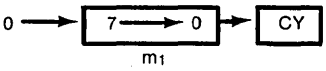
Mnemonic	Description	Operation
RST T	Interrupt to location T	(SP-1) ← PC_H (SP-2)← PC_L PC ← T

ROTATE and SHIFT

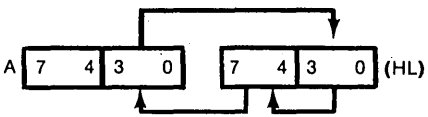
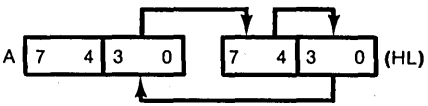
- Register -

Mnemonic	Description	Operation
RLC r	Rotate register r left circular	
RL r	Rotate register r left through carry	
RRC r	Rotate register r right circular	
RR r	Rotate register r right through carry	
SLA r	Shift register r left arithmetic	
SRA r	Shift register r right arithmetic	
SRL r	Shift register r right logical	

- Memory -

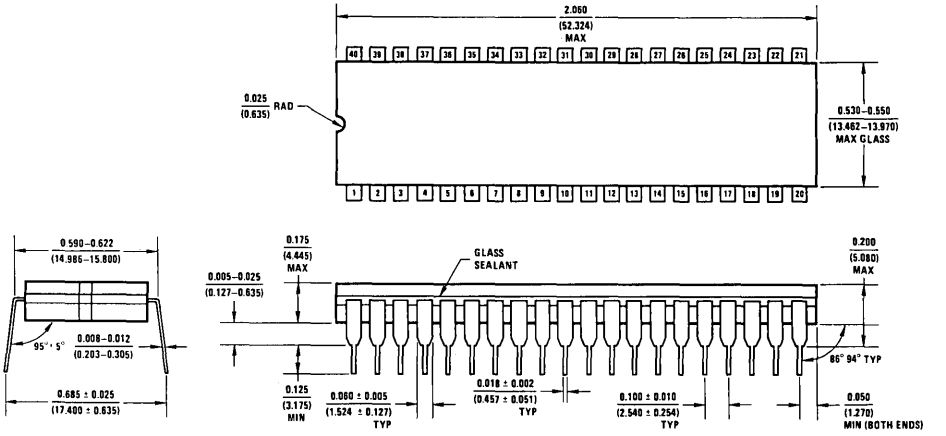
Mnemonic	Description	Operation
RLC m_1	Rotate memory left circular	
RL m_1	Rotate memory left through carry	
RRC m_1	Rotate memory right circular	
RR m_1	Rotate memory right through circular	
SLA m_1	Shift memory left arithmetic	
SRA m_1	Shift memory right arithmetic	
SRL m1	Shift memory right logical	

- Register/Memory -

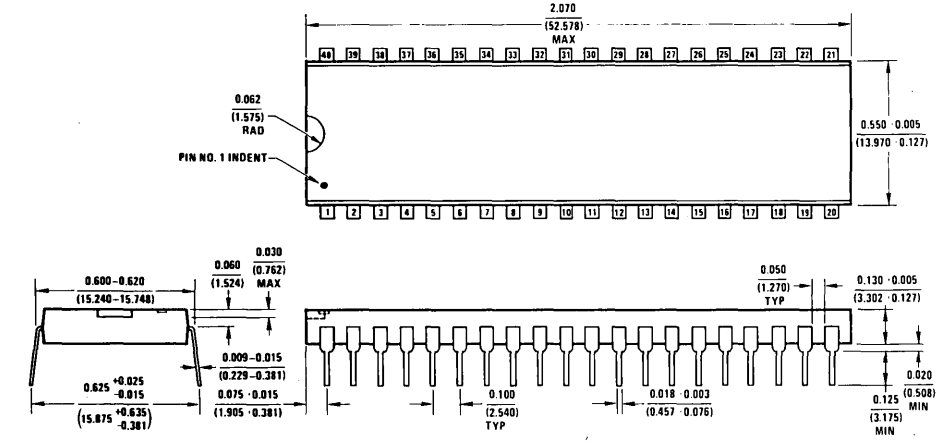
Mnemonic	Description	Operation
RLD	Rotate digit left and right between ACC and memory (HL)	
RRD	Rotate digit right and left between ACC and memory (HL)	

Physical Dimensions

inches (millimeters)



Ceramic Dual-In-Line Package (J)



Plastic Dual-In-Line Package (N)

Ordering Information

NSC800XX

- **I = Industrial Temperature (-40° C to +85° C)**
- **M = Military Temperature (Availability to be announced)**

- **N = Plastic Package**
- **J = Cerdip Package**

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel.: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/II
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel.: (03)355-3711
TWX: 232-2015 NSCJJ

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex
1121008 CABINE SAO PAULO

NS Electronics Pty. Ltd.
Cnr. Stud Rd & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

NSC810 RAM-I/O-Timer

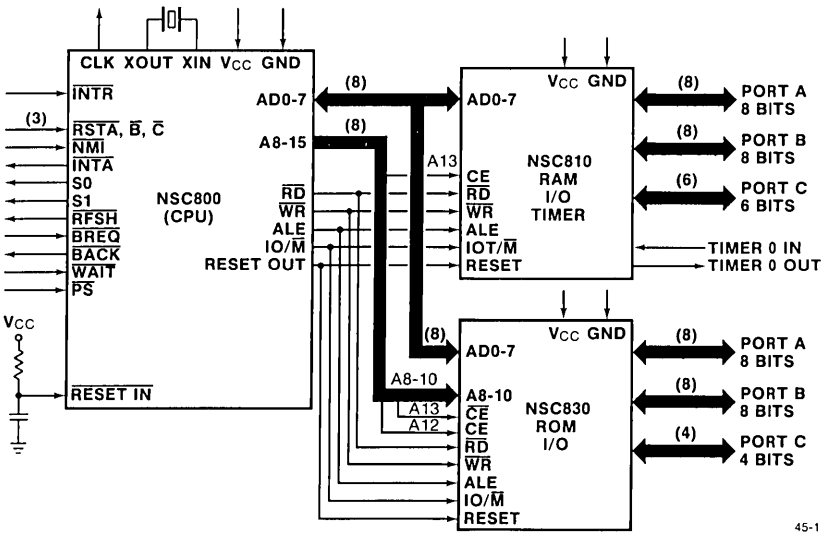
General Description

The NSC810 is a RAM-I/O-Timer device contained in a standard 40-pin, dual-in-line package. The chip, which is fabricated using P²CMOS silicon gate technology, functions as a memory, an input/output peripheral interface and a timing device. The memory is comprised of 1024 bits of static RAM organized as 128 x 8. The I/O portion consists of 22 programmable input/output bits arranged as three separate ports, with each bit individually definable as an input or output. The port bits can be set or cleared individually and can be written or read in bytes. Several types of strobed mode operations are available through Port A. The timer portion of the device consists of two programmable 16-bit binary down-counters each capable of operation in any one of six modes. Timer counts are extendable by one of the available prescale values.

Features

- 128 x 8 Random Access Memory
- Three Programmable I/O Ports
- Two 16-Bit Programmable Counter/Timers
- Single 5V Power Supply
- Very Low Power Consumption
- Fully Static Operation
- Single-Instruction I/O Bit Operations
- Timer Operation — DC to 4MHz
- Directly Compatible with NSC800 Family

NSC800 Microcomputer Family Block Diagram



Absolute Maximum Ratings

Storage Temperature Range.....-65°C to 150°C
Voltage at Any Pin with Respect to Ground ....-0.3V to V_{CC} +0.3V
Lead Temperature (Soldering, 10 seconds).....300°C
Power Dissipation 1W

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40°C to +85°C	5V
Commercial	0°C to +70°C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating voltage range components to be announced.

DC Electrical Characteristics

T_A = 0°C to + 70°C, V_{CC} = +5V ± 10%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}	Logical 1 Input Voltage	0.7 V_{CC}		V_{CC}	V	
V_{IL}	Logical 0 Input Voltage	0		0.3 V_{CC}	V	
V_{OH}	Logical 1 Output Voltage	2.4			V	I_{OH} = -1.0 mA
		V_{CC}-0.5			V	I_{OUT} = -10 μA
V_{OL}	Logical 0 Output Voltage	0		0.4	V	I_{OL} = 2 mA
		0		0.1	V	I_{OUT} = 10 μA
I_{IL}	Input Leakage Current	-1.0		1.0	μA	0 ≤ V_{IN} ≤ V_{CC}
I_{OL}	Output Leakage Current	-1.0		1.0	μA	0 ≤ V_{IN} ≤ V_{CC}
I_{CC}	Active Supply Current		5		mA	I_{OUT} = 0; $f_{(XIN)}$ = 5MHz

AC Electrical Characteristics

T_A = 0° C to + 70° C, V_{CC} = +5V ± 10%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
t_{ACC}	Access Time from ALE			250	ns	CL = 100 pF
t_{AS}	Address 0-7, CE, IOT/$\overline{M}$ Set-Up Time	50			ns	
t_{AH}	Address 0-7, CE, IOT/$\overline{M}$ Hold Time	50			ns	
t_{RDD}	Data Bus Disable	0		75	ns	CL = 100 pF
t_{ARW}	ALE to Read or Write Strobe	75			ns	
t_{RD}	Read Strobe Width	200			ns	
t_{WR}	Write Strobe Width	200			ns	
t_{ALE}	ALE Strobe Width (high)	100			ns	
t_{DS}	Data Set-up Time	50			ns	
t_{DH}	Data Hold Time	75			ns	
t_{RWA}	Read or Write to Next ALE Time	100			ns	
t_{PS}	Peripheral Data Set-up Time	100			ns	
t_{PH}	Peripheral Data Hold Time	100			ns	
t_{DO}	Port Data Output Valid			200	ns	Port Loading = 100 pF
t_{SB}	$\overline{STB}$ to Buffer Full Valid			250	ns	
t_{SW}	Minimum $\overline{STB}$ Width	200			ns	
t_{SS}	Peripheral Data Set-up Time	100			ns	
t_{SH}	Peripheral Data Hold with Respect to $\overline{STB}$	100			ns	
t_{SI}	$\overline{STB}$ to $\overline{INTR}$ Output			250	ns	
t_{RB}	$\overline{RD}$ to BF Output			250	ns	
t_{RI}	$\overline{RD}$ to $\overline{INTR}$ Output			250	ns	
t_{WB}	$\overline{WR}$ to BF Output			250	ns	
t_{WI}	$\overline{WR}$ to $\overline{INTR}$ Output			250	ns	
t_{PE}	Peripheral Bus Enable			200	ns	
t_{PZ}	Peripheral Bus Disable (TRI-STATE®)			150	ns	

Timer AC Electrical Characteristics (See Figure 5.)

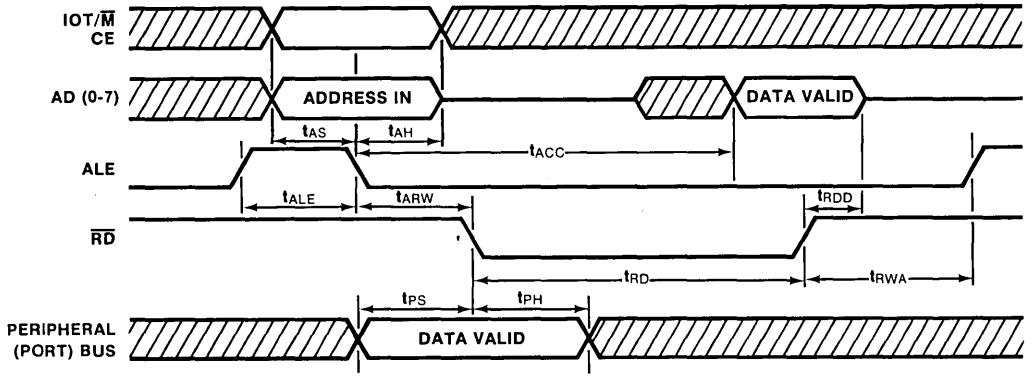
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
F_C	Clock Frequency	dc		2.0	MHz	
F_{CP}	Clock Frequency	dc		4.0	MHz	Prescale selected
t_{CW}	Clock Pulse Width	200			ns	
t_{CWP}	Clock Pulse Width	100			ns	Prescale selected
t_{CW}	Gate Width	150			ns	High or low time
t_{GS}	Gate Set-up Time	100			ns	With respect to negative going clock edge
t_{GH}	Gate Hold Time	50			ns	With respect to negative going clock edge
t_{CO}	Clock to Output Delay			250	ns	CL = 100 pF

Capacitance

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance		4		pF	
C_{OUT}	Output Capacitance		6		pF	

General Timing Waveforms

Read Cycle

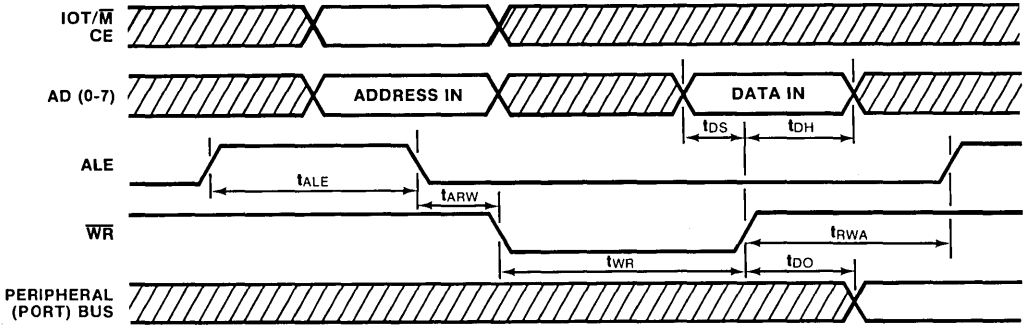


NOTE: Diagonal lines indicate interval of invalid data.

20-1

(Read from RAM, Port or Timer)

Write Cycle

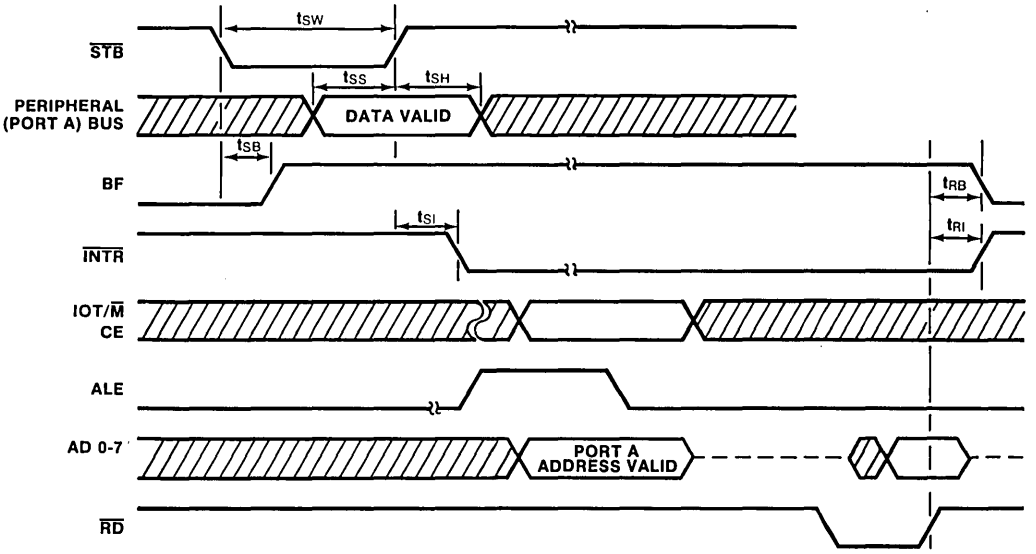


NOTE: Diagonal lines indicate interval of invalid data.

(Write to RAM, Port or Timer)

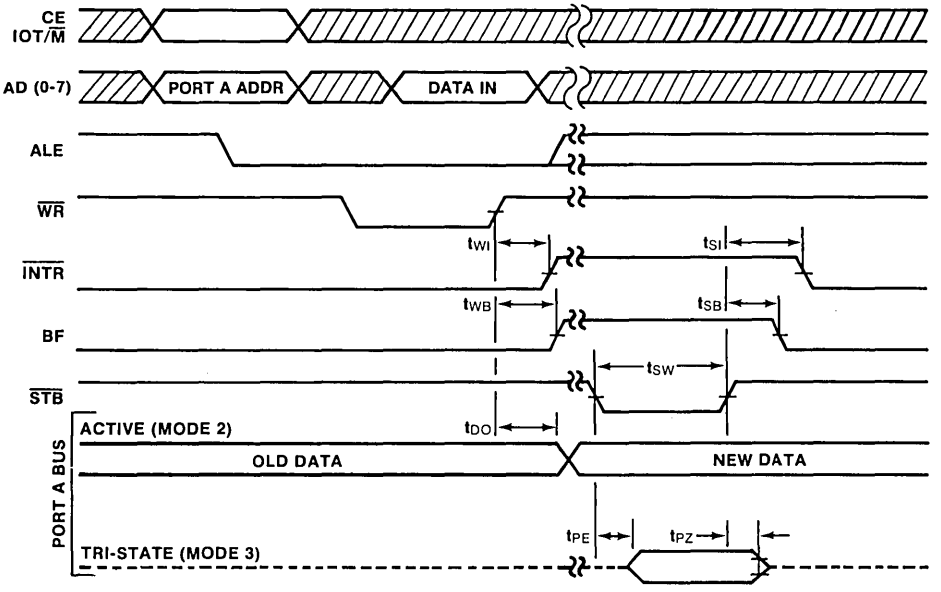
20-2

Handshake Timing



NOTE: Diagonal lines indicate interval of invalid data.

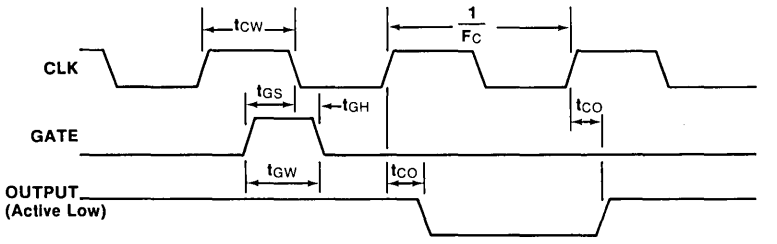
Strobed Mode Input 20-3



NOTE: Diagonal lines indicate interval of invalid data.

Strobed Mode Output 20-4

Timer Waveforms



20-5

NSC810 Functional Pin Descriptions

The function and mnemonic for the NSC810 input/output signals are described below:

INPUT SIGNALS

Reset (RESET): RESET is an active-high input that resets all registers to 0 (low). The RAM contents remain unaltered.

Input/Output Timer or RAM Select (IOT/$\overline{M}$): IOT/$\overline{M}$ is an I/O memory select input line. A logic 1 (high) input selects the I/O-Timer portion of the chip; a logic 0 (low) input selects the RAM portion of the chip. IOT/$\overline{M}$ is latched at the falling edge of ALE.

Chip Enable (CE): CE is an active-high input that allows access to the NSC810. CE is latched at the falling edge of ALE.

Read ($\overline{RD}$): The $\overline{RD}$ is an active-low input that enables a read operation of the RAM or I/O-Timer location.

Write ($\overline{WR}$): The $\overline{WR}$ is an active-low input that enables a write operation to RAM or I/O-Timer locations.

Address Latch Enable (ALE): The falling edge of the ALE input latches AD0-AD7, CE and IOT/$\overline{M}$ inputs to form the address for RAM, I/O or timer.

Timer 0 Input (T0IN): T0IN is the clock input for timer 0.

OUTPUT SIGNALS

Timer 0 Output (T0OUT): T0OUT is the programmable output of timer 0.

POWER SUPPLY PINS

Positive DC Voltage (V_{CC}): V_{CC} is the 5-volt supply pin.

Ground (V_{SS}): V_{SS} is the ground reference pin.

INPUT/OUTPUT SIGNALS

Address/Data Bus (AD0-AD7): The multiplexed bidirectional address/data bus, AD0-AD7 pins, are in the high impedance state when the NSC810 is not selected. AD0-AD7 will latch address inputs at the falling edge of ALE. The address will designate a location in RAM, I/O or timer. $\overline{WR}$ input enables 8-bit data to be written into the addressed location. $\overline{RD}$ input enables 8-bit data to be read from the addressed location. The $\overline{RD}$ or $\overline{WR}$ inputs occur while ALE is low .

Port A, 0-7 (PA0-PA7): Port A is an 8-bit basic mode input/output port, also capable of strobed mode I/O utilizing three control signals from Port C. Port A strobed mode outputs can be active or any of the three states of the TRI-STATE output.

Port B, 0-7 (PB0-PB7): Port B is an 8-bit basic mode input/output port (see *Figure 3* Port functions).

Port C, 0-5 (PC0-PC5): Port C is a 6-bit basic mode I/O port. Each pin has a programmable second function, as follows:

PC0/$\overline{INTR}$: $\overline{INTR}$ is an active-low strobed mode interrupt request to the Central Processor Unit (CPU).

PC1/BF: BF is an active-high buffer full output to peripheral devices.

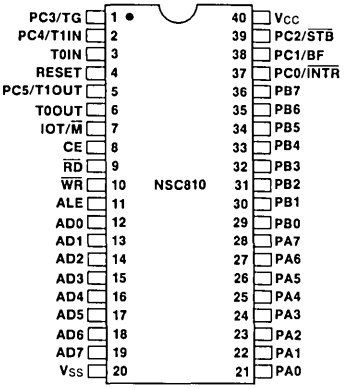
PC2/$\overline{STB}$: $\overline{STB}$ is an active-low strobe input from peripheral devices.

PC3/TG: TG is the timer gating signal.

PC4/T1 IN: T1IN is the clock input for timer 1.

PC5/T1 OUT: T1OUT is the programmable output of timer 1.

Pin Configuration



20-6

NSC810 Functional Description

Refer to *Figure 1* for a detailed block diagram of the NSC810.

Random Access Memory (RAM): The memory portion of the RAM-I/O-Timer is accessed by a 7-bit address input to pins AD0 through AD6. The IOT/$\overline{M}$ input must be low (RAM select) and the CE input must be high at the falling edge of ALE to address the RAM. Address bit AD7 is a "don't care" for RAM addressing. Timing for RAM read and write operations is shown in the RAM timing diagrams.

Input/Output (I/O): The I/O portion of the NSC810 contains three sets of I/O called Ports. There are two ports (A and B) which contain eight bits each and one port (Port C) which has six bits. Any bit or combination of bits in a port may be addressed with Set or Clear commands. A port can also be addressed as an 8-bit word (6 bits for Port C). All ports share common function of Read, Write, Bit-Set and Bit-Clear. Additionally, Port A is programmable for strobed (handshake) mode input or output. Port C has programmable second functions for each bit associated with strobed modes and timer functions. Table 1 defines the address location of the ports, timers and control registers.

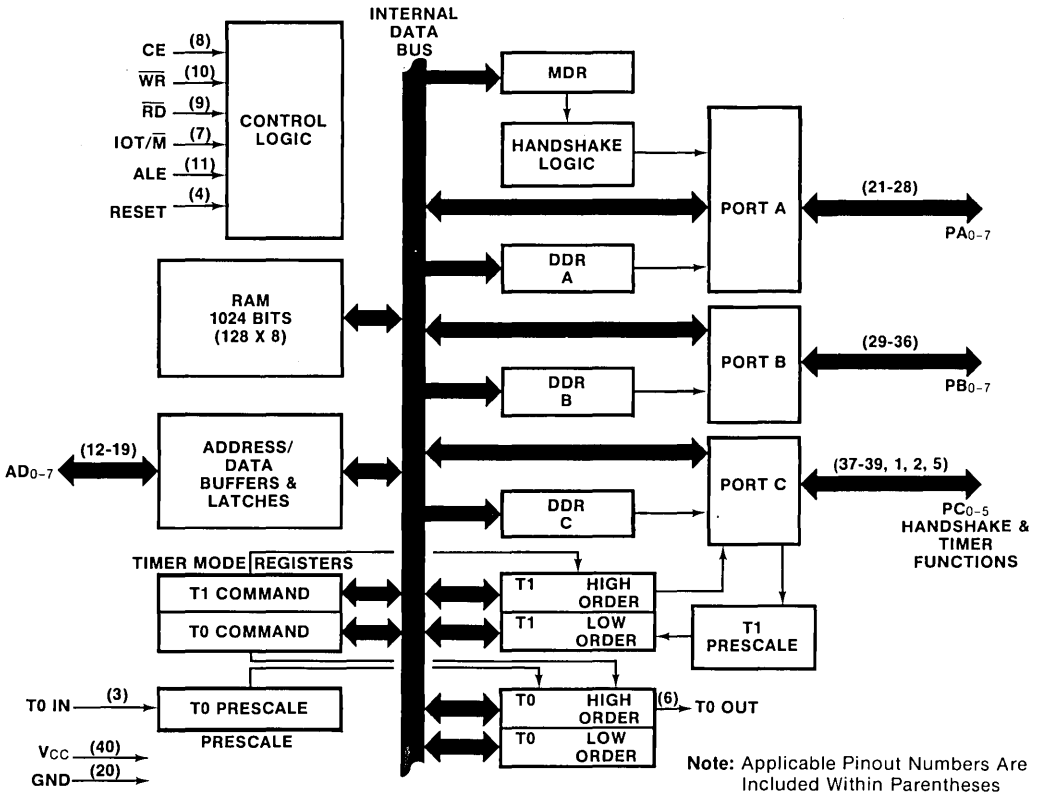


FIGURE 1. NSC810 Detailed Block Diagram

20-7

MODE DEFINITION REGISTER (MDR)

The Mode Definition Register (MDR) defines the operating mode for Port A. While Ports B and C are always in the basic I/O mode, there are four operating modes for Port A:

Mode 0 - Basic I/O (Input or Output)
Mode 1 - Strobed Mode Input
Mode 2 - Strobed Mode Output — Active Peripheral Bus
Mode 3 - Strobed Mode Output — TRI-STATE Peripheral Bus

The MDR has the I/O address assignment xxx00111 and is illustrated for the four modes:

Mode \ Bit	7	6	5	4	3	2	1	0
0	x	x	x	x	x	x	x	0
1	x	x	x	x	x	x	0	1
2	x	x	x	x	x	0	1	1
3	x	x	x	x	x	1	1	1

x = don't care

DATA DIRECTION REGISTERS (DDR)

Each port bit has a data direction register (DDR) that defines the I/O state of the bit. The bit is configured as an input if a "0" is written into its DDR, or as an output if a "1" is written. The DDR bits cannot be individually written to; the entire DDR byte is affected by a write to the DDR address. Thus, all data must be consistent with the direction desired for each port bit.

Table 1.

I/O and Timer Address Designations

8-Bit Address Field Bits 7	6	5	4	3	2	1	0	Designation I/O Port, Timer, etc.	R (Read) W (Write)
x	x	x	0	0	0	0	0	Port A (byte)	R/W
x	x	x	0	0	0	0	1	Port B (byte)	R/W
x	x	x	0	0	0	1	0	Port C (byte)	R/W
x	x	x	0	0	0	1	1	Not Used	—
x	x	x	0	0	1	0	0	DDR - Port A	W
x	x	x	0	0	1	0	1	DDR - Port B	W
x	x	x	0	0	1	1	0	DDR - Port C	W
x	x	x	0	0	1	1	1	Mode Definition Reg.	W
x	x	x	0	1	0	0	0	Port A - Bit Clear	W
x	x	x	0	1	0	0	1	Port B - Bit Clear	W
x	x	x	0	1	0	1	0	Port C - Bit Clear	W
x	x	x	0	1	0	1	1	Not Used	—
x	x	x	0	1	1	0	0	Port A - Bit Set	W
x	x	x	0	1	1	0	1	Port B - Bit Set	W
x	x	x	0	1	1	1	0	Port C - Bit Set	W
x	x	x	0	1	1	1	1	Not Used	—
x	x	x	1	0	0	0	0	Timer 0 (LB)	*
x	x	x	1	0	0	0	1	Timer 0 (HB)	*
x	x	x	1	0	0	1	0	Timer 1 (LB)	*
x	x	x	1	0	0	1	1	Timer 1 (HB)	*
x	x	x	1	0	1	0	0	STOP Timer 0	W
x	x	x	1	0	1	0	1	START Timer 0	W
x	x	x	1	0	1	1	0	STOP Timer 1	W
x	x	x	1	0	1	1	1	START Timer 1	W
x	x	x	1	1	0	0	0	Timer 0 Mode	R/W
x	x	x	1	1	0	0	1	Timer 1 Mode	R/W

x = don't care
LB = Low Order Byte
HB = High Order Byte

* A write accesses the Modulus Register.
A read accesses the Timer.

Any write or read operations on a port contradicting the DDR will not affect the port output or input. However, a read of a port bit, defined as an output, will cause a read from the output latch, and a write to a port bit, defined as an input, will modify the output latch. Refer to *Figure 2*.

PORT FUNCTIONS - BASIC I/O

Basic I/O is the mode of operation of Ports B and C and mode 0 of Port A (defined by the MDR). Read and write byte operations, and bit operations can be executed in the basic I/O mode. The timing for basic input and basic output modes is shown in the AC Characteristics tables.

When a read occurs, the information is latched from the peripheral (Port) bus during the leading (falling) edge of the $\overline{RD}$ strobe. When a write occurs, the port bus is modified after the trailing (rising) edge of the $\overline{WR}$ strobe with data from the AD bus. Port output data remains valid at the output pin from one trailing edge of $\overline{WR}$ strobe to the trailing edge of the next $\overline{WR}$ strobe which then modifies that port.

BIT OPERATIONS

The I/O features of the RAM-I/O-Timer allow modification of a single bit or several bits of a port with Bit-Set and Bit-Clear (see *Figure 2*). The address is set up to indicate that a bit set (or clear) is taking place. The incoming data on the address/data bus is latched at the trailing edge of the $\overline{WR}$ strobe and is treated as a mask. All bits containing "1's" will cause the indicated operation to be performed on the corresponding port bit. All bits of the data mask with "0's" cause the corresponding port bits to remain unchanged. Three sample operations are given, using Port B as an example:

Operation	Set B7	Clear B2 & B0	Set B4, B3 & B1
Address	xxx01101	xxx01001	xxx01101
Data	10000000	00000101	00011010
Port Pins			
Prior State	00001111	10001111	10001010
Next State	10001111	10001010	10011010

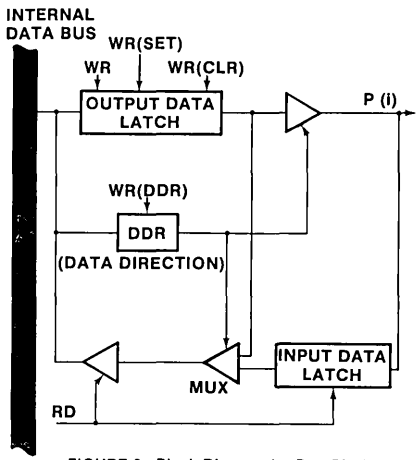


FIGURE 2. Block Diagram for Port Bit (i) 20-8

PORT A - STROBED (HANDSHAKE) MODE

Port A can be programmed (via the MDR) into one of 3 types of strobed mode for handshake communication with intelligent peripherals. When Port A is in mode 1, 2, or 3 (see description of MDR), Port C pins 0, 1, and 2 are used as handshake signals between the peripheral and the CPU. These handshake signals are designated $\overline{STB}$, BF, and $\overline{INTR}$. Bit Set and Clear operations are not allowed on these bits of Port C when Port A is in strobed mode. Timing parameters and timing diagrams are detailed under AC Characteristics.

$\overline{INTR}$ (Strobe Mode Interrupt) is an active-low interrupt from the I/O to the CPU. In strobed Input Mode, the CPU reads the valid data at Port A to clear the interrupt. In strobed Output Mode, the CPU clears the interrupt by writing to Port A.

The interrupt is enabled or disabled by setting or clearing Bit 2 of Port C.

$\overline{STB}$ (Strobe) is an active-low input from the peripheral device, signaling that data-transfer is about to begin. This strobe is interpreted as an "output request" if Port A is in a strobed output mode, or as a "data-valid" signal if Port A is in strobed input mode.

BF (Buffer Full) is an output from the I/O to the peripheral signaling that data transfer is complete. In strobed input mode, this strobe indicates that data is received into Port A and that no further data should be transmitted by the peripheral device until the port has been read (emptied). In strobed output mode BF indicates that the request from the peripheral has been processed by the CPU and the valid data now appears in Port A.

The bits of Port C that are used for handshake control of Port A (bits C0, C1, & C2) must be direction-defined appropriately in the DDR. Also, the DDR of Port A must be consistent with the mode specified in the MDR. Register set-up configurations for the three handshake modes are illustrated in Table 2.

Table 2.

Mode Definition Register Configurations

Mode	MDR	DDR Port A	DDR Port C	Port C Output Latch
Strobed Input	xxxxxx01	00000000	xxx011	xxx1xx
Strobed Output (Active)	xxxxx011	11111111	xxx011	xxx1xx
Strobed Output (TRI-STATE)	xxxxx111	11111111	xxx011	xxx1xx

TIMERS

The two timers in the RAM-I/O-Timer are 16-bit binary down counters, each timer having six modes of operation. Full count is reached at "n+1", where "n" is the value loaded into the modulus register. Read and write commands can occur at any time, asynchronous to timer operation by addressing the timer read buffer or modulus register, respectively. Each timer has a mode register and a write-only start/stop register. Each timer also has a prescaler which divides the incoming clock signal by a programmable value, extending the effective ranges of the timers while maintaining 16-bit precision. Selected prescale values are ÷1 or ÷2 for Timer 1, and ÷1, ÷2, or ÷64 for Timer 0. A diagram representing one timer and associated registers is shown in *Figure 3.*

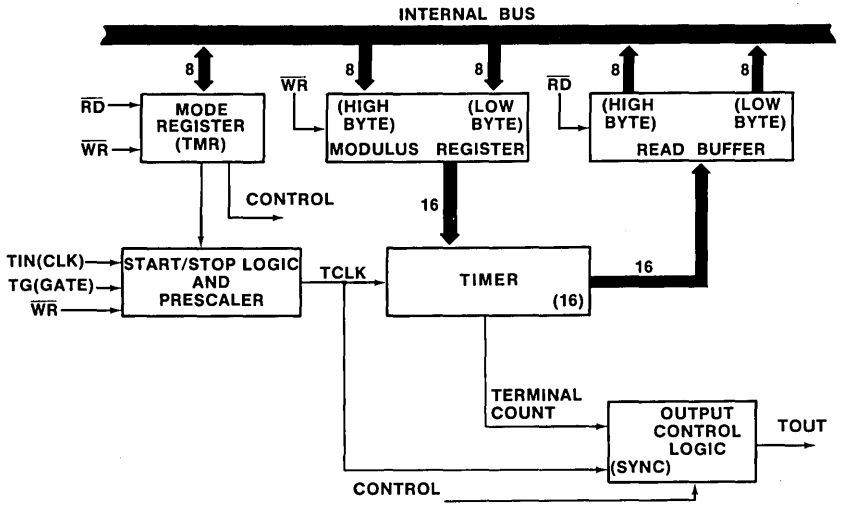
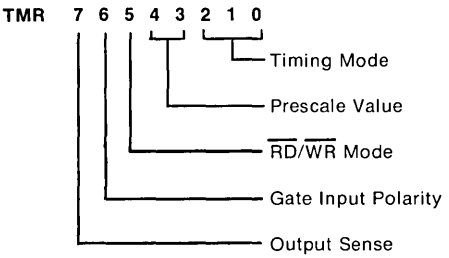


FIGURE 3. Timer Internal Block Diagram (One of Two)

20-9

SELECTING TIMER MODE

The Timer Mode Register (TMR) may be written or read at any time; however, to assure accurate timing it is important to modify the mode when the timer is stopped. The timer mode is selected from one of six modes with TMR bits 0, 1, and 2. Bits 3 and 4 select the prescale value if the prescaler is to be used. Bits 5, 6 and 7 select the read/write mode, gate input polarity, and output sense (active high or low). The bit functions of the TMR are further illustrated in *Figure 4.* Details of each timer mode are illustrated in *Figure 5.*



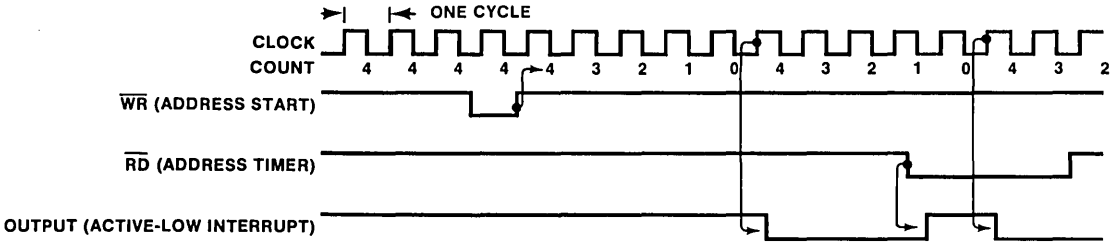
BIT 2 1 0		Timer Function
0 0 1	—	Event Counter
0 1 0	—	Event Timer (Stop watch)
0 1 1	—	Event Timer (Resetting)
1 0 0	—	One Shot
1 0 1	—	Square Wave
1 1 0	—	Pulse Generator

NOTE: *000 and 111 are no op's (timer halted, output inactive).*

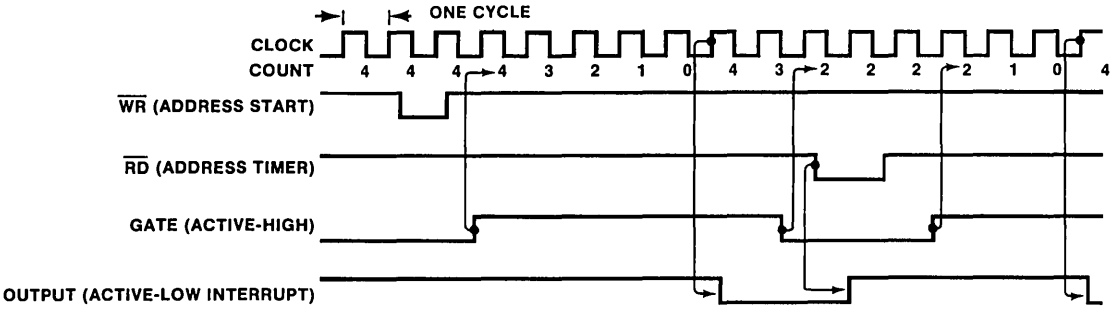
FIGURE 4. Timer Mode Register (Mode Selection)

Timer Mode Examples

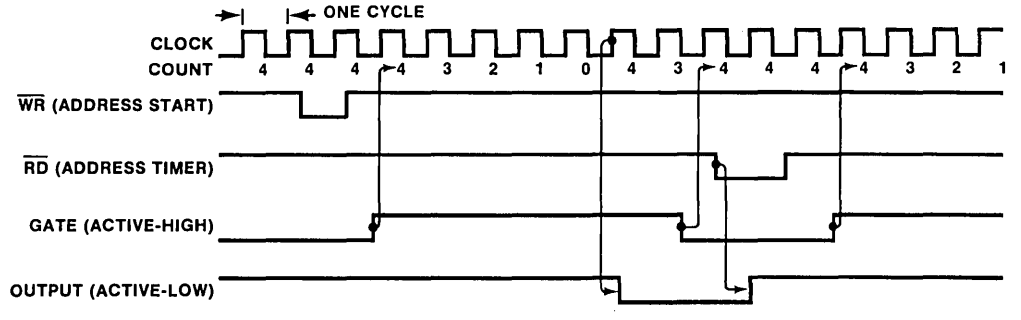
(Modulus register is loaded with 0004 for these examples.)



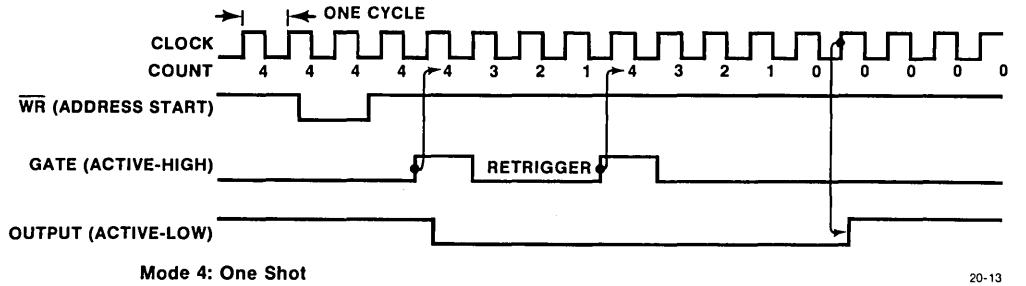
Mode 1: Counter 20-10



Mode 2: Timer 20-11



Mode 3: Timer 20-12



Mode 4: One Shot 20-13

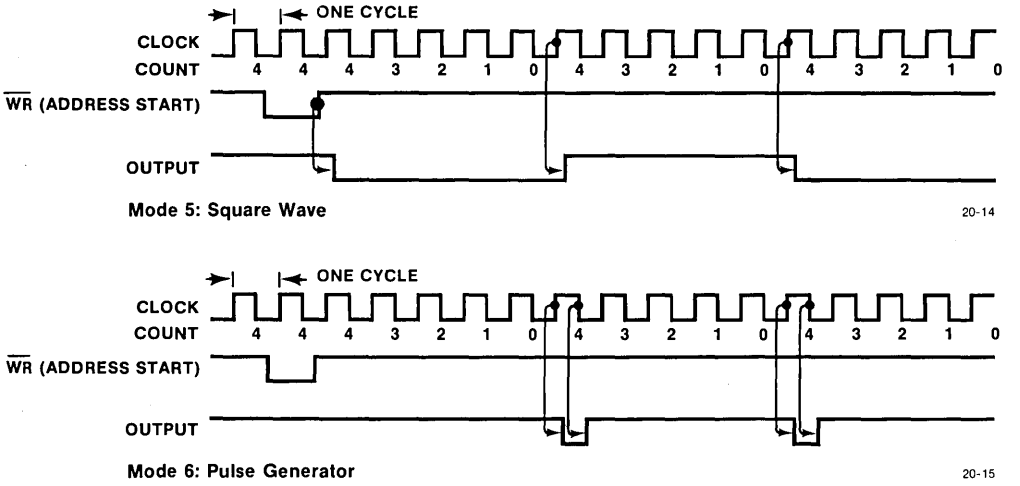


FIGURE 5. Timer Mode Examples

Prescale

The timer output is affected by the prescale selection. The output responds to the timer clock, not the incoming clock (TIN); so, TOUT will be prescaled by the same value as the timer. Although the 16-bit prescaled count of the timer may be read, the internal value of the prescaler cannot be read by the user. A "00" for either timer represents ÷ 1 (no prescale). Timer 0 has the two possibilities of ÷ 2 or ÷ 64:

TIMER Bit 4 3	Prescale
0 0	÷ 1
0 1	÷ 2
1 x	÷ 64

Timer 1 has only the ÷ 2 prescale available; TMR bit 4 is a "don't care."

TIMER Bit 4 3	Prescale
x 0	÷ 1
x 1	÷ 2

Read/Write

A two-byte word (or a single byte when one byte is a "don't care") may be read from or written to the timers. To program the timer buffers, TMR bit 5 must be set as follows:

0 — Double byte read or write Low byte first, then high byte (see note)

1 — Single byte read or write Low byte only - high byte "don't care" or high byte only with low byte "don't care"

NOTE: *The order of low byte first, high byte second must be maintained for proper Read/Write communications*

Gate Input Polarity and Output Sense

Either timer (or both) programmed in one of the gated modes (010, 011, or 100) can select the gate input TG as true or complement. A "1" in bit 6 of the TMR selects the complement of TG; a "0"selects the true value. Similarly, TMR, bit 7 selects active-low output if programmed with a "0", or active-high if programmed "1". See *Figure 6.*

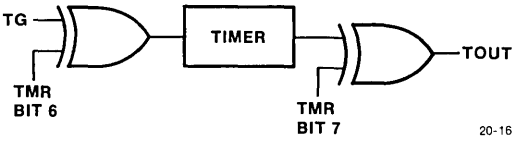


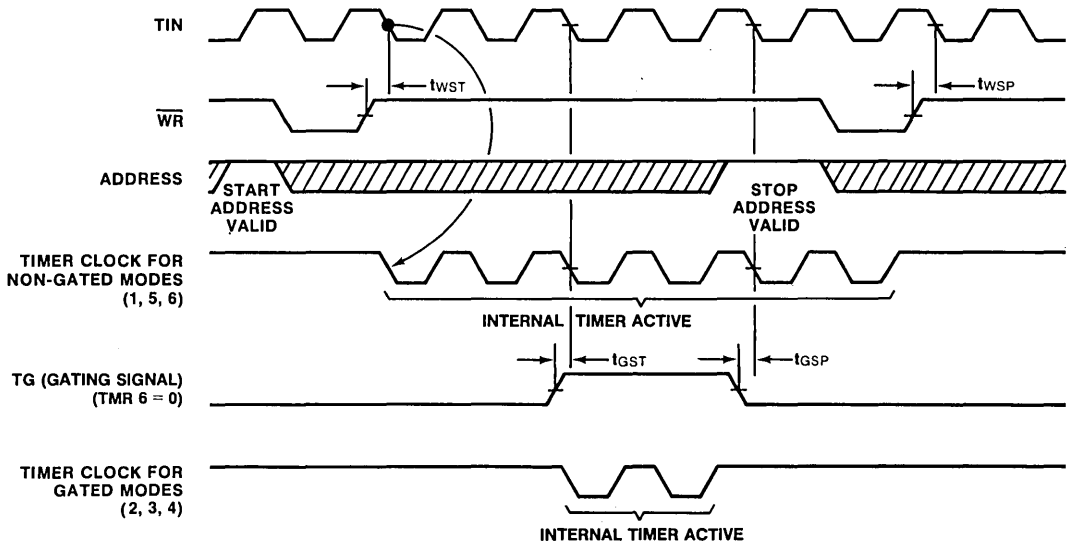
FIGURE 6. Gate Input Polarity and Output Sense

STARTING/STOPPING THE TIMERS

To start either timer, the applicable START address must be written into I/O location xxx10101 for Timer 0, and into xxx10111 for Timer 1. The data for the start (or stop) write is "don't care" so that the trailing edge of the $\overline{WR}$ strobe may be referenced for accurate timing applications. The timer will start as soon as this edge propagates within the chip.

The timer will start only when the gate (TG) is active if the timer is in a gated mode.

Stopping the timer occurs with the trailing edge of the $\overline{WR}$ strobe when the applicable STOP address is written into I/O location xxx10100 for Timer 0, and into xxx10110 for Timer 1. The timer will also stop when the gate goes to an inactive state if the timer is in mode 2 or 3. The timer is always halted when the TMR bits 0, 1, and 2 are all "1's" or all "0's". START/STOP timing is illustrated in *Figure 7*.



NOTES: For Mode 4 (one shot), only start-timing applies

t_{WST} - $\overline{WR}$ set-up for starting timer 150 ns
t_{WSP} - $\overline{WR}$ set-up for stopping timer 150 ns
t_{GST} - TG (gate) set-up for starting timer 100 ns
t_{GSP} - TG (gate) set-up for stopping timer 100 ns

NOTE: Diagonal lines indicate interval of invalid data.

FIGURE 7. Start/Stop Timing

20-17

TIN, TOUT, AND TG

Timer 0 has dedicated pins for its clock, T0IN, and its output, T0OUT. Timer 1 must borrow its input and output pins from Port C. This is accomplished by writing to the TMR for Timer 1. If mode 1, 2, 3, 4, 5, or 6 is specified in TMR 1, the pins from Port C (PC-3, PC-4, and PC-5) are automatically made available to the timer(s) for gating (TG), T1IN, and T1OUT, respectively. These pins are also taken from Port C any time Timer 0 is in mode 2, 3, or 4. This is also automatically accomplished by writing TMR 0.

TG (PC-3), the timer gate, is used to hardware control the starting/stopping (or triggering) of the timers. Either timer or both may use TG.

The DDR for Port C must be programmed with the correct I/O direction for TG and the input and output of Timer 1. Several examples showing various programmed conditions for the timers are shown in *Table 3.*

Table 3.

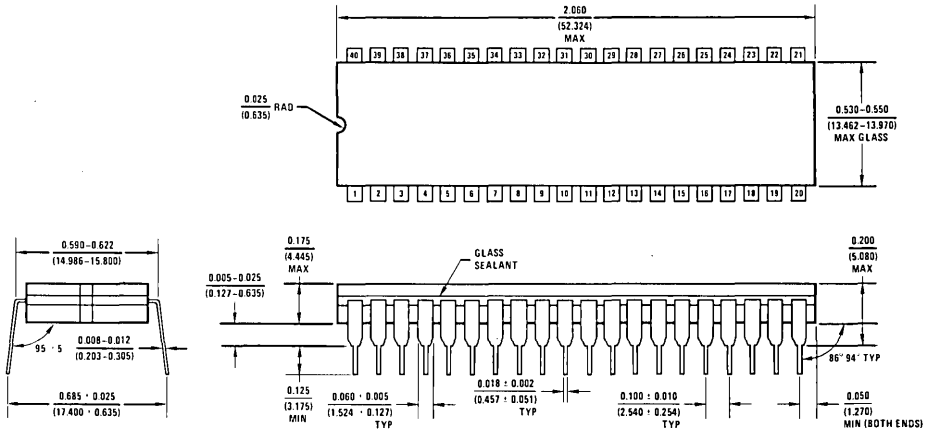
Timer Programming Selection

Mode Register Bit								Output Sense Active L/H	Gate Polarity Active L/H	Mode Description Read/Write Mode Single/Double Byte	Prescale Value	Timing Mode	Port C DDR					
7	6	5	4	3	2	1	0						5	4	3	2	1	0
TIMER 0																		
x	x	x	x	x	0	0	0	x	x	x	x	0	x	x	x	x	x	x
0	x	0	0	0	0	0	1	L	x	D	÷ 1	1	x	x	x	x	x	x
1	x	0	1	x	1	1	0	H	x	D	÷ 64	6	x	x	x	x	x	x
1	0	0	0	1	1	0	0	H	H	D	÷ 2	4	1	0	0	x	x	x
0	1	1	0	0	0	1	0	L	L	S	÷ 1	2	1	0	0	x	x	x
TIMER 1																		
x	x	x	x	x	1	1	1	x	x	x	x	7	x	x	x	x	x	x
0	x	0	x	0	0	0	1	L	x	D	÷ 1	1	1	0	0	x	x	x
1	0	1	x	1	1	0	1	H	H	S	÷ 2	5	1	0	0	x	x	x
0	1	0	x	0	0	1	1	L	L	D	÷ 1	3	1	0	0	x	x	x

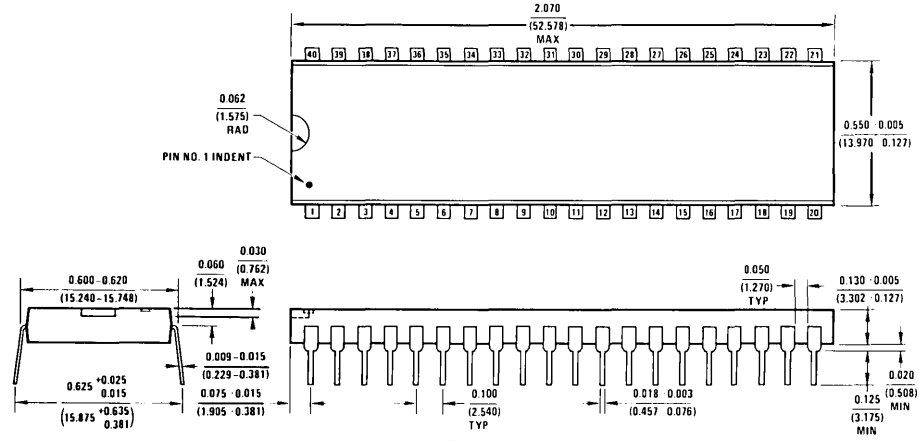
x = "don't care"

Physical Dimensions

inches (millimeters)



Ceramic Dual-In-Line Package [Cerdip (J)]



Plastic Dual-in-Line Package (N)

Ordering Information

NSC810XX

- **I** = **Industrial Temperature (-40°C to +85°C)**
- **M** = **Military Temperature (Availability to be announced)**

- **N** = **Plastic Package**
- **J** = **Ceramic Dual-In-Line Package [Cerdip (J)]**

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty, Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

NSC830 ROM-I/O; NSC831 I/O Only

General Description

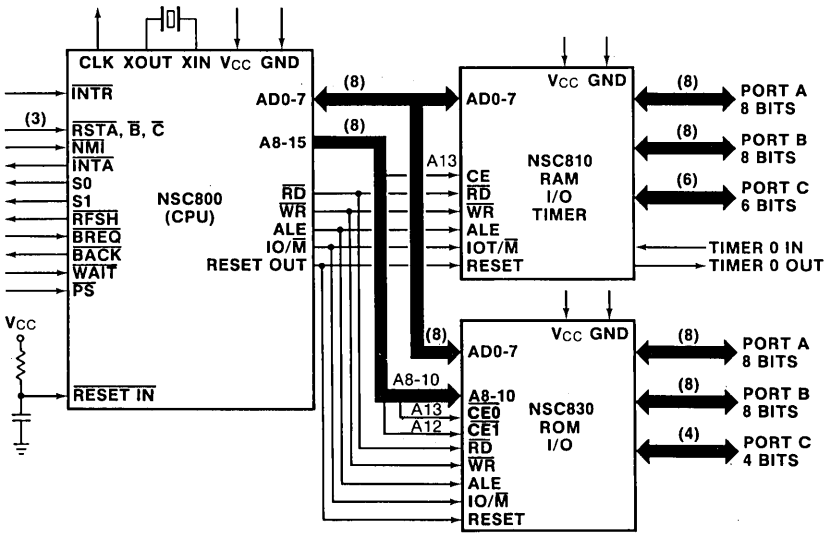
The NSC830 is a ROM-I/O device contained in a standard 40-pin, dual-in-line package. The chip, which is fabricated using P²CMOS silicon gate technology, functions as a memory, and an input/output peripheral interface device. The memory is comprised of 16,384 bits of ROM organized as 2048 x 8. The I/O portion consists of 20 programmable input/output bits arranged as three separate ports, with each bit individually definable as an input or output. The port bits can be set or cleared individually and can be written to or read from in bytes. Several types of strobed mode operations are available through Port A.

The NSC831 I/O Only is similar to the NSC830 except it has no ROM. The NSC831 is useful for prototyping work prior to ordering the NSC830, and when on-chip ROM is not required.

Features

- 2K x 8 Read Only Memory
- Three Programmable I/O Ports
- Single 5V Power Supply
- Very Low Power Consumption
- Fully Static Operation
- Single-Instruction I/O Bit Operations
- Directly Compatible with NSC800 Family
- Strobed Modes Available on Port A

NSC800 Microcomputer Family Block Diagram



μPG-FL5M109/Printed in U.S. A.

Absolute Maximum Ratings

Storage Temperature Range.....-65° C to 150° C
Voltage on Any Pin with Respect to Ground... -0.3V to V_{CC} + 0.3V
Lead Temperature (Soldering, 10 seconds).....300° C
Power Dissipation 1W

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40° C to +85° C	5V
Commerical	0° C to +70° C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating voltage range components to be announced.

DC Electrical Characteristics

T_A = 0° C to + 70° C, V_{CC} = +5V ± 10%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}	Logical 1 Input Voltage	0.7V_{CC}		V_{CC}	V	
V_{IL}	Logical 0 Input Voltage	0		0.3 V_{CC}	V	
V_{OH}	Logical 1 Output Voltage	2.4 V_{CC} -0.5			V	I_{OH} = -1.0mA I_{OUT} = -10μA
V_{OL}	Logical 0 Output Voltage	0 V_{CC} -0.5		0.4	V V	I_{OL} = 2mA I_{OUT} = 10μA
I_{IL}	Input Leakage Current	-1.0		1.0	μA	0 ≤ V_{IN} ≤ V_{CC}
I_{OL}	Output Leakage Current	-1.0		1.0	μA	0 ≤ V_{IN} ≤ V_{CC}
I_{CC}	Active Supply Current		5		mA	I_{OUT} = 0; f $_{(XIN)}$ = 5MHz

AC Electrical Characteristics

T_A = 0°C to + 70°C, V_{CC} = +5V ± 10%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
t_{ACC}	Access Time from ALE (Note 1)			250	ns	CL = 100 pF
t_{AS}	Address Set-Up time	50			ns	
t_{AH}	Address Hold Time	50			ns	
t_{RDD}	Data Bus Disable	0		75	ns	CL = 100 pF
t_{ARW}	ALE to Read or Write Strobe	75			ns	
t_{RD}	Read Strobe Width	200			ns	
t_{WR}	Write Strobe Width	200			ns	
t_{ALE}	ALE Strobe Width (high)	100			ns	
t_{DS}	Data Set-up Time	50			ns	
t_{DH}	Data Hold Time	75			ns	
t_{RWA}	Read or Write to Next ALE Time	100			ns	
t_{PS}	Peripheral Data Set-up Time	100			ns	
t_{PH}	Peripheral Data Hold Time	100			ns	
t_{DO}	Port Data Output Valid			200	ns	Port Loading = 100 pF
t_{SB}	$\overline{STB}$ to Buffer Full Valid			250	ns	
t_{SW}	Minimum $\overline{STB}$ Width (Note 2)	200			ns	
t_{SS}	Peripheral Data Set-up Time	100			ns	
t_{SH}	Peripheral Data Hold with Respect to STB	100			ns	
t_{SI}	$\overline{STB}$ to $\overline{INTR}$ Output			250	ns	
t_{RB}	$\overline{RD}$ to BF Output			250	ns	
t_{RI}	$\overline{RD}$ to $\overline{INTR}$ Output			250	ns	
t_{WB}	$\overline{WR}$ to BF Output			250	ns	
t_{WI}	$\overline{WR}$ to $\overline{INTR}$ Output			250	ns	
t_{PE}	Peripheral Bus Enable			200	ns	
t_{PZ}	Peripheral Bus Disable (TRI-STATE)			150	ns	

Note: 1 See Read and Write Waveforms.

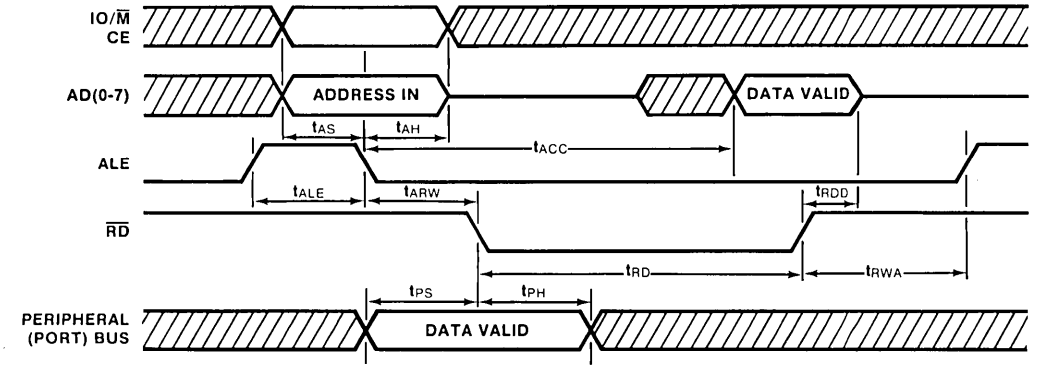
Note: 2 See Strobe Mode Input and also Strobed Mode Output Waveforms.

Capacitance

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance		4		pF	
C_{OUT}	Output Capacitance		6		pF	

Timing Waveforms

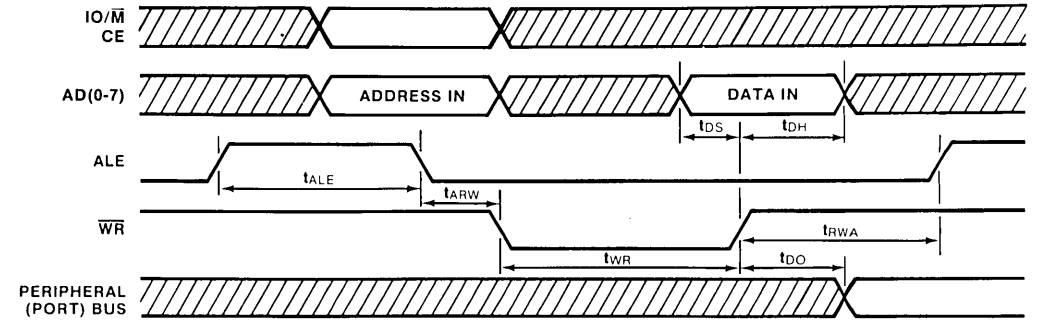
Read Cycle



NOTE: Diagonal lines indicate interval of invalid data.

(Read from ROM, or Port) 46-1

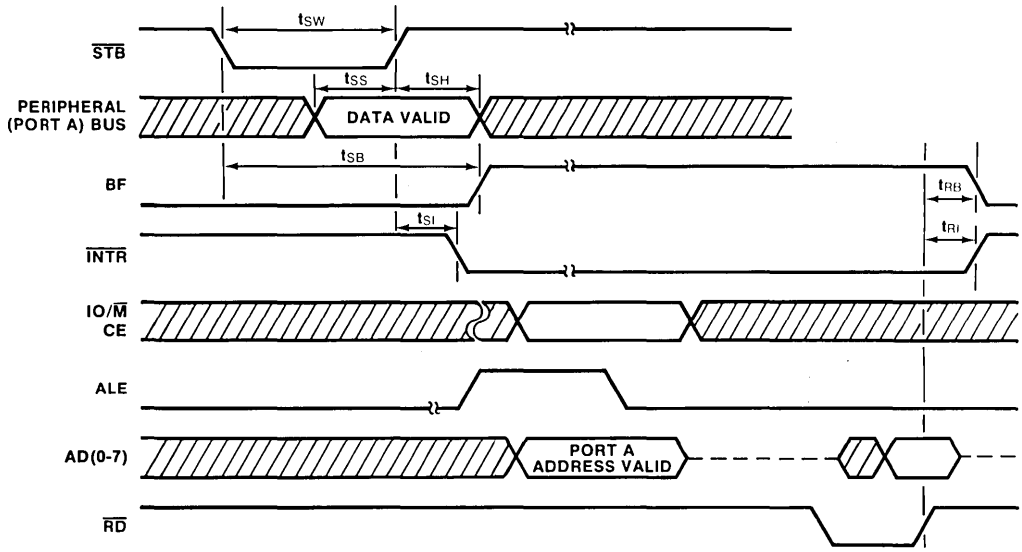
Write Cycle



NOTE: Diagonal lines indicate interval of invalid data.

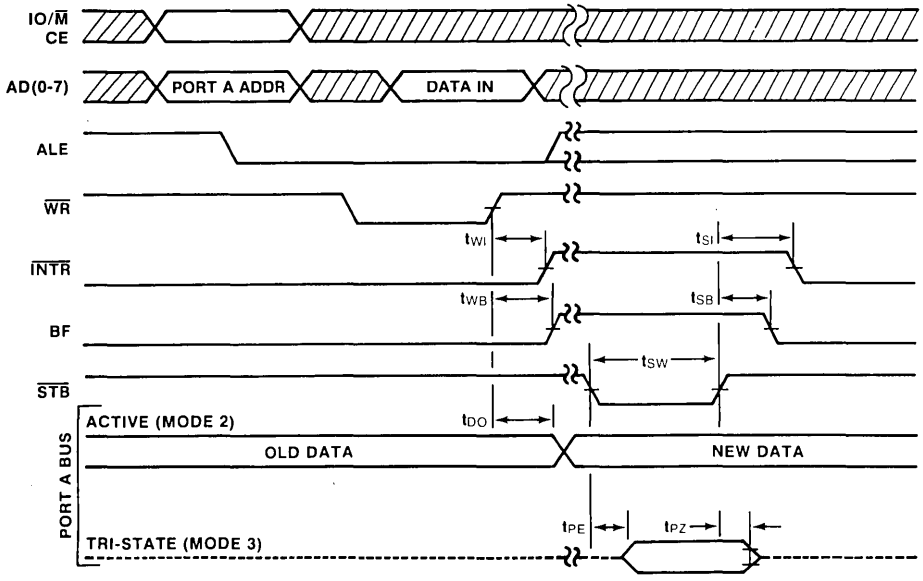
(Write to Port) 46-2

Handshake Timing



NOTE: Diagonal lines indicate interval of invalid data.

Strobed Mode Input 46-3



NOTE: Diagonal lines indicate interval of invalid data.

Strobed Mode Output 46-4

NSC830 Functional Pin Description

The following describes the function of all NSC830 input/output pins. Some of these descriptions reference internal circuits.

INPUT SIGNALS

Master Reset (RESET): An active-high input on the RESET pin initializes the chip causing the three I/O ports (A, B and C) to revert to the input mode. The three ports, the three data direction registers and the mode definition register are reset low (0).

Input/Output/Memory Select (IO/$\overline{M}$): The IO/$\overline{M}$ pin is a latched, select input line. A high (1) input selects the I/O portion of the chip; a low (0) input selects the ROM portion of the chip. The select input is latched by the trailing edge (high to low transition) of the ALE signal.

Chip Enable ($CE_0/\overline{CE_0}$, $\overline{IOR}/CE_1/\overline{CE_1}$): The chip enable inputs are mask programmable at the factory. The CE inputs permit the use of multiple NSC830's in a system without using a chip select decoder. The CE inputs must be active at the falling edge of ALE. At ALE time, the CE inputs are latched to provide access to the NSC830. The $\overline{IOR}$ input performs the same function as the combination of IO/$\overline{M}$ input high and the $\overline{RD}$ input low.

Read ($\overline{RD}$): When the $\overline{RD}$ (or the $\overline{IOR}$, when mask programmed) input is an active low, data is read from the AD(0-7)bus. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD (0-7) bus is in the high impedance state.

Write ($\overline{WR}$): When the CE inputs are active, and the IO/$\overline{M}$ input is high, an active low $\overline{WR}$ input causes the selected output port to be written with the data from the AD(0-7) bus.

Address Latch Enable (ALE): The trailing edge (high to low transition) of the ALE input signal latches the address/data present on the AD(0-7) bus, A(8-10) bus, plus the input control signals on IO/$\overline{M}$, $CE_0/\overline{CE_0}$, and $CE_1/\overline{CE_1}$.

Address Bus A (8-10): The high-order bits of the ROM address are input on this 3-bit bus and are latched by the high-to-low transition of the ALE input. These bits do not affect the I/O operations.

Power (V_{CC}): 5-volt supply.

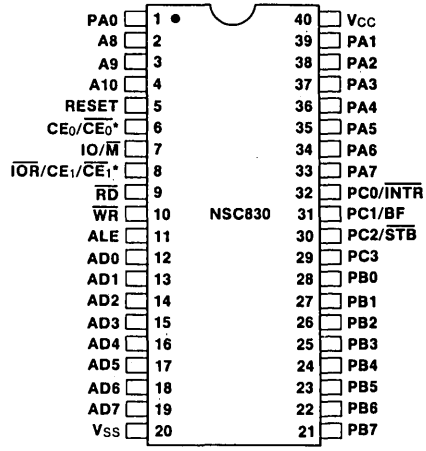
Ground (V_{SS}): Ground reference.

INPUT/OUTPUT SIGNALS

Bidirectional Address/Data Bus AD (0-7): The lower 8 bits of the ROM or I/O address are applied to these pins, and latched by the trailing edge of ALE. During read operations, 8 bits are present on these pins, and are read when $\overline{RD}$ or $\overline{IOR}$ is low. During an I/O write cycle, Port A, B, or C is written with the data present on this bus at the trailing edge of the $\overline{WR}$ strobe.

Ports A, B, C [PA (0-7), PB (0-7), PC (0-3)]: These are general purpose I/O pins. Their input/output direction is determined by the contents of the Data Direction Registers (DDRs).

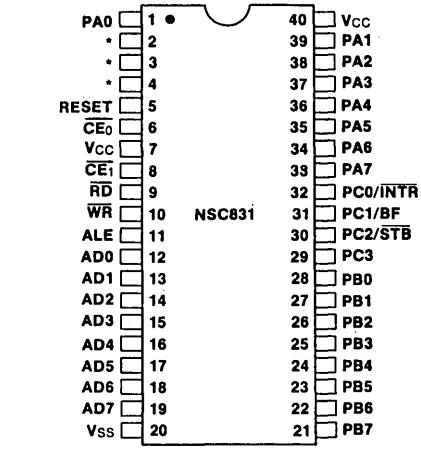
PIN CONFIGURATION



*NOTE: Pin 6 is mask programmable as CE_0 or $\overline{CE_0}$.
Pin 8 is mask programmable as $\overline{IOR}$, CE_1, or $\overline{CE_1}$.

46-6

PIN CONFIGURATION



*NOTE: Tie pins 2, 3 and 4 to either V_{CC} or V_{SS}.

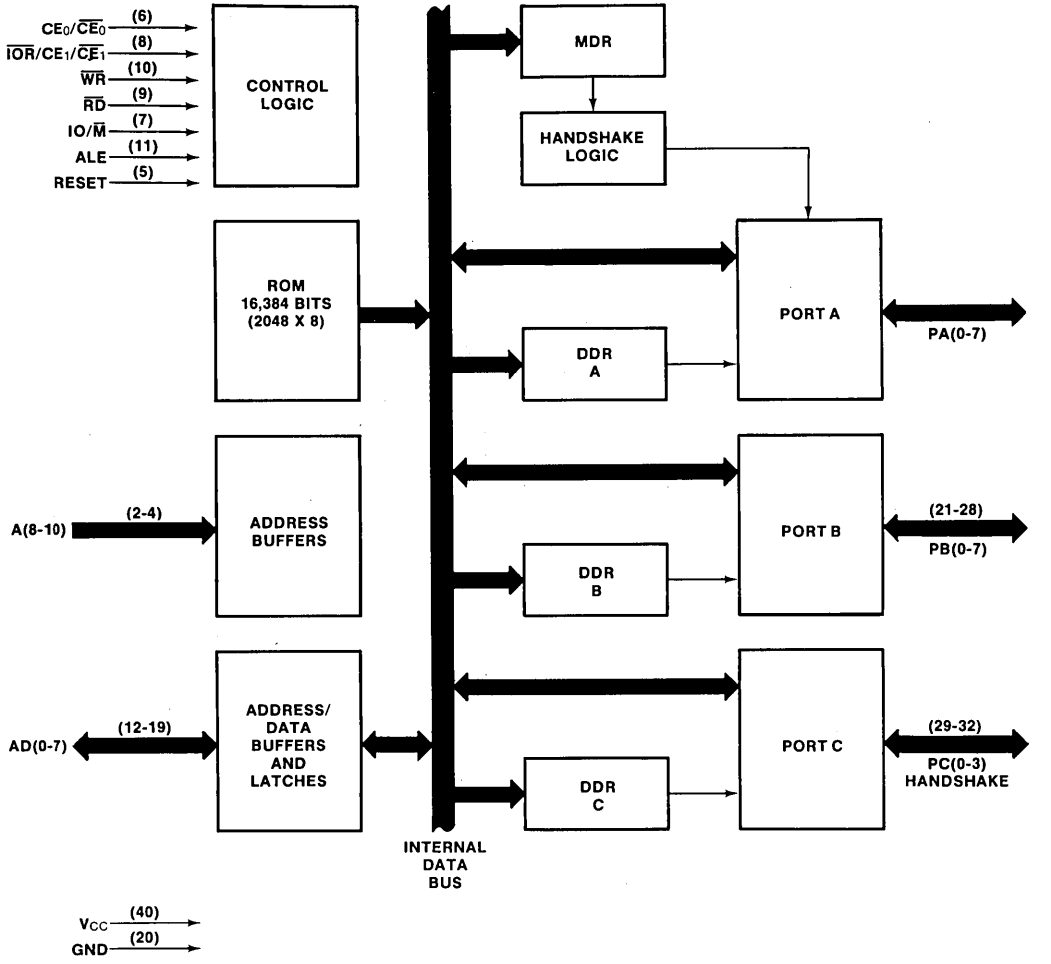
46-7

NSC830 Functional Description

Refer to *Figure 1* for a detailed block diagram of the NSC830, while reading the following paragraphs.

Read Only Memory (ROM): The memory portion of the ROM-I/O is accessed by an 11-bit address input to pins AD (0-7) and A (8-10). The IO/$\overline{M}$ input must be low (ROM select) and the chip enable inputs in the active programmed state at the falling edge of ALE to address the ROM. Timing for ROM read and write operations is shown in the timing diagrams.

Input/Output (I/O): The I/O portion of the NSC830 contains three sets of I/O called Ports. There are two ports (A and B) which contain 8 bits each and one port (Port C) which has 4 bits. Any bit or combination of bits in a port may be addressed with Set or Clear commands. A port can also be addressed as an 8-bit word (4 bits for Port C). All ports share common functions of Read, Write, Bit-Set and Bit-Clear. Additionally, Port A is programmable for strobed (handshake) mode input or output. Port C has a programmable second function for each bit associated with strobed modes. *Table 1* defines the address location of the ports, and control registers.



NOTE: Applicable pinout numbers are included within parentheses.

FIGURE 1. NSC830 Block Diagram

46-5

Table 1. I/O and Address Designations

8-Bit Address Field (Bits 7 6 5 4 3 2 1 0)	Designation I/O Port, etc.	R (Read) W (Write)
x x x x 0 0 0 0	Port A (byte)	R/W
x x x x 0 0 0 1	Port B (byte)	R/W
x x x x 0 0 1 0	Port C (byte)	R/W
x x x x 0 0 1 1	Not Used	—
x x x x 0 1 0 0	DDR - Port A	W
x x x x 0 1 0 1	DDR - Port B	W
x x x x 0 1 1 0	DDR - Port C	W
x x x x 0 1 1 1	Mode Definition Reg.	W
x x x x 1 0 0 0	Port A - Bit Clear	W
x x x x 1 0 0 1	Port B - Bit Clear	W
x x x x 1 0 1 0	Port C - Bit Clear	W
x x x x 1 0 1 1	Not Used	—
x x x x 1 1 0 0	Port A - Bit Set	W
x x x x 1 1 0 1	Port B - Bit Set	W
x x x x 1 1 1 0	Port C - Bit Set	W
x x x x 1 1 1 1	Not Used	—

x = don't care

MODE DEFINITION REGISTER (MDR)

The Mode Definition Register (MDR) defines the operating mode for Port A. While Ports B and C are always in the basic I/O mode, there are four operating modes for Port A:

Mode 0 - Basic I/O (Input or Output)
Mode 1 - Strobed Mode Input
Mode 2 - Strobed Mode Output — Active Peripheral Bus
Mode 3 - Strobed Mode Output — TRI-STATE (high impedance) Peripheral Bus

The MDR has the I/O address assignment xxx00111. The bit configuration for the mode selection is illustrated below:

MODE \ BIT	7	6	5	4	3	2	1	0
0	x	x	x	x	x	x	x	0
1	x	x	x	x	x	x	0	1
2	x	x	x	x	x	0	1	1
3	x	x	x	x	x	1	1	1

x = don't care

DATA DIRECTION REGISTERS (DDR)

Each port bit has a data direction register (DDR) which defines the I/O state of the bit. The bit is configured as an input if a "0" is written into its DDR, or as an output if a "1" is written. The DDR bits cannot be individually written to; the entire DDR byte is affected by a write to the DDR address. Thus all data must be consistent with the direction desired for each port.

Any write or read operations on a port contradicting the DDR will not affect the port output or input. However, a read of a port bit defined as an output will cause a read from the output latch, and a write to a port bit defined as an input will modify the output latch.

PORT FUNCTIONS - BASIC I/O

Basic I/O is the mode of operation of Ports B and C and mode 0 of Port A (defined by the MDR). Read, write, and bit operations can be executed in the basic I/O mode. The timing for basic input and basic output modes is shown in the AC Characteristics tables.

When a read occurs the information is latched from the peripheral bus on the leading (falling) edge of the $\overline{RD}$ strobe. When a write occurs the port bus is modified after the trailing (rising) edge of the $\overline{WR}$ strobe with data from the AD bus. Port output data remains valid on the output pin from one trailing edge of $\overline{WR}$ strobe to the trailing edge of the next $\overline{WR}$ strobe.

BIT OPERATIONS

The I/O features of the RAM-I/O allow modification of a single bit or several bits of a port with Bit-Set and Bit-Clear (see *Figure 2*). The address is set up to indicate that a bit set (or clear) is taking place. The incoming data on the address/data bus is latched at the trailing edge of the $\overline{WR}$ strobe and is treated as a mask. All bits containing "1's" will cause the indicated operation to be performed on the corresponding port bit. All bits of the data mask with "0's" cause the corresponding port bits to remain unchanged. Three sample operations are given, using Port B as an example:

Operation	Set B7	Clear B2 & B0	Set B4, B3 & B1
Address	xxx01101	xxx01001	xxx01101
Data	10000000	00000101	00011010
Port Pins			
Prior State	00001111	10001111	10001010
Next State	10001111	10001010	10011010

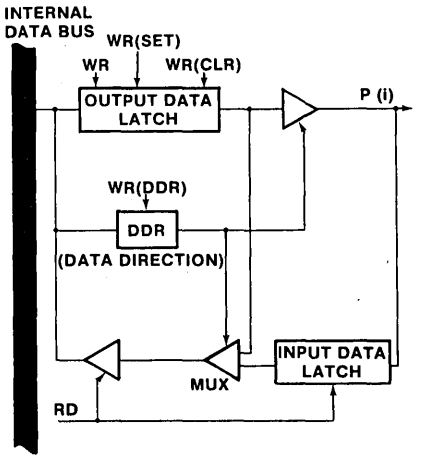


FIGURE 2. Block Diagram for Port Bit (i)

20-8

PORT A - STROBED (HANDSHAKE) MODE

Port A can be programmed (via the MDR) into one of 3 types of strobed mode for handshake communication with intelligent peripherals. When Port A is in mode 1, 2, or 3 (see description of MDR), Port C pins 0, 1, and 2 are used as signals to and from the peripheral and to the CPU, controlling handshake operations. These control signals are designated $\overline{STB}$, BF, and $\overline{INTR}$. Bit Set and Clear operations are not allowed on these bits of Port C when Port A is in strobed mode. Timing parameters and timing diagrams are detailed under AC Characteristics.

$\overline{INTR}$ (Strobe Mode Interrupt) is an active-low interrupt from the I/O to the CPU. In strobed Input Mode, the CPU reads the valid data at Port A to clear the interrupt. In strobed Output Mode, the CPU clears the interrupt by writing to Port A.

The interrupt is enabled or disabled by setting or clearing Bit 2 of Port C.

$\overline{STB}$ (Strobe) is an active-low input from the peripheral device, signaling that data-transfer is about to begin. This strobe is interpreted as an "output request" if Port A is in a strobed output mode, or as a "data-valid" signal if Port A is in strobed input mode.

BF (Buffer Full) is an output from the I/O to the peripheral signaling that data transfer is complete. In strobed input mode this strobe indicates that data is received into Port A and that no further data should be transmitted by the peripheral device until the port has been read (emptied). In strobed output mode the BF indicates that the request from the peripheral has been processed by the CPU and the valid data now appears in Port A.

The bits of Port C that are used for handshake control of Port A (bits C0, C1, & C2) must be direction-defined appropriately in the DDR. Also, the DDR of Port A must be consistent with the mode specified in the MDR. Register set-up configurations for the three handshake modes are illustrated in *Table 2.*

Table 2. Mode Definition Register Configurations

Mode	MDR	DDR Port A	DDR Port C	Port C Output Latch
Strobed Input	xxxxxx01	00000000	xxx011	xxx1xx
Strobed Output (Active)	xxxxx011	11111111	xxx011	xxx1xx
Strobed Output (TRI-STATE)	xxxxx111	11111111	xxx011	xxx1xx

APPROVED FORMATS FOR CUSTOM PROGRAMMED PARTS

INPUT MEDIUM:

2716 EPROM
2708 EPROM
PAPER TAPE

IMPORTANT - EPROM LABELLING

Only one customer program may be included in a single order. The following method must be used to identify the EPROMs comprising a program.

a) The EPROMs used for storing a custom program are designated as shown:

2716:	Block A	0-2047
2708:	Block A	0-1023
	Block B	1024-2047

b) All EPROMs must be labelled (stickers, paint, etc.) with this block designation plus a customer assigned print or identification number.

Example:

1) Customer Data
   - Custom Program Length - 2K
   - Medium - Two 2708's
   - Customer Print or I.D. No. C123-45

2) EPROM Labels

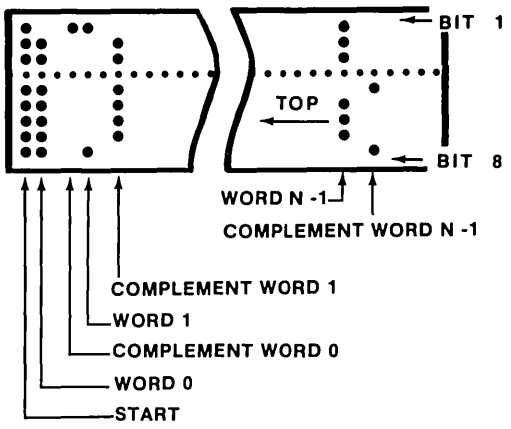
C123-45 A 0-1023	C123-45 B 1024-2047

Paper Tape

Tapes may only be submitted in binary complement format. The following information should be written on the paper tape.

Company Name
Customer Print or I.D. No.
NSC Part No.
A Punch = ("1" or "0")
This is _____ logic (POS or NEG)

BINARY COMPLEMENT FORMAT



NOTE 1: Tape must be blank except for the data words.
NOTE 2: Tape must start with a rubout character.
NOTE 3: Data is comprised of two words, the first being the actual data and the second being the complement of the data.

Verification

You will receive a listing of the options ordered and the input data. If you also wish to receive EPROMs for verification, please send additional blank EPROMs as necessary for this purpose. You can use software (the listing) or hardware (EPROMs) to verify the program.

You will be asked for a GO/NO GO response within one week after you receive the listing.

VERIFICATION LISTING

The verification listing has six sections:

1. A cover sheet with provision for "STOP, DO NOT PROCEED" or "VERIFICATION CERTIFIED" signatures.
2. Description of the options you have chosen.
3. A description of the log designations and assumptions used to process the data.
4. A listing of the data you have submitted.
5. An error summary.
6. A definition of the standard logic definitions for the ROM and the reduced form of the data. This list shows the output word corresponding to each address coded in binary.

Ordering Information for Custom Programmed Parts

The following information must be submitted with each customer microcomputer program. An order will not be processed unless it is accompanied by this information. This form acts as a Traveler from Customer through Customer Service to ROM programming. Please retain a copy of this form to compare against the verification listing. The form will be sent back to the customer by Customer Service.

	National Microcomputer Part Number
	ROM Letter Code (National Use Only)
Name	Date
Address	Customer Print or I.D. No.
City / State / Zip	Purchase Order No.
Telephone ( )	Name of person National can contact (Print)
Authorized Signature	Date

INPUT MEDIUM

See following page for approved formats. Please check the medium you are using.

- ☐ **Paper Tape**
- ☐ **2716 EPROM**
- ☐ **2708 EPROM**

___ **Total number of EPROMs**

OPTIONS FOR NSC830 ROM - I/O

Option 1 = ☐

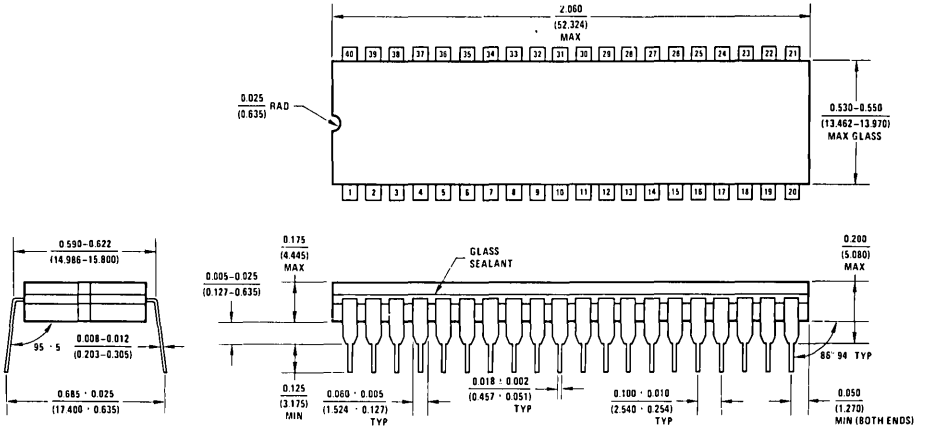
CE_0 Select, enter: 0 for CE_0
1 for $\overline{CE_0}$

Option 2 = ☐

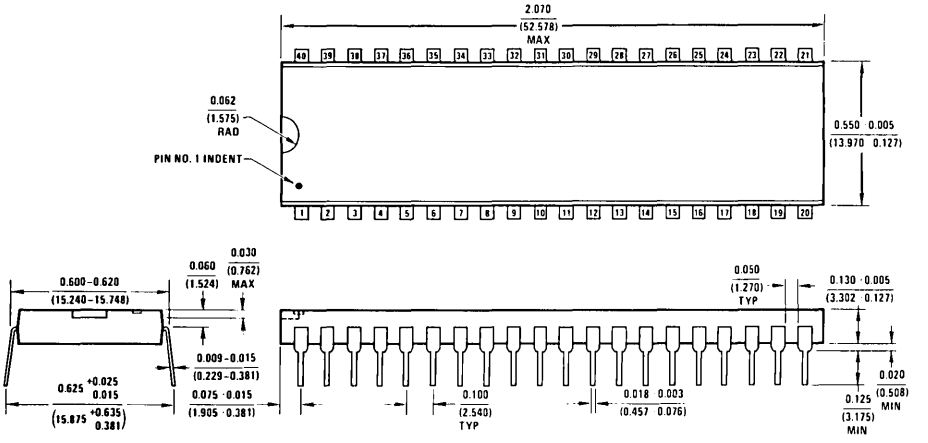
CE_1/IOR Select, enter: 0 for $\overline{IOR}$
1for CE_1
2 for $\overline{CE_1}$

Physical Dimensions

inches (millimeters)



Ceramic Dual-In-Line Package [Cerdip (J)]



Plastic Dual-In-Line Package (N)

Ordering Information
NSC830XX

- I = Industrial Temperature (-40°C to +85°C)
- M = Military Temperature (Availability to be announced)
- N = Plastic Package
- J = Ceramic Dual-In-Line Package [Cerdip (J)]

Ordering Information
NSC831X

- J Package
- N Package

(0 -70°C only)

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

MM74PC00 Quad 2-Input NAND Gate
MM74PC02 Quad 2-Input NOR Gate
MM74PC04 Hex Inverter
MM74PC08 Quad 2-Input AND Gate
MM74PC32 Quad 2-Input OR Gate

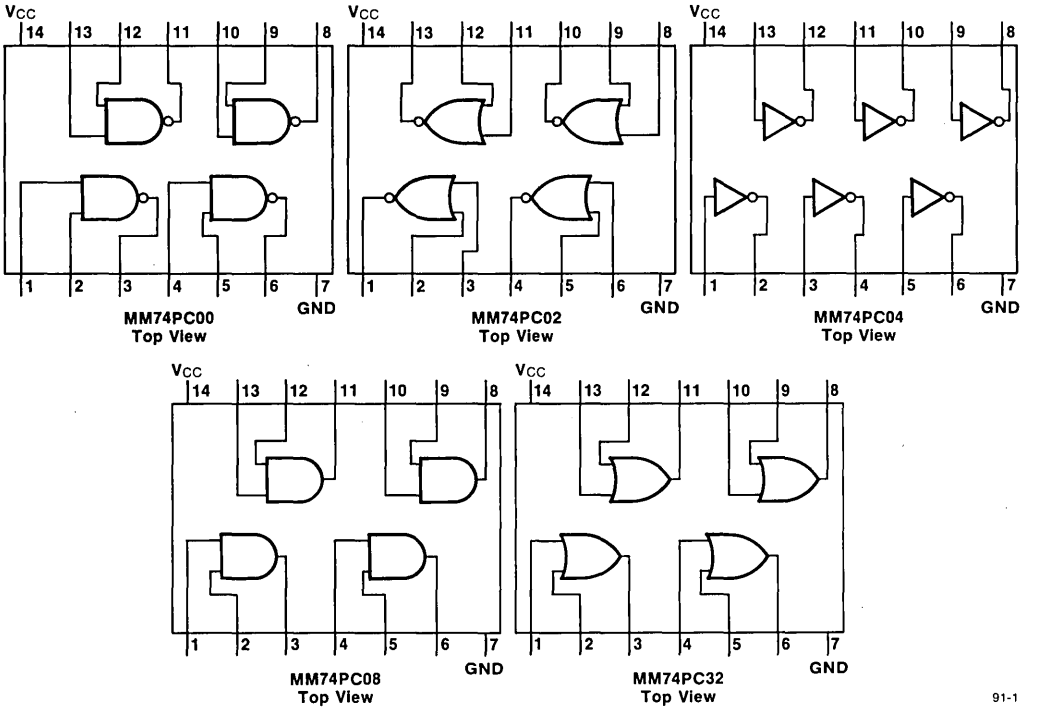
General Description

These Logic gates are fabricated using National's P2CMOS technology. This technology offers wide operating voltage, low power consumption, high noise immunity, and high speed. Function and pinout compatibility with Series-74 devices minimizes design time for those designers already familiar with the 74 logic family. These components may be utilized in completing NSC800 high-performance, low-power designs.

Features

- Single 5V Power Supply
- Low Power Dissipation
- Drive Capability of 100pF Load
- Fully Compatible with CMOS Logic Levels
- TTL Drive Capability
  When V_{CC} = 5V
- Fast: Typical Propagation Delay
  18ns (V_{CC} = 10V, C_L = 15pF)

Connection Diagrams



91-1

Absolute Maximum Ratings

Storage Temperature Range .....-65°C to 150°C
Voltage at Any Pin with Respect to Ground.....-0.3V to V_{CC} +0.3V
Lead Temperature (Soldering, 10 seconds) .....300°C
Power Dissipation.....500mW

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Charactristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40°C to +85°C	5V
Commercial	0°C to +70°C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating voltage range components to be announced.

DC Electrical Characteristics

Min/Max limits apply across temperature range, unless otherwise specified.

Symbol	Parameter	V_{CC}	Conditions	Min	Typ	Max	Units
CMOS TO CMOS							
V_{IH}	Input High Voltage	5V	V_O = 0.5V or 4.5V	4.00			V
V_{IL}	Input Low Voltage	5V	V_O = 0.5V or 4.5V			1.00	V
V_{OH}	Output High Voltage	5V	V_I = V_{CC} or GND	4.95			V
V_{OL}	Output Low Voltage	5V	V_I = V_{CC} or GND			0.05	V
I_{OH}	Output High (Source) Current	5V	V_O = 4.6V, V_I = 0V or 5V	-2.0			mA
I_{OL}	Output Low (Sink) Current	5V	V_O = 0.4V, V_I = 0V or 5V	+2.0			mA
I_{CC}	Supply Current	5V	V_I = V_{CC} or GND All Outputs Open			15	μA
I_I	Input Current	5V	V_I = V_{CC} or GND			±1	μA

CMOS TO LSTTL INTERFACE (BUFFERED OUTPUT GATES) **MM74PC04**
V_{CC} = 5V ± 5% unless otherwise specified. **MM74PC08**

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{I(1)}$	Input High Voltage		V_{CC} -1.5			V
$V_{I(0)}$	Input Low Voltage				0.8	V
$V_{O(1)}$	Output High Voltage	V_I = $V_{I(1)}$ Min or $V_{I(0)}$ Max I_O = -2.0mA	2.4			V
$V_{O(0)}$	Output Low Voltage	V_I = $V_{I(1)}$ Min or $V_{I(0)}$ Max I_O = +2.0mA			0.4	V

DC Characteristics - Cont'd.

CMOS TO LSTTL INTERFACE (UNBUFFERED OUTPUT GATES)
V_{CC} = 5V ± 5% unless otherwise specified.

MM74PC00
MM74PC02
MM74PC32

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{I(1)}$	Input High Voltage	V_O = 4.4V or 0.4V I_O = ± 10μA	V_{CC} -1.5			V
$V_{I(0)}$	Input Low Voltage	V_O = 4.4V or 0.4V I_O = ± 10μA			0.8	V
$V_{O(1)}$	Output High Voltage	V_I = 4.0V or 1.0V I_O = ± 2.0mA	2.4			V
$V_{O(0)}$	Output Low Voltage	V_I = 4.0V or 1.0V I_O = ± 2.0mA			0.4	V

OUTPUT DRIVE
V_{CC} = 5V, T_A = -40°C to +85°C

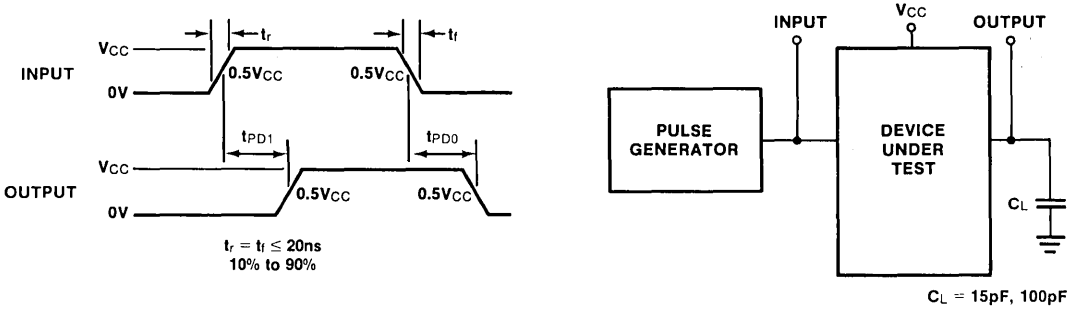
Symbol	Parameter	Conditions	Min	Typ	Max	Units
MM74PC00 Quad 2-Input NAND Gate						
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-10.5			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	3.7			mA
MM74PC02 Quad 2-Input NOR Gate						
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-10			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	4			mA
MM74PC04 Hex Inverter						
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-10.5			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	3.7			mA
MM74PC08 Quad 2-Input AND Gate						
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-10.5			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	3.3			mA
MM74PC32 Quad 2-Input OR Gate						
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-14			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	7			mA

AC Electrical Characteristics

$T_A = 25°C$

Symbol	Parameter	V_{CC}	Conditions	Min	Typ 15pF	Typ 100pF	Max	Units
MM74PC00 Quad 2-Input NAND Gate								
t_{PD0}	Propagation Delay to Logical "0"	5V			18	30		ns
t_{PD1}	Propagation Delay to Logical "1"	5V			18	30		ns
MM74PC02 Quad 2-Input NOR Gate								
t_{PD0}	Propagation Delay to Logical "0"	5V			18	22		ns
t_{PD1}	Propagation Delay to Logical "1"	5V			14	25		ns
MM74PC04 Hex Inverter								
t_{PD0}	Propagation Delay to Logical "0"	5V			12	20		ns
t_{PD1}	Propagation Delay to Logical "1"	5V			12	20		ns
MM74PC08 Quad 2-Input AND Gate								
t_{PD0}	Propagation Delay to Logical "0"	5V			18	28		ns
t_{PD1}	Propagation Delay to Logical "1"	5V			18	28		ns
MM74PC32 Quad 2-Input OR Gate								
t_{PD0}	Propagation Delay to Logical "0"	5V			22	30		ns
t_{PD1}	Propagation Delay to Logical "1"	5V			22	30		ns

Timing Waveforms

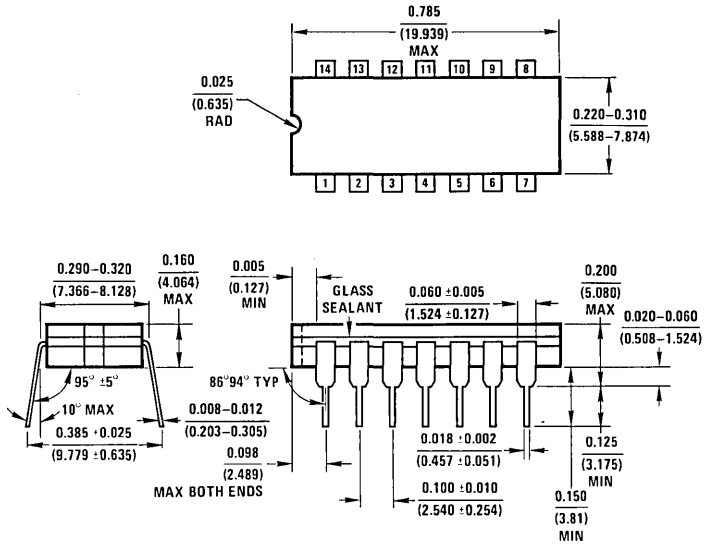


Propagation Delay (CMOS to CMOS)

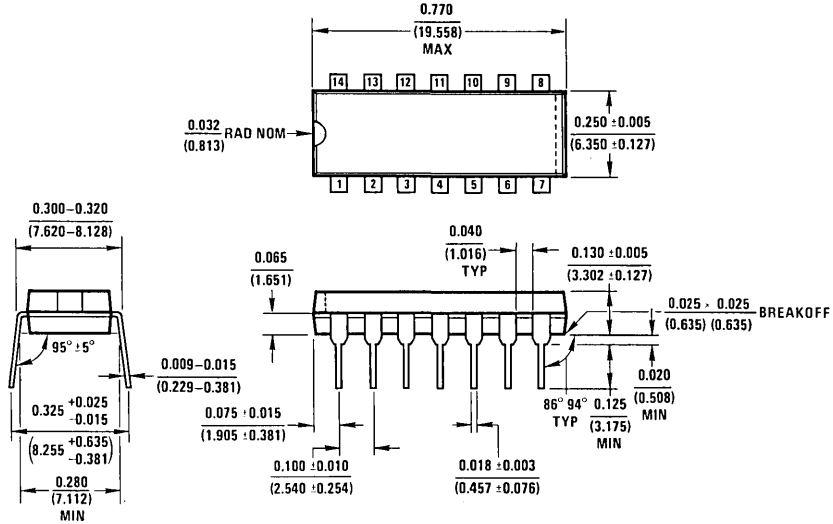
91-2

Physical Dimensions

inches (millimeters)



Ceramic Dual-in-Line Package [Cerdip (J)]



Plastic Dual-in-Line Package (N)

Ordering Information

MM74PC00XX
MM74PC02XX
MM74PC04XX
MM74PC08XX
MM74PC32XX —— I Industrial Temperature (-40°C to +85°C)

N Plastic Package
J Ceramic Dual-in-Line Package [Cerdip (J)]

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

MM74PC74 Dual D Flip-Flop

General Description

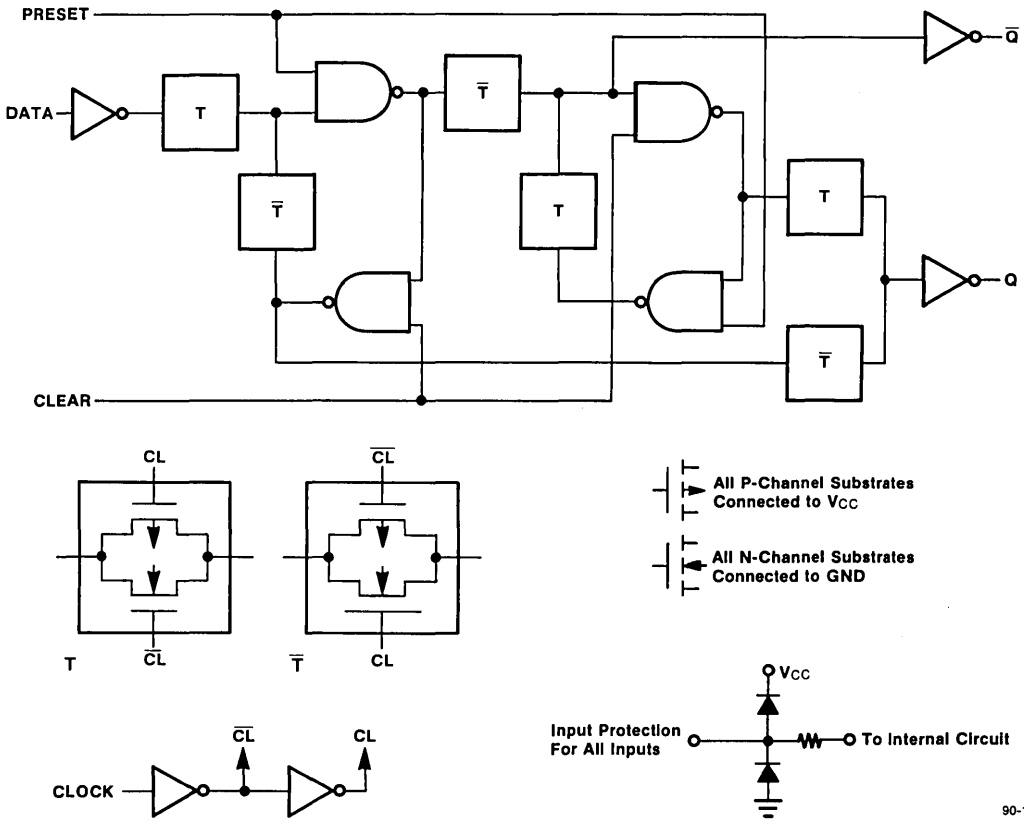
The MM74PC74 Dual D Flip-Flop (edge triggered) is fabricated using National's P²CMOS technology. This technology offers wide operating voltage, low power consumption, high noise immunity, and high speed. Function and pinout compatibility with Series-74 devices minimizes design time for those designers already familiar with the 74 logic family. The MM74PC74 may be utilized in completing NSC800 high-performance, low-power designs.

Each flip-flop has independent data, preset, clear and clock inputs, plus Q and $\overline{Q}$ outputs. The logic level present at the data input is transferred to the output during the rising edge of the clock pulse. Preset or clear status is independent of the clock and is caused by a low level at the preset, or clear input.

Features

- Single 5V Power Supply
- Typical Propagation Delay from Clock of 40ns (@5V)
- Low Power Dissipation
- Drive Capability of 100pF Load
- Fully Compatible with CMOS Logic Levels
- TTL Drive Capability
  When V_{CC} = 5V

Logic Diagram



Absolute Maximum Ratings

Storage Temperature Range .....-65°C to 150°C
Voltage at Any Pin with Respect to Ground.....-0.3V to V_{CC} +0.3V
Lead Temperature (Soldering, 10 seconds) .....300°C
Power Dissipation.....500mW

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Charactristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40°C to +85°C	5V
Commercial	0°C to +70°C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating range components to be announced.

DC Electrical Characteristics

Min/Max limits apply across temperature range, unless otherwise specified.

Symbol	Parameter	V_{CC}	Conditions	Min	Typ	Max	Units
CMOS TO CMOS							
V_{IH}	Input High Voltage	5V	V_O = 0.5V or 4.5V	4.00			V
V_{IL}	Input Low Voltage	5V	V_O = 0.5V or 4.5V			1.00	V
V_{OH}	Output High Voltage	5V	V_I = V_{CC} or GND	4.95			V
V_{OL}	Output Low Voltage	5V	V_I = V_{CC} or GND			0.05	V
I_{OH}	Output High (Source) Current	5V	V_O = 4.6V, V_I = 0V or 5V	-2.0			mA
I_{OL}	Output Low (Sink) Current	5V	V_O = 0.4V, V_I = 0V or 5V	+2.0			mA
I_{CC}	Supply Current	5V	V_I = V_{CC} or GND All Outputs Open			60	μA
I_I	Input Current	5V	V_I = V_{CC} or GND			±1	μA

CMOS TO LSTTL INTERFACE (BUFFERED OUTPUT GATES)
V_{CC} = 5V ± 5% unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{I(1)}$	Input High Voltage	V_O = $V_{O(1)}$ Min or $V_{O(0)}$ Max at specified current	V_{CC} -1.5			V
$V_{I(0)}$	Input Low Voltage	V_O = $V_{O(0)}$ Min or $V_{O(0)}$ Max at specified current			0.8	V
$V_{O(1)}$	Output High Voltage	V_I = $V_{I(1)}$ Min or $V_{I(0)}$ Max I_O = -2.0mA	2.4			V
$V_{O(0)}$	Output Low Voltage	V_I = $V_{I(1)}$ Min or $V_{I(0)}$ Max I_O = +2.0mA			0.4	V

OUTPUT DRIVE

V_{CC} = 5V, T_A = -40° C to +85° C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-16			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	5.8			mA

AC Electrical Characteristics

T_A = 25° C

Symbol	Parameter	V_{CC}	Conditions	Min	Typ 15pF	Typ 100pF	Max	Units
	Propagation Delay to Logical "0" or Logical "1" from Clock to Q or $\overline{Q}$	5V			22	31		ns
	Propagation Delay to Logical "0" from Preset	5V			40	47		ns
	Propagation Delay to Logical "0" from Clear	5V			30	40		ns
	Propagation Delay to Logical "1" from Preset	5V			30	40		ns
	Propagation Delay to Logical "1" from Clear	5V			40	47		ns
t_{SETUP}	Time Prior to Clock Pulse That Data Must Be Present			15				ns
t_{HOLD}	Time After Clock Pulse That Data Must Be Held					10		ns
t_{WL}, t_{WH}	Minimum Clock Pulse Width					50		ns
	Minimum Preset and Clear Pulse Width					50		ns
	Maximum Clock Rise and Fall Time			1.0				μs
	Maximum Clock Frequency			10				MHz

Capacitance*

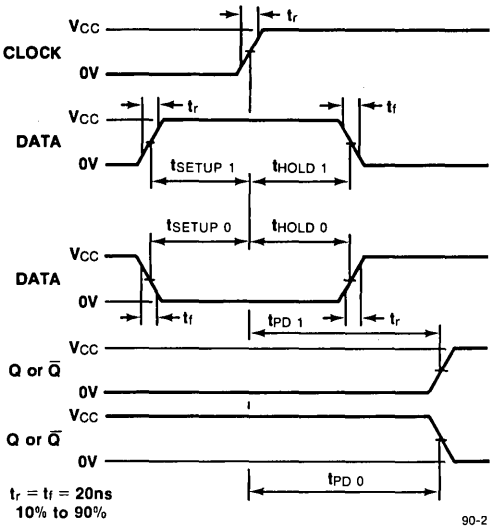
F = 1MHz, V_{BIAS} = 2.5V, V_{CC} = +5V, T_A = 25° C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{IN}	Data Input Capacitance			7		pF
C_{IN}	Clock Input Capacitance			13		pF
C_{OUT}	Q, $\overline{Q}$ Output Capacitance			10		pF

*This parameter is sampled and not 100% tested.

Switching Time Waveforms

CMOS to CMOS



AC Test Circuit

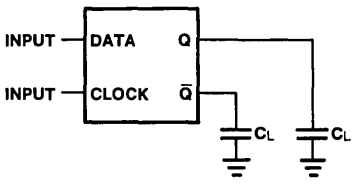
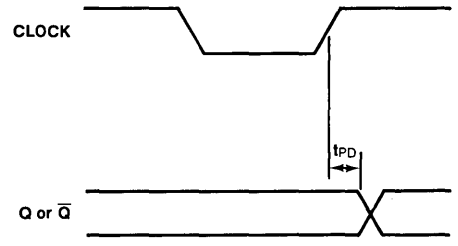


Figure 1 illustrates the calculation of a more useful propagation delay. The figure uses a 5-volt supply with a tolerance of ±10%, ambient temperature of +25° C, and a load capacitance of 100pF. The AC Characteristics table depicts t_{PD}, at 5 volts, 25° C, equalling 31ns. Use the graph in *Figure 1* to get the degradation multiple for 150pF. The number shown is 1.13. The adjusted propagation delay is, therefore 31 x 1.13 or 35ns.

Propagation Delays



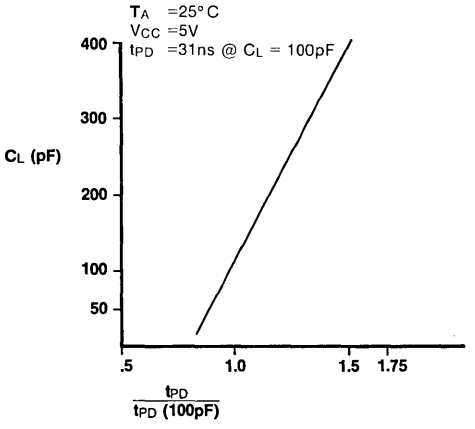
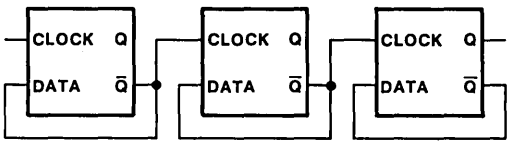


FIGURE 1. Normalized Typical Propagation Delay vs Load Capacitance

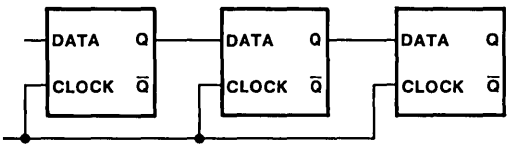
90-4

Typical Applications

Ripple Counter (Divide by 2^n)



Shift Register

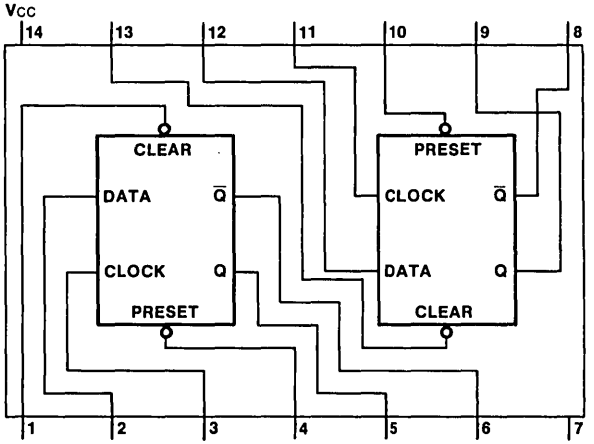


90-6

Truth Table

Preset	Clear	Q_n	$\overline{Q}_n$	
0	0	--	--	Not Allowed
0	1	1	0	
1	0	0	1	
1	1	*Q_n	*$\overline{Q}_n$	

*No Change in output from previous state.



Connection Diagram

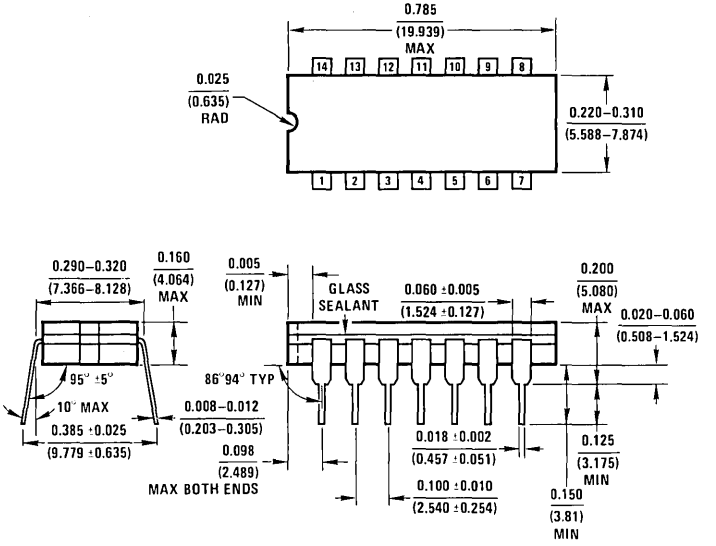
Top View

Note: A logic "0" on clear sets Q to logic "0." A logic "0" on preset sets Q to logic "1."

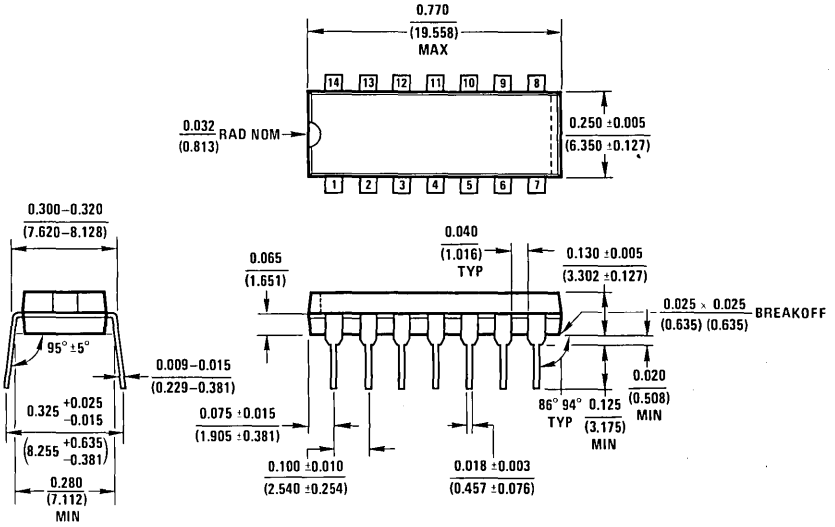
90-5

Physical Dimensions

Inches (millimeters)



Ceramic Dual-in-Line Package [Cerdip (J)]



Plastic Dual-in-Line Package (N)

Ordering Information

MM74PC74XX

- I = Industrial Temperature (-40°C to +85°C)
- N = Plastic Package
- J = Ceramic Dual-in-Line Package [Cerdip (J)]

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

MM74PC138 3-Line to 8-Line Decoder/Demultiplexer

General Description

The MM74PC138 is fabricated using National's P2CMOS technology, which offers wide operating voltage, low power consumption, high noise immunity, and high speed. The speed of the MM74PC138 compares favorably with the speed of low power Schottky. Function and pinout compatibility with the 74LS138 and the 8205 minimizes design time for those designers already familiar with these two devices.

Three enable inputs are provided (two active low and one active high) to reduce the need for external gates or inverters when expanding a system.

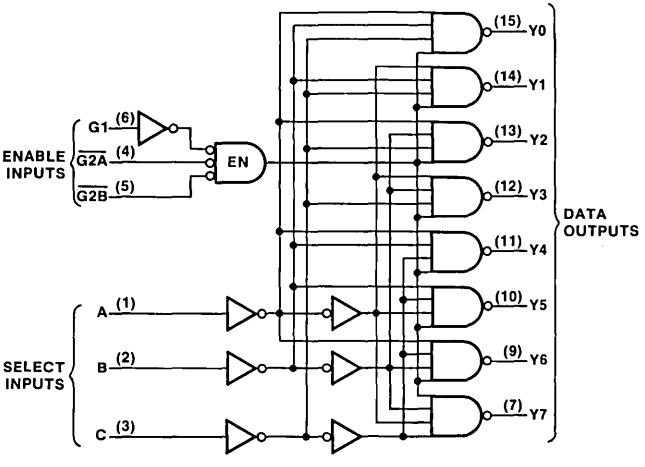
The MM74PC138's simple expansion in decoder and demultiplexer applications is particularly useful in completing NSC800 high-performance, low-power designs, while reducing component count.

When using the device as a demultiplexer, one of the three enable inputs (G1, $\overline{G2A}$, $\overline{G2B}$) serves as the data input terminal while the remaining enable inputs are enabled. The information will then be transmitted to the selected output, as determined by the 3-Line select address (A, B, C).

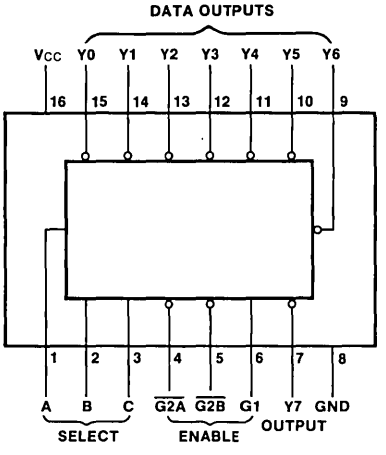
Features

- Simple Expansion - Three Enable Inputs
- P2CMOS Technology
- High Density 16-Pin Package
- Outputs Sink 6mA Minimum
- Single 5V Power Supply,
- High Noise Immunity: 0.45 V_{CC} Typical
- Low Quiescent Power Dissipation
- Full Interface to CMOS Logic Levels
- TTL Drive Capability
  When V_{CC} = 5V

MM74PC138 Decoder/Demultiplexer



Logic Diagram 93-1



Connection Diagram 93-2

Absolute Maximum Ratings

Storage Temperature Range ..... -65° C to 150° C
Voltage at Any Pin with Respect to Ground.....-0.3V to V_{CC} +0.3V
Lead Temperature (Soldering, 10 seconds) .....300° C
Power Dissipation.....500mW

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40° C to +85° C	5V
Commercial	0° C to +70° C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating voltage range components to be announced.

DC Electrical Characteristics

Min/Max limits apply across temperature range, unless otherwise specified.

Symbol	Parameter	V_{CC}	Conditions	Min	Typ	Max	Units
CMOS TO CMOS							
V_{IH}	Input High Voltage	5V	V_O = 0.5V or 4.5V	4.00			V
V_{IL}	Input Low Voltage	5V	V_O = 0.5V or 4.5V			1.00	V
V_{OH}	Output High Voltage	5V	V_I = V_{CC} or GND	4.95			V
V_{OL}	Output Low Voltage	5V	V_I = V_{CC} or GND			0.05	V
I_{OH}	Output High (Source) Current	5V	V_O = 4.6V, V_I = 0V or 5V	-2.0			mA
I_{OL}	Output Low (Sink) Current	5V	V_O = 0.4V, V_I = 0V or 5V	+2.0			mA
I_{CC}	Supply Current	5V	V_I = V_{CC} or GND All Outputs Open			300	μA
I_I	Input Current	5V	V_I = V_{CC} or GND			±1	μA

CMOS TO LSTTL INTERFACE (UNBUFFERED OUTPUT GATES)
V_{CC} = 5V ± 5% unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{I(1)}$	Input High Voltage	V_O = 4.4V or 0.4V I_O = ± 10μA	V_{CC}-1.5			V
$V_{I(0)}$	Input Low Voltage	V_O = 4.4V or 0.4V I_O = ± 10μA			0.8	V
$V_{O(1)}$	Output High Voltage	V_I = 4.0V or 1.0V I_O = ± 2.0mA	2.4			V
$V_{O(0)}$	Output Low Voltage	V_I = 4.0V or 1.0V I_O = ± 2.0mA			0.4	V

DC Electrical Characteristics - (Cont'd.)

OUTPUT DRIVE
V_{CC} = 5V, T_A = -40° C to +85° C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{SOURCE}	Output Source Current (P-Channel)	V_{OUT} + 2.4V	-20			mA
I_{SINK}	Output Sink Current (N-Channel)	V_{OUT} = 0.4V	6			mA

AC Electrical Characteristics

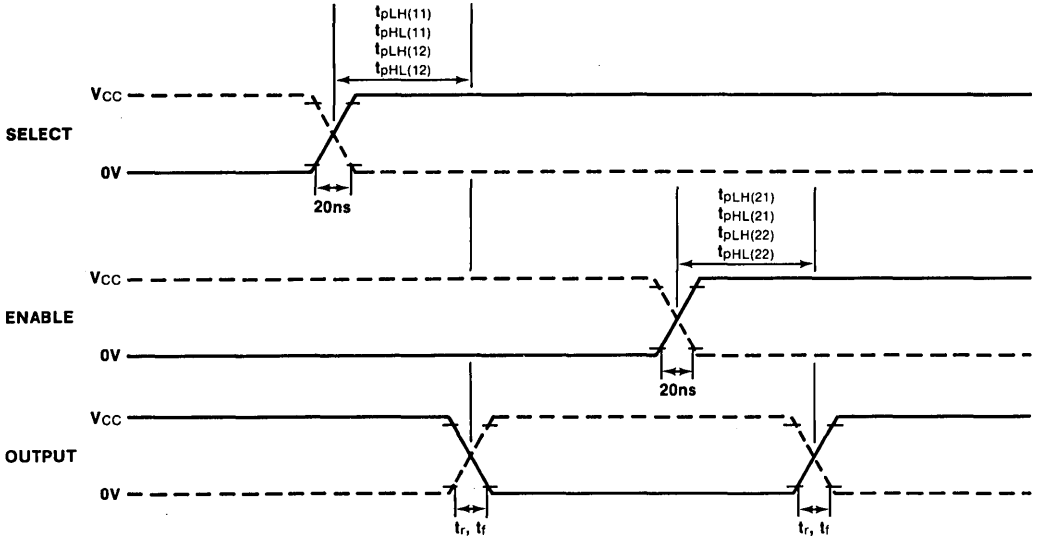
TA = 25° C

Symbol	Parameter	V_{CC}	Conditions	Min	Typ 15pF	Typ 100pF	Max	Units
Select to Output Propagation Delay Time								
$t_{pLH(11)}$	Low-to-High Level Input; Low-to-High Level Output	5V			26	39		ns
$t_{pHL(11)}$	Low-to-High Level Input; High-to-Low Level Output	5V			20	41		ns
$t_{pLH(12)}$	High-to-Low Level Input; Low-to-High Level Output	5V			20	39		ns
$t_{pHL(12)}$	High-to-Low Level Input; High-to-Low Level Output	5V			29	41		ns
Enable to Output,Propagation Delay Time								
$t_{pLH(21)}$	Low-to-High Level Input; Low-to-High Level Output	5V			33	37		ns
$t_{pHL(21)}$	Low-to-High Level Input; High-to-Low Level Output	5V			26	37		ns
$t_{pLH(22)}$	High-to-Low Level Input; Low-to-High Level Output	5V			21	37		ns
$t_{pHL(22)}$	High-to-Low Level Input; High-to-Low Level Output	5V			27	37		ns

Capacitance

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance					
	Select Input Pin			10		pF
	Enable Input Pin			6		pF
C_{OUT}	Output Capacitance					
	Output Pin			25		pF

Timing Waveforms



Switching Time Waveforms

Functional Pin Definitions

INPUT SIGNALS

Enable Inputs (G1, $\overline{G2A}$, $\overline{G2B}$): G1 is active high. $\overline{G2A}$ and $\overline{G2B}$ are both active low. When G1 is high and both $\overline{G2A}$ and $\overline{G2B}$ are low, the data output line selected by the Select Inputs (A, B, C) is low and all other data outputs are high. A logic low on G1 or a high on either $\overline{G2A}$ or $\overline{G2B}$ causes all data outputs to go high regardless of the levels on select inputs.

Select Inputs (A, B, C): The bit configuration, when enabled, selects one of the eight data output lines to a logic low level and causes all other data output lines to go to a logic high level. When enable gate EN (conditioned by G1, $\overline{G2A}$, and $\overline{G2B}$) is disabled, all data output lines are logic high. (See truth table.)

OUTPUT SIGNALS

Data Outputs (Y0-Y7): The eight active low output lines, of which only one can be logic low at a time, are selected by the value of the binary input on the select input pins, if the enable gate EN is enabled. When enable gate EN (conditioned by G1, $\overline{G2A}$, and $\overline{G2B}$) is disabled, all data output lines are high.

Functional Description

Decoder

The MM74PC138 contains a one-of-eight binary decoder. It accepts a three-bit binary code input that activates one of the eight outputs at a time, if the enable gate EN is active. For example, when a binary code of "011" (C = 0, B = A = 1) is present at the select inputs and the device is enabled, an active low signal is present at data output line Y3. All of the other data output lines are therefore high (see truth table).

Enable Gate

To reduce package count the MM74PC138 has an on-chip enable gate with both active high (G1) and active low ($\overline{G2A}$, $\overline{G2B}$) inputs for ease of interface to other system components. The enable gate EN with three enable inputs ANDed together provides one enable signal for the decoder. See logic diagram on page 1.

Truth Table

SELECT			ENABLE			DATA OUTPUTS								EN
			$\overline{G}$ 2	$\overline{G}$ 2	G 1									
C	B	A	A	B		Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
L	L	L	L	L	H	L	H	H	H	H	H	H	H	1
L	L	H	L	L	H	H	L	H	H	H	H	H	H	1
L	H	L	L	L	H	H	H	L	H	H	H	H	H	1
L	H	H	L	L	H	H	H	H	L	H	H	H	H	1
H	L	L	L	L	H	H	H	H	H	L	H	H	H	1
H	L	H	L	L	H	H	H	H	H	H	L	H	H	1
H	H	L	L	L	H	H	H	H	H	H	H	L	H	1
H	H	H	L	L	H	H	H	H	H	H	H	H	L	1
X	X	X	L	L	L	H	H	H	H	H	H	H	H	0
X	X	X	H	L	L	H	H	H	H	H	H	H	H	0
X	X	X	L	H	L	H	H	H	H	H	H	H	H	0
X	X	X	H	H	L	H	H	H	H	H	H	H	H	0
X	X	X	H	L	H	H	H	H	H	H	H	H	H	0
X	X	X	L	H	H	H	H	H	H	H	H	H	H	0
X	X	X	H	H	H	H	H	H	H	H	H	H	H	0

H = high level
L = Low level
X = Don't care

APPLICATIONS

Typical applications include expansion of devices in systems that use input ports, output ports, and memory components. When the MM74PC138 is enabled, one of its eight outputs goes low, thus a single device, or group of devices, is selected (e. g., one row of a memory system). The three chip-enable inputs (two active low and one active high) on the MM74PC138 allow easy system expansion. For very large systems these decoders can be cascaded (see *Figure 1*) so each decoder drives up to eight other decoders for expansion, such as memory systems or input/output subsystems.

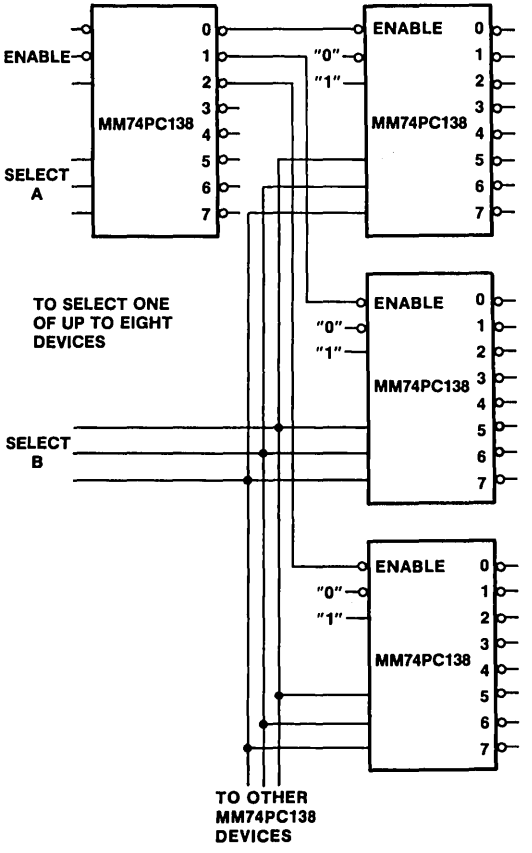


FIGURE 1. Cascaded MM74PC138 Devices 93-4

The circuit shown in *Figure 2* can be used to generate enable signals for I/O ports or memory interface systems. When 1K ROM and/or RAM memory devices are used, a 24K memory system can be constructed. Additional lines are shown for addressing individual memory addresses.

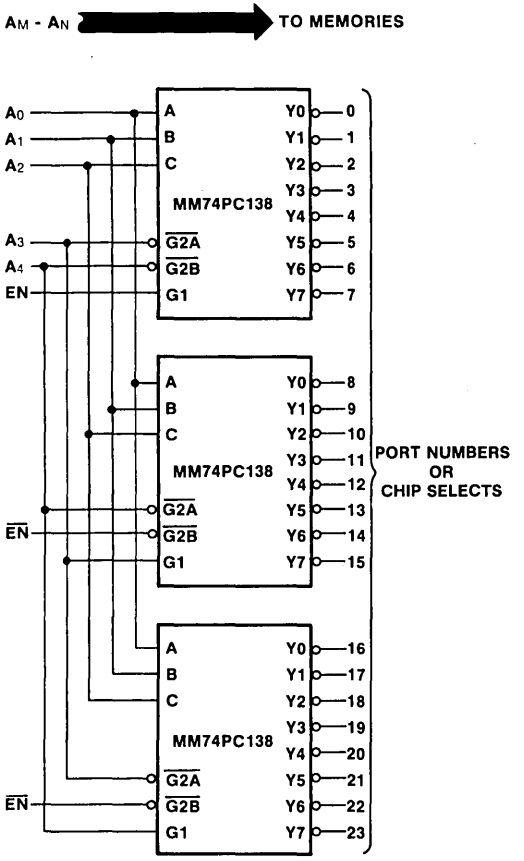


FIGURE 2. I/O Port Decoder or Memory Interface

93-5

A logic sequencer can be designed using the MM74PC138 for decoding and activating the line for the desired state in any sequence or time duration by programming the enable and select inputs (see *Figure 3*).

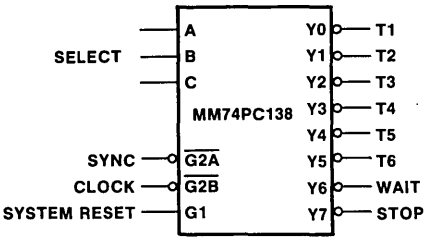


FIGURE 3. Logic Sequencer 93-6

For demultiplexer applications, one of the three enable inputs (G1, $\overline{G2A}$, $\overline{G2B}$) serves as the data input terminal, while the other enable inputs are enabled (see *Figure 4*). The transmitted data is distributed to the selected output as determined by the 3-line select address (A, B, C).

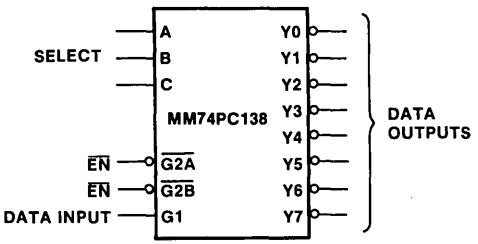
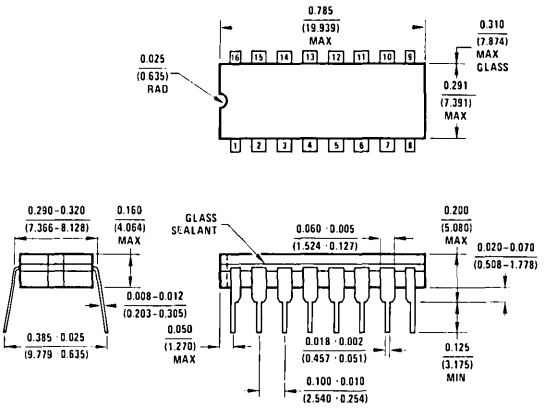


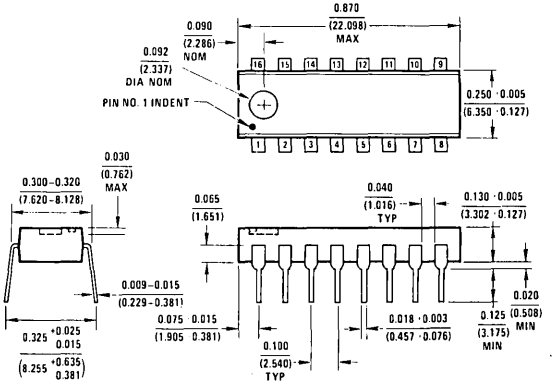
FIGURE 4. Eight-Output Demultiplexer 93-7

Physical Dimensions

inches (millimeters)



Ceramic Dual-in-Line Package [Cerdip (J)]



Plastic Dual-in-Line Package (N)

Ordering Information

MM74PC138XX

- **I** = Industrial Temperature (-40°C to +85°C)
- **N** = Plastic Package
- **J** = Ceramic Dual-in-Line Package [Cerdip (J)]

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Elsenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

MM82PC08 8-Bit Bidirectional Transceiver

General Description

The MM82PC08 is an 8-bit TRI-STATE® high-performance, low-power P²CMOS transceiver. It provides bidirectional drive for bus-oriented microprocessor and digital communications systems. Straight through bidirectional transceivers are featured.

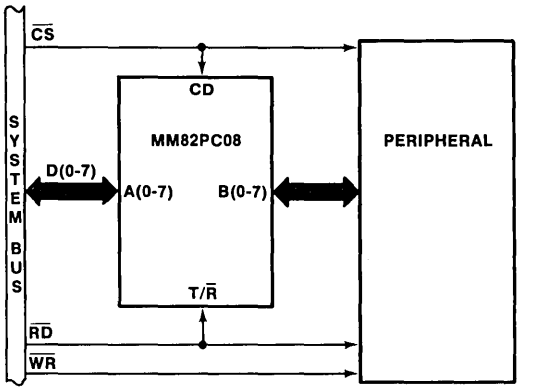
One input, Transmit/Receive, determines the direction of logic signals through the bidirectional transceiver; Transmit specifies data flow from Port A to Port B; Receive specifies data flow from Port B to Port A. The Chip Disable input disables both ports by placing them in the high-impedance state.

The MM82PC08 may be utilized in completing NSC800 high-performance, low-power designs.

Features

- P²CMOS Technology
- 8-Bit Bidirectional Data Flow Reduces System Package Count
- Bidirectional TRI-STATE Inputs/Outputs Interface with Bus-Oriented Systems
- Full Interface to CMOS Logic Levels
- Pinouts Simplify System Interconnections
- Transmit/Receive and Chip Disable Simplify Control Logic
- Compact 20-Pin Dual-In-Line Package
- Low Power
- Both Ports Have 100pF Load Drive Capability
- TTL Drive Capability
  When V_{CC} = 5V

MM82PC08 Basic System Configuration



89-1

Absolute Maximum Ratings

Storage Temperature Range .....-65°C to 150°C
Voltage at Any Pin with Respect to Ground.....-0.3V to V_{CC} +0.3V
Lead Temperature (Soldering, 10 seconds) .....300°C
Power Dissipation.....500mW

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Charactristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40°C to +85°C	5V
Commercial	0°C to +70°C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating voltage range components to be announced.

DC Electrical Characteristics

Min/Max limits apply across temperature range, unless otherwise specified.

Symbol	Parameter	V_{CC}	Conditions	Min	Typ	Max	Units
CMOS TO CMOS							
V_{IH}	Input High Voltage	5V	V_O = 0.5V or 4.5V	4.00			V
V_{IL}	Input Low Voltage	5V	V_O = 0.5V or 4.5V			1.00	V
V_{OH}	Output High Voltage	5V	V_I = V_{CC} or GND	4.95			V
V_{OL}	Output Low Voltage	5V	V_I = V_{CC} or GND			0.05	V
I_{OH}	Output High (Source) Current	5V	V_O = 4.6V, V_I = 0V or 5V	-2.0			mA
I_{OL}	Output Low (Sink) Current	5V	V_O = 0.4V, V_I = 0V or 5V	+2.0			mA
I_{CC}	Supply Current	5V	V_I = V_{CC} or GND All Outputs Open			300	μA
I_I	Input Current	5V	V_I = V_{CC} or GND			±1	μA

CMOS TO LSTTL INTERFACE (BUFFERED OUTPUT GATES)
V_{CC} = 5V ± 5% unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{I(1)}$	Input High Voltage	V_O = 4.4V or 0.4V I_O = ±10μA	V_{CC} -1.5			V
$V_{I(0)}$	Input Low Voltage	V_O = 4.4V or 0.4V I_O = ±10μA			0.8	V
$V_{O(1)}$	Output High Voltage	V_I = $V_{I(1)}$ Min or $V_{I(0)}$ Max I_O = -2.0mA	2.4			V
$V_{O(0)}$	Output Low Voltage	V_I = $V_{I(1)}$ Min or $V_{I(0)}$ Max I_O = +2.0mA			0.4	V

OUTPUT DRIVE
V_{CC} = 5V, T_A = -40°C to +85°C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-16			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	5			mA

AC Electrical Characteristics

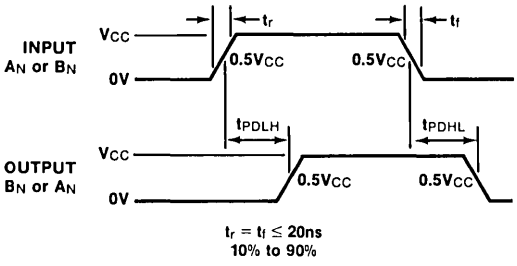
T_A = 25°C

Symbol	Parameter	V_{CC}	Conditions	Min	Typ 15pF	Typ 100pF	Max	Units
Port Data/Mode Specifications								
t_{PDLH}	Propagation Delay to Logical "1" from Port A, B to Port B, A	5V	See *Figure 1*		32	40		ns
t_{PDHL}	Propagation Delay to Logical "0" from Port A, B to Port B, A	5V	See *Figure 1*		32	40		ns
t_{PTRH}	Propagation Delay from High Impedance to Logical "1" from $T/\overline{R}$ to Port	5V	See *Figure 1*		44	55		ns
t_{PTRL}	Propagation Delay from High Impedance to Logical "0" from $T/\overline{R}$ to Port	5V	See *Figure 1*		52	65		ns
t_{PZH}	Propagation Delay from High Impedance to Logical "1" from CD to Port	5V	See *Figure 3* $S_3 = 0$ $S_4 = 1$ R_2 = 5K C_3 = 15pF, 100pF		40	50		ns
t_{PZL}	Propagation Delay from High Impedance to Logical "0" from CD to Port	5V	(See *Figure 3*) $S_3 = 1$ $S_4 = 0$ R_2 = 5K C_3 = 15pF, 100pF		52	65		ns

Symbol	Parameter	V_{CC}	Conditions	Min	Typ	Max	Units
Transmit/Receive Mode Specifications							
t_{PHZTR}	Propagation Delay from Logical "1" to High Impedance from $T/\overline{R}$ to Port	5V	(See *Figure 2*) $S_1 = 0$ $S_2 = 1$		50		ns
t_{PLZTR}	Propagation Delay from Logical "0"to High Impedance from $T/\overline{R}$ to Port	5V	(See *Figure 2*) $S_1 = 1$ $S_2 = 0$		55		ns
t_{PHZ}	Propagation Delay from Logical "1" to High Impedance from CD to Port	5V	(See *Figure 3*) $S_3 = 0$ $S_4 = 1$ R_2 = 1K C_3 = 5pF		50		ns
t_{PLZ}	Propagation Delay from Logical "0" to High Impedance from CD to Port	5V	(See *Figure 3*) $S_3 = 1$ $S_4 = 0$ R_2 = 1K C_3 = 5pF		55		ns

Switching Time Waveforms (CMOS to CMOS)

Port to Port



AC Test Circuit

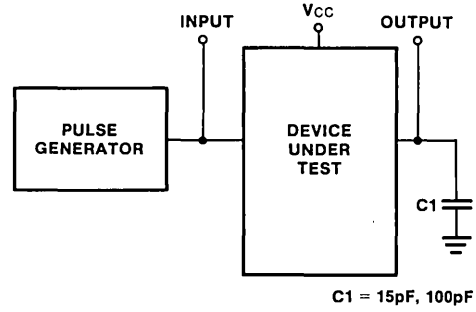
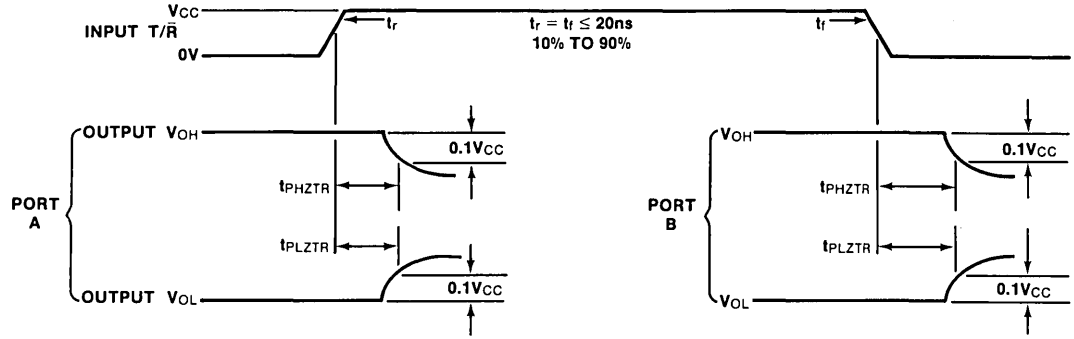


FIGURE 1.

89-2

Propagation Delays

T/$\overline{R}$ to High-Impedance



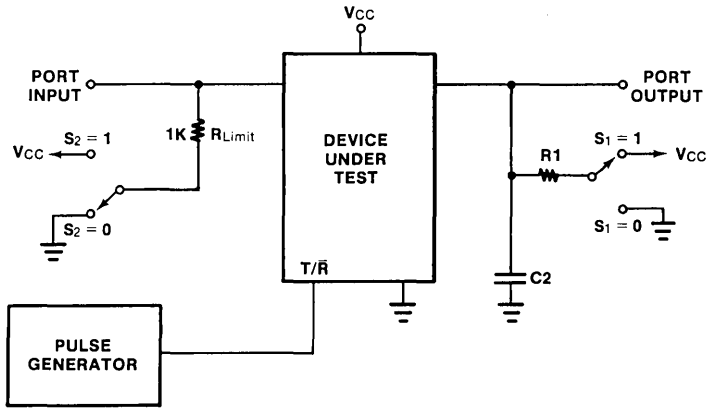
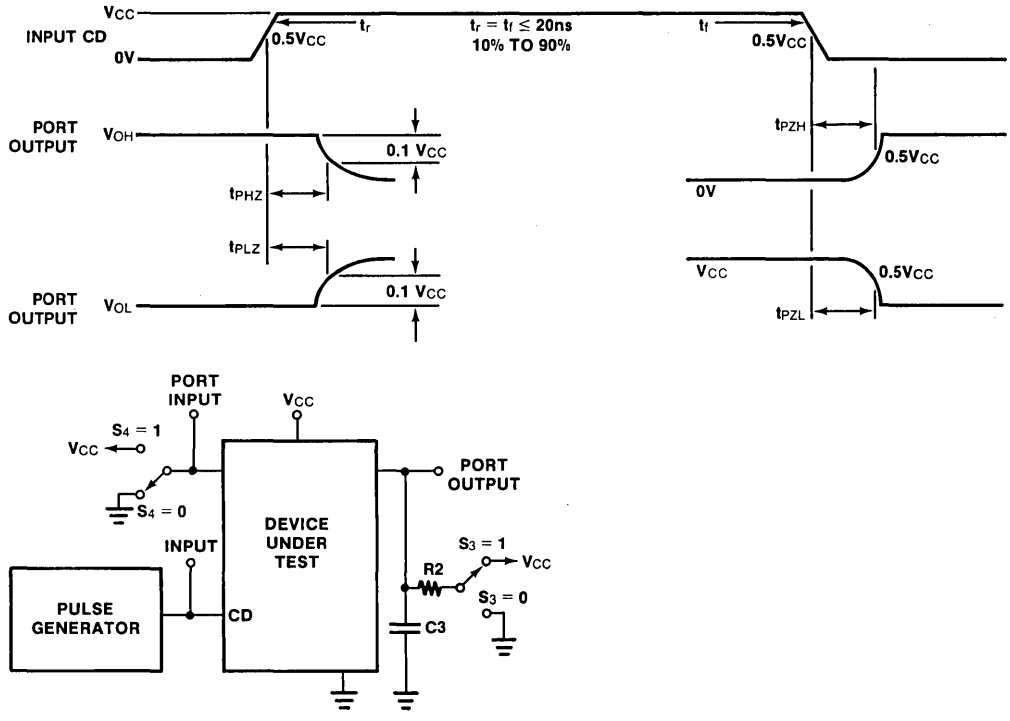


FIGURE 2.

89-3

CD to High-Impedance, CD to Port Output



NOTE: Port input is in a fixed logical condition. See AC Table.

FIGURE 3. 89-4

Functional Pin Descriptions

INPUT SIGNALS

Chip Disable (CD): When CD is high, Port A and Port B are disabled. A low on CD allows data to be transmitted in the direction specified by T/$\overline{R}$.

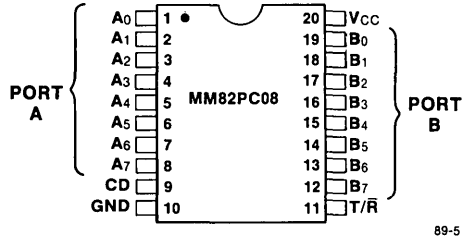
Transmit/Receive (T/$\overline{R}$): When T/$\overline{R}$ is high, Port A is designated as "IN" and Port B is designated as "OUT". When T/$\overline{R}$ is low, the flow is reversed so that Port B is "IN" and Port A is "'OUT".

INPUT/OUTPUT SIGNALS

Port A (A0-A7): Port A is an 8-bit bidirectional port with TRI-STATE outputs for bus-oriented microprocessor and digital communications systems.

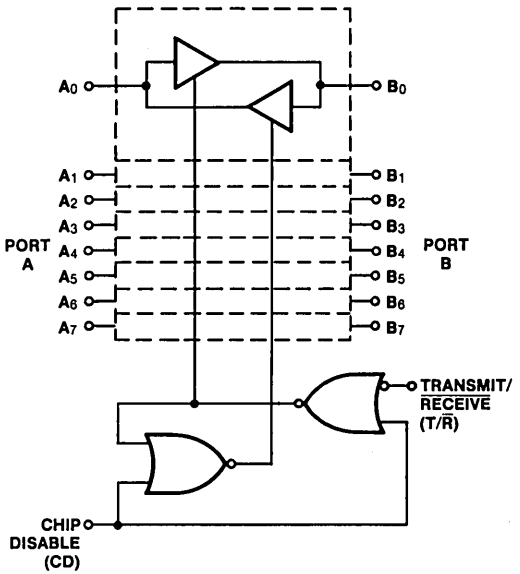
Port B (B0-B7): Port B is identical to Port A including drive capability.

Pin Configuration



89-5

Logic Diagram



Truth Table

INPUTS		RESULTING CONDITIONS	
Chip Disable	Transmit/Receive	PORT A	PORT B
0	0	OUT	IN
0	1	IN	OUT
1	X	High Z	High Z

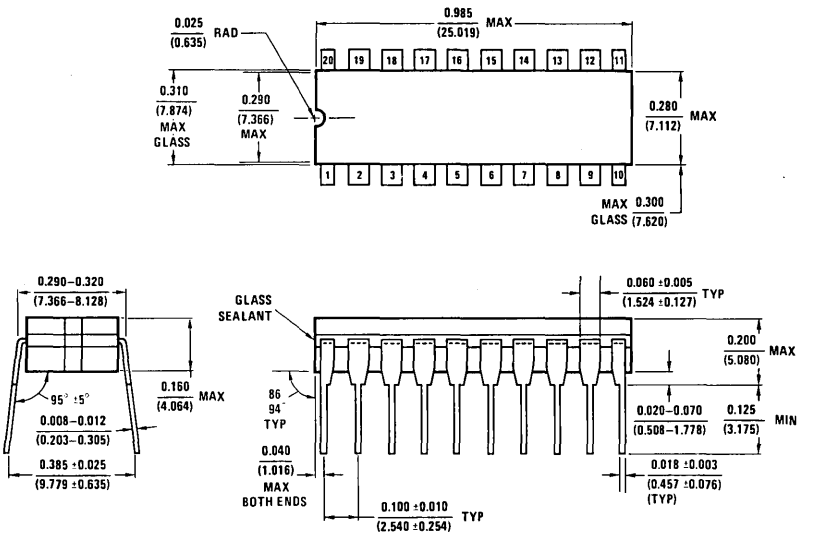
X = don't care

89-6

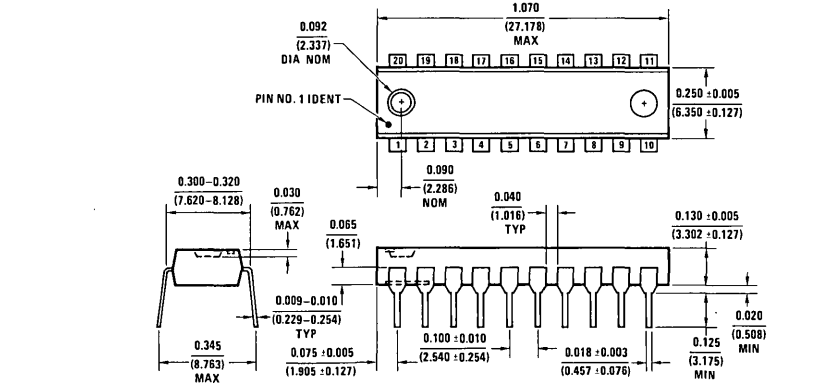
FIGURE 4.

Physical Dimensions

inches (millimeters)



Ceramic Dual-in-Line Package [Cerdip (J)]



Plastic Dual-in-Line Package (N)

Ordering Information

MM82PC08XX

- I = Industrial Temperature (-40°C to +85°C)
- M = Militry Temperature (Availability to be announced)
- N = Plastic Package
- J = Ceramic Dual-in-Line Package [Cerdip (J)]

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor

Preliminary
January 1981

MM82PC12 8-Bit Input/Output Port

General Description

The MM82PC12 is a P²CMOS 8-bit input/output port contained in a standard 24-pin dual-in-line package. The MM82PC12 can be used to implement latches, gated buffers, or multiplexers. Thus, all of the major peripheral and input/output functions of a microcomputer system can be implemented with this device.

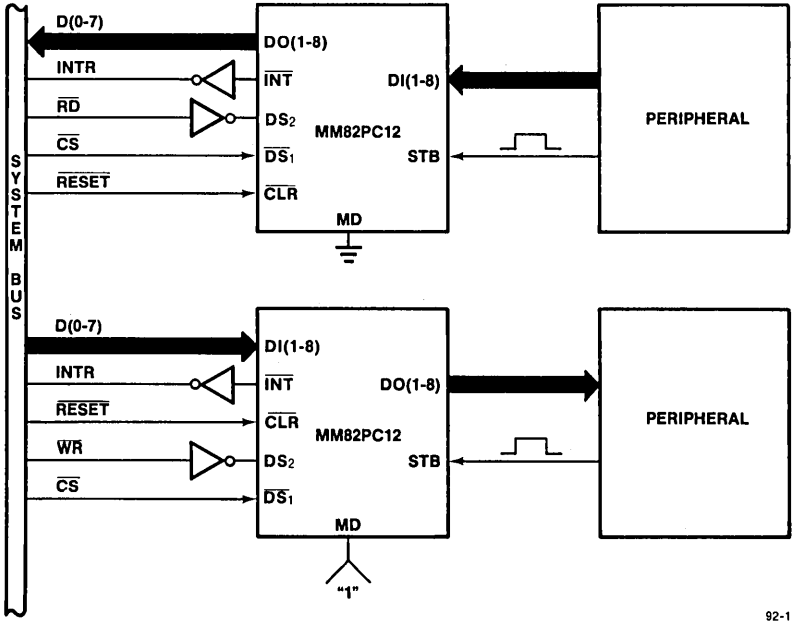
The MM82PC12 includes an 8-bit latch with TRI-STATE® output buffers, and device selection and control logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor.

The MM82PC12 is pinout and function compatible with standard INS8212 and DP8212 devices.

Features

- Drive Capability - 100pF Load
- High Noise Immunity
- Low Power Dissipation
- Full Interface to CMOS Logic Levels
- P²CMOS Technology
- TTL Drive Capability When Vcc = 5V
- 8-Bit Data Latch and Buffer
- Service Request Flip-Flop for Generation and Control of Interrupts
- 1μA Input Load Curent
- Reduces System Package Count by Replacing Buffers, Latches, and Multiplexers in Microcomputer Systems

MM82PC12 System Configuration



92-1

Absolute Maximum Ratings

Storage Temperature Range .....-65°C to 150°C
Voltage at Any Pin with Respect to Ground.....-0.3V to V_{CC} +0.3V
Lead Temperature (Soldering, 10 seconds) .....300°C
Power Dissipation.....500mW

NOTE: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Charactristics.*

Operating Range

Component Type	Ambient Temperature*	V_{CC}**
Industrial	-40°C to +85°C	5V
Commercial	0°C to +70°C	5V

*Availability of Military temperature range components to be announced.
**Availability of extended operating voltage range components to be announced.

DC Electrical Characteristics

Min/Max limits apply across temperature range, unless otherwise specified.

Symbol	Parameter	V_{CC}	Conditions	Min	Typ	Max	Units
CMOS TO CMOS							
V_{IH}	Input High Voltage	5V	V_O = 0.5V or 4.5V	4.00			V
V_{IL}	Input Low Voltage	5V	V_O = 0.5V or 4.5V			1.00	V
V_{OH}	Output High Voltage	5V	V_I = V_{CC} or GND	4.95			V
V_{OL}	Output Low Voltage	5V	V_I = V_{CC} or GND			0.05	V
I_{OH}	Output High (Source) Current	5V	V_O = 4.6V, V_I = 0V or 5V	-2.0			mA
I_{OL}	Output Low (Sink) Current	5V	V_O = 0.4V, V_I = 0V or 5V	+2.0			mA
I_{CC}	Supply Current	5V	V_I = V_{CC} or GND All Outputs Open			300	μA
I_I	Input Current	5V	V_I = V_{CC} or GND			±1	μA

CMOS TO LSTTL INTERFACE (UNBUFFERED OUTPUT GATES)
V_{CC} = 5V ± 5% unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{I\ (1)}$	Input High Voltage	V_O = 4.4V or 0.4V I_O = ±10μA	V_{CC} -1.5			V
$V_{I\ (0)}$	Input Low Voltage	V_O = 4.4V or 0.4V I_O = ±10μA			0.8	V
$V_{O\ (1)}$	Output High Voltage	V_I = 4.0V or 1.0V I_O = ±2.0mA	2.4			V
$V_{O\ (0)}$	Output Low Voltage	V_I = 4.0V or 1.0V I_O = ±2.0mA			0.4	V

OUTPUT DRIVE
V_{CC} = 5V, T_A = -40° C to +85° C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{SOURCE}	Output Source Current (P-channel)	V_{OUT} = 2.4V	-12.5			mA
I_{SINK}	Output Sink Current (N-channel)	V_{OUT} = 0.4V	6.8			mA

AC Electrical Characteristics

T_A = 25° C

Symbol	Parameter	V_{CC}	Conditions	Min	Typ 15pF	Typ 100pF	Max	Units
t_{PW}	Pulse Width	5V	$t_r = t_f$ = 5ns		25	25		ns
t_{PD}	Data to Output Delay	5V			20	25		ns
t_{WE}	Write Enable to Output Delay	5V			37	45		ns
t_{SET}	Data Setup Time	5V			15	15		ns
t_H	Data Hold Time	5V			20	20		ns
t_R	Reset to Output Delay	5V			31	38		ns
t_S	Set to Output Delay	5V			20	28		ns
t_C	Clear to Output Delay	5V			21	28		ns
t_{E0}, t_{D0}	Output Enable/Disable Time	5V			28	28		ns
t_{E1}, t_{D1}	Output Enable/Disable Time	5V			36	36		ns

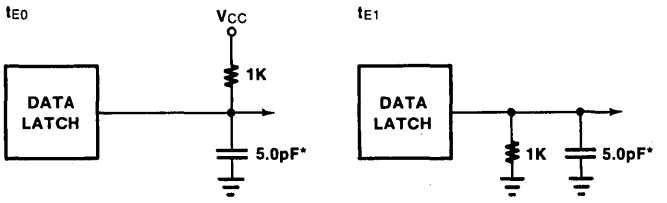
Capacitance*

F = 1MHz, V_{BIAS} = 2.5V, V_{CC} = +5V, T_A = 25° C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{IN}	DS_1, MD, DI_1-DI_8 Input Capacitance			14		pF
C_{IN}	DS_2, $\overline{CLR}$, STB Input Capacitance			16		pF
C_{OUT}	DO_1-DO_8 Output Capacitance			10		pF

*This parameter is sampled and not 100% tested.

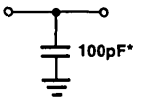
AC Test Circuits



Output Test Circuit for TRI-STATE Delays

*including jig and probe capacitance

92-2



Output Test Circuit for Propagation Delays

*including jig and probe capacitance

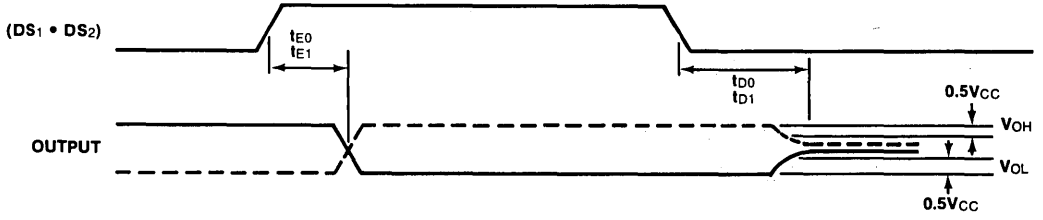
92-3

Switching Time Waveforms

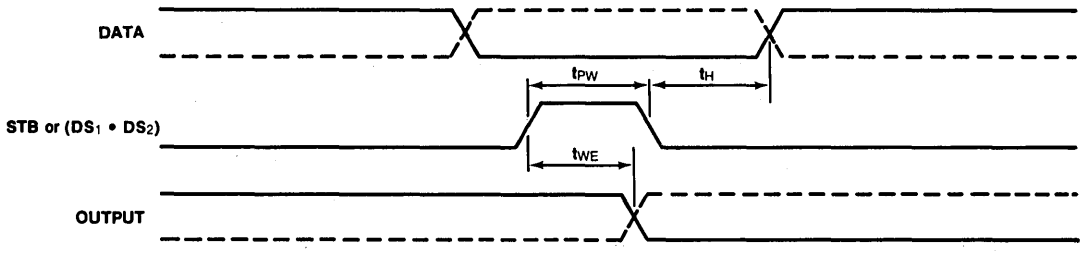
92-4

Timing Waveforms

Read Timing

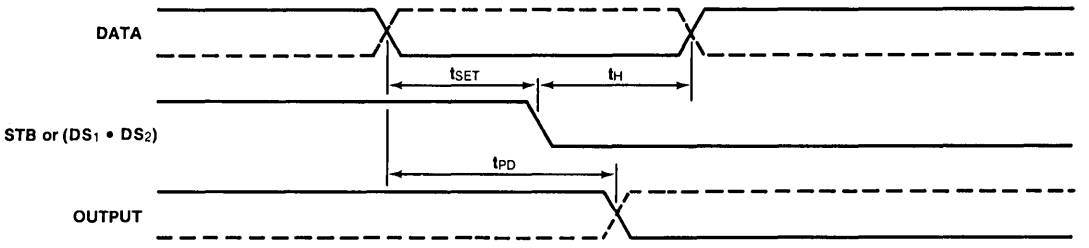


Write Timing

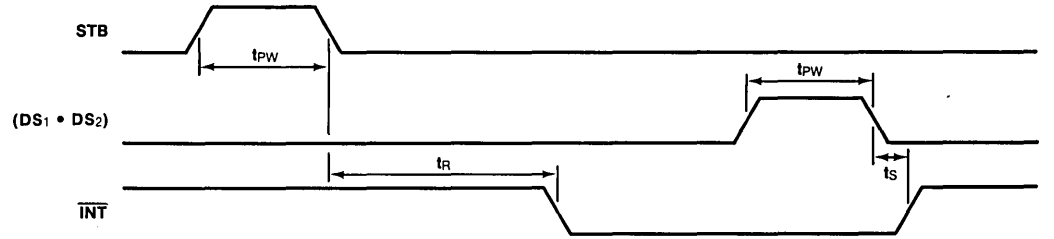


Timing Waveforms (Cont'd.)

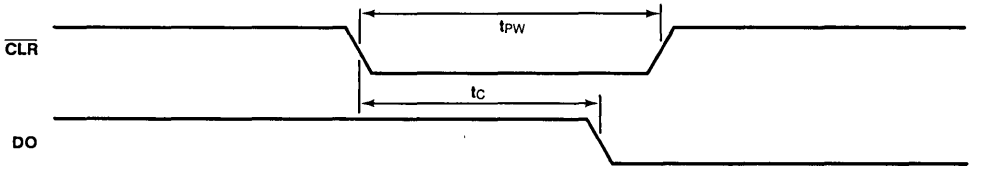
Data Setup, Hold, Propagation Delay Timing



Interrupt Timing

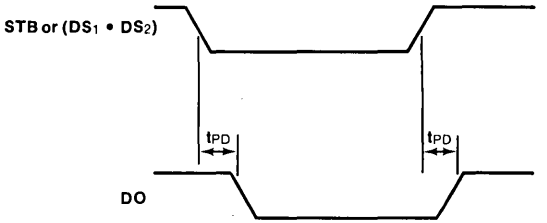


Clear Timing



92-5

Propagation Delays



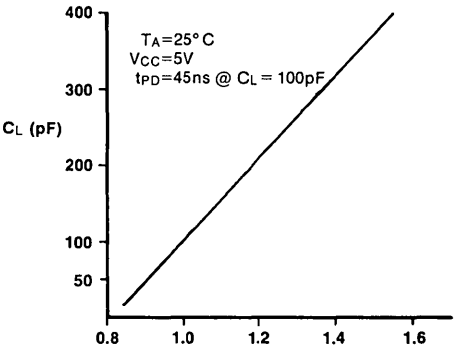


FIGURE 1. Normalized Typical Propagation Delay vs. Load Capacitance

92-6

Figure 1 illustrates the calculations of a more useful propagation delay. The figure uses a 5-volt supply with a tolerance of ±10%, ambient temperature of +25°C, and a load capacitance of 100pF. The AC Characteristics table depicts t_{PD}, at 5 volts, 25°C, equalling 25ns. Use the graph in *Figure 1* to get the degradation multiple for 150pF. The number shown is 1.09. The adjusted propagation delay is, therefore 25 x 1.09 or 27ns.

Functional Pin Descriptions

The following describes the function of all the MM82PC12 input/output pins. Some of these descriptions reference internal circuits.

INPUT SIGNALS

Device Select ($\overline{DS_1}$, DS_2): When $\overline{DS_1}$ is low and DS_2 is high, the device is selected. The output buffers are enabled and the service request flip-flop is asynchronously reset (cleared) when the device is selected.

Mode (MD): When MD is high (output mode), the output buffers are enabled and the source of the data latch clock input is the device selection logic ($DS_1 \bullet DS_2$). When MD is low (input mode), the state of the output buffers is determined by the device selection logic ($DS_1 \bullet DS_2$) and the source of the data latch clock input is the strobe (STB) input.

Strobe (STB): STB is used as the data latch clock input when the mode (MD) input is low (input mode). STB is also used to synchronously set the service request flip-flop, which is negative edge triggered.

Data In (DI_1-DI_8): Data In is the eight-bit data input to the data latch, which consists of eight D-type flip-flops, incorporating a level sensitive clock. While the data latch clock input is high, the Q output of each flip-flop follows the data input. When the clock input returns low, the data latch stores the data input. Clear ($\overline{CLR}$) is only effective when the clock is low (latch in the latched state).

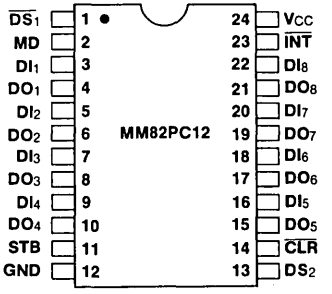
Clear ($\overline{CLR}$): When $\overline{CLR}$ is low, the data latch is reset (cleared) if the clock is also low. The clock input high overrides the clear ($\overline{CLR}$) input data latch reset. $\overline{CLR}$ being low also resets the service request flip-flop. The service request flip-flop is in the non-interrupting state when reset.

OUTPUT SIGNALS

Interrupt ($\overline{INT}$): The interrupt pin goes low (interrupting state) when either the service request flip-flop is synchronously set by the strobe (STB) input or the device is selected.

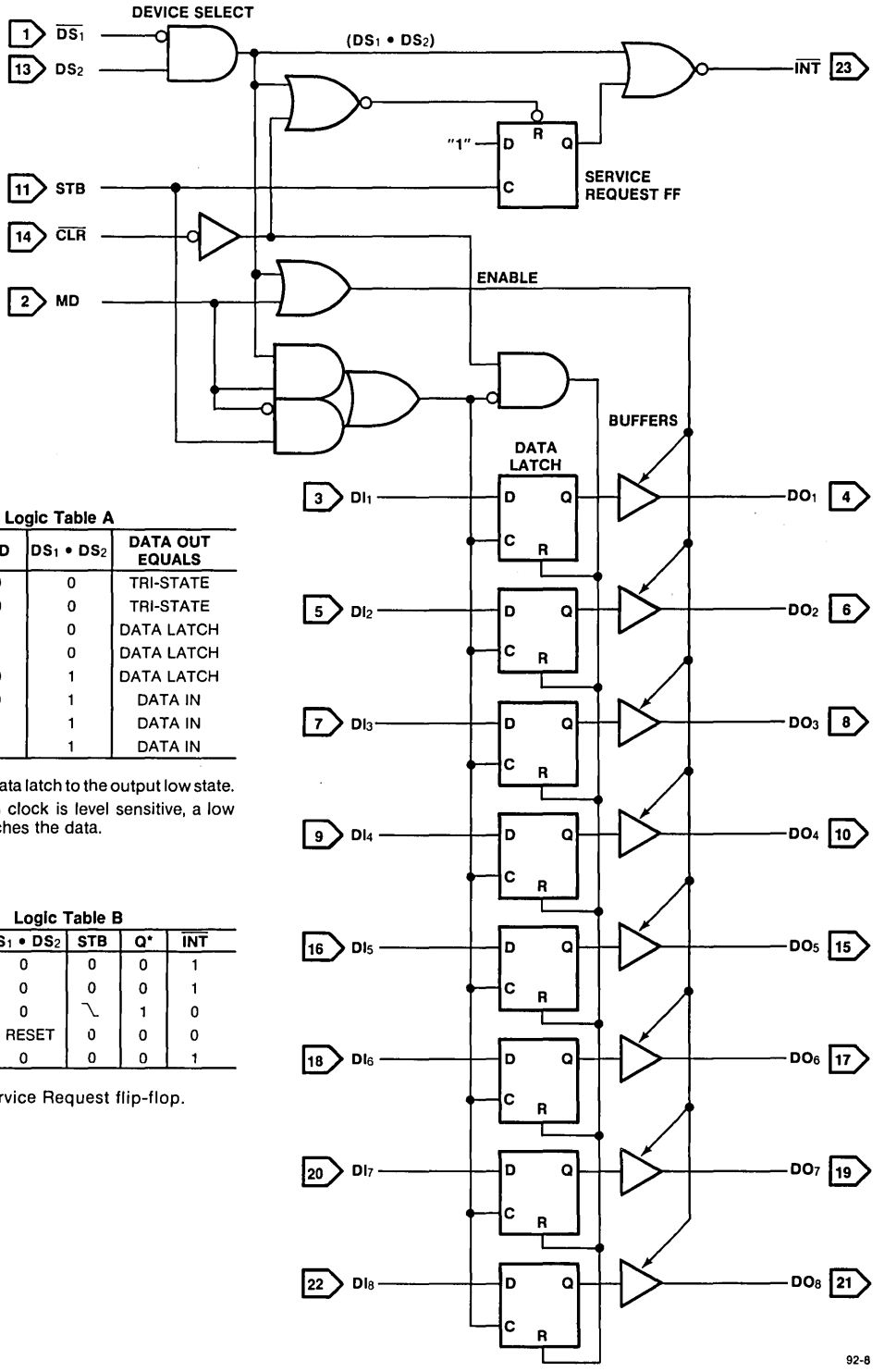
Data Out (DO_1-DO_8): Data Out is the eight-bit data output of data buffers, which are TRI-STATE, non-inverting stages. These buffers have a common control line that either enables the buffers to transmit the data from the data latch outputs or disables the buffers by placing them in the high-impedance state.

Pin Configuration



92-7

Logic Diagram



Logic Table A

STB	MD	$DS_1 \bullet DS_2$	DATA OUT EQUALS
0	0	0	TRI-STATE
1	0	0	TRI-STATE
0	1	0	DATA LATCH
1	1	0	DATA LATCH
0	0	1	DATA LATCH
1	0	1	DATA IN
0	1	1	DATA IN
1	1	1	DATA IN

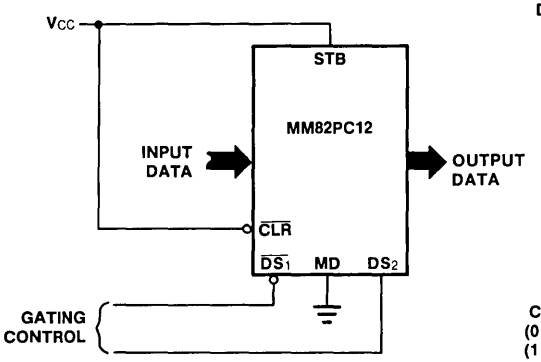
$\overline{CLR}$ resets data latch to the output low state.
The data latch clock is level sensitive, a low level clock latches the data.

Logic Table B

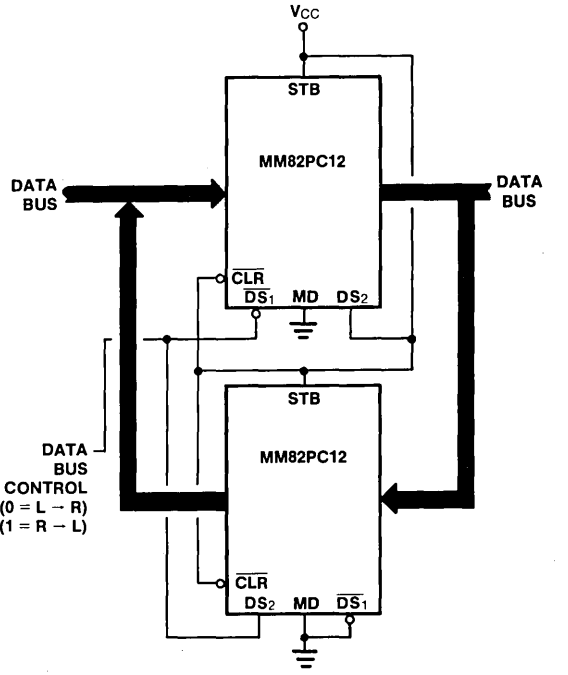
$\overline{CLR}$	$DS_1 \bullet DS_2$	STB	Q*	$\overline{INT}$
0 RESET	0	0	0	1
1	0	0	0	1
1	0	↘	1	0
1	1 RESET	0	0	0
1	0	0	0	1

*Internal Service Request flip-flop.

Applications in Microcomputer Systems

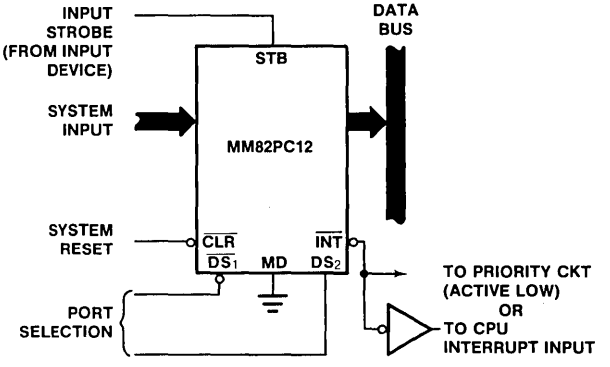


Gated Buffer (TRI-STATE)

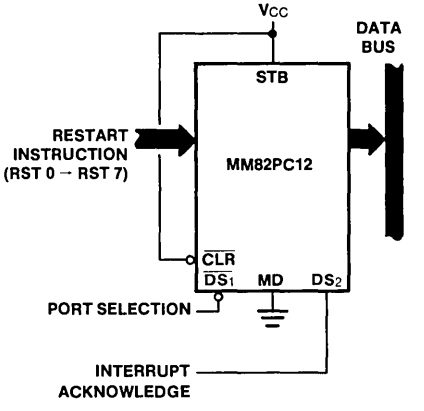


Bidirectional Bus Driver

92-9



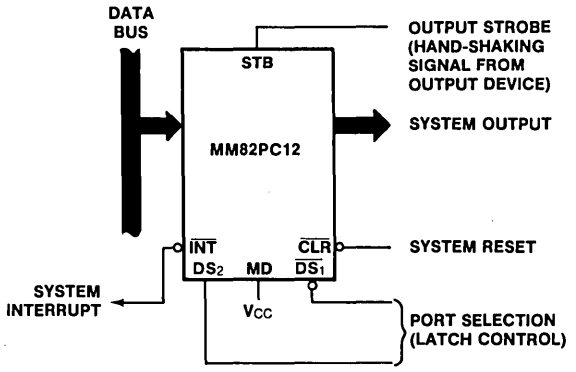
Interrupting Input Port



Interrupt Instruction Port

92-10

Applications in Microcomputer Systems (Cont'd.)

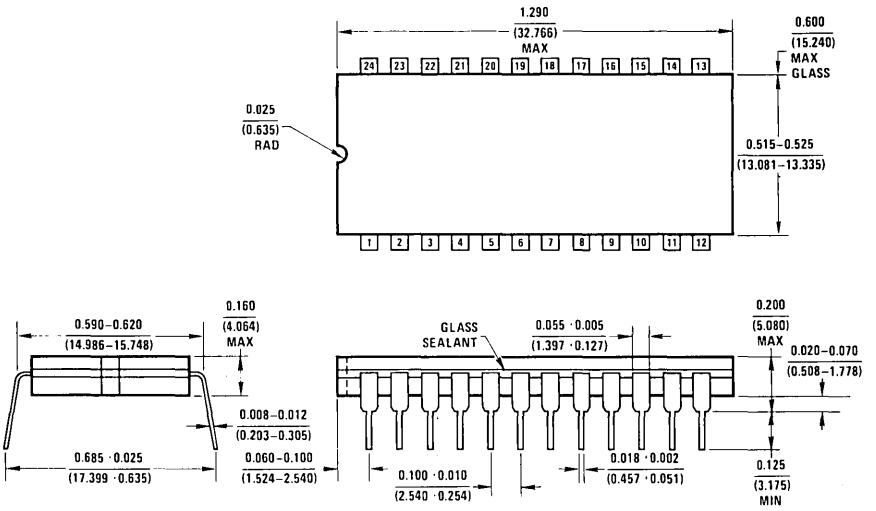


Output Port (with Hand-Shaking)

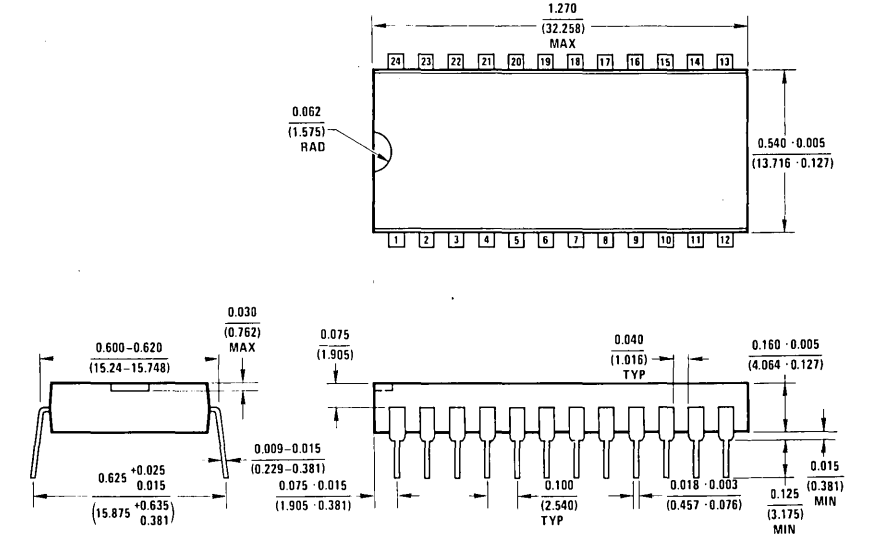
92-11

Physical Dimensions

inches (millimeters)



Ceramic Dual-in-Line Package [Cerdip (J)]



Plastic Dual-in-Line Package (N)

Ordering Information

MM82PC12XX

- I Industrial Temperature (-40°C to +85°C)
- M Military Temperature (Availability to be announced)

- N Plastic Package
- J Ceramic Dual-in-Line Package [Cerdip (J)]

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya. Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor.
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

Comparison Study NSC800 vs. 8085, Z80®

National Semiconductor
Application Brief
Keith Winter
January 1981

Introduction

The NSC800 is an 8-bit parallel processor with a Z80 compatible instruction set manufactured using National's P²CMOS process. This process combines the speed of silicon gate NMOS with the low power inherent to CMOS.

The NSC800 has a 16-bit address bus which consists of the upper eight address bits (A8-A15) and the lower eight address bits (AD0-AD7). Address bits A0-A7 are time multiplexed on the 8-bit bidirectional address/data bus (AD0-AD7).

There are several advantages to using a multiplexed address/data bus. Multiplexing frees pins on the CPU and peripheral packages for other purposes, such as status outputs, DMA control lines, and multiple interrupts. This can reduce system component count. Fewer bus signal lines are required for device interconnections in most applications (16 lines for multiplexed bus systems vs. 24 lines for non-multiplexed systems). This reduces PC board complexity.

For small systems which utilize other members of the NSC800 family (NSC810 RAM-I/O-TIMER, NSC830 ROM-I/O, NMC6504 4K x 1 Static RAM , NMC6514 1K x 4 Static RAM , no extra demultiplexing is necessary as they have internal address latches which perform the required demultiplexing. In addition to the above parts, a complete family of low power speed compatible logic and interface parts is also available. See list of NSC800 devices on page three.

NSC800 vs. 8085

In terms of bus structure, the NSC800 is similar to the 8085. Both processors utilize a multiplexed bus and timing relationships are approximately the same. The 8085 does not guarantee that output data on AD0-AD7 are valid on both the leading and trailing edges of $\overline{WR}$. For the NSC800, data are valid on both the leading and trailing edges of $\overline{WR}$.

Both the NSC800 and the 8085 use ALE, S0, S1, and IO/$\overline{M}$ to indicate status. The lower eight address bits are guaranteed to be valid on the data bus at the trailing edge (high to low transition) of ALE (Address Latch Enable). This signal is used by the external system components to separate the address and data buses. When the only components utilized in the system are members of the NSC800 family (which contain on-chip demultiplexors), ALE needs only to be connected to the enable inputs. If non-NSC800 family components are used, ALE can be used to enable an 8-bit latch to perform the function of bus separation.

Decoding status bits S0 and S1, in conjunction with IO/$\overline{M}$, notifies the external system of the type of the ensuing M cycle. Table 1 shows a truth table of the encoded information. During a halt status the NSC800 will continue to refresh dynamic RAM.

Table 1.

Machine Cycle Status - NSC800 and 8085

S0	S1	IO/M	STATUS
1	0	0	Memory Write
0	1	0	Memory Read
1	0	1	I/O Write
0	1	1	I/O Read
1	1	0	Opcode Fetch
0	1	0	Bus Idle*
0	0	0	Halt

*ALE not suppressed during Bus Idle

Direct Memory Access (DMA) control signals $\overline{BREQ}$ and $\overline{BACK}$ of the NSC800 perform the same functions as HOLD and HLDA on the 8085. The NSC 800 allows simple wire ORing by using active low states for the DMA control signals. An active low on the $\overline{BREQ}$ (Bus Request) line, tested during the last T state of the current M cycle, initiates a DMA condition. The NSC800 will then respond with an active low $\overline{BACK}$ (Bus Acknowledge) signal causing the address, data and control buses (TRI-STATE® circuits) to go to the high impedance state, and notifies the interrupting device that the system bus is available for use. There is a difference in the timing relationship between these functions for the two processors. The 8085 responds with HLDA, one-half T state after it recognizes HOLD. The NSC800 responds with $\overline{BACK}$, one T state after it recognizes $\overline{BREQ}$.

During Input/Output cycles for peripherals, the NSC800 automatically inserts one wait state. This reduces the external hardware required for slow peripherals. The 8085 does not insert its own wait state during these I/O cycles. When they are needed, the 8085 user must design his system to contain the additional hardware required to do the wait state insertion. When more than one wait state is required, additional wait states can be added to the I/O cycles in a similar way on both the NSC800 and the 8085. On the NSC800, this is accomplished by bringing the $\overline{WAIT}$ control signal active low during T2 of an I/O or memory cycle. The 8085 is controlled in the same way through the use of the READY line.

The NSC800 instruction set is Z80 compatible and more powerful than the 8085's. The NSC800 does not support the RIM and SIM instructions of the 8085 (RIM and SIM can be emulated with I/O instructions), but has an improved instruction set for enhanced system performance. The NSC800 has two functions, $\overline{RFSH}$ and PS, instead of the two serial I/O lines SOD and SID. $\overline{RFSH}$ (Refresh) is a status signal which indicates that an eight bit refresh address is present on the address/data bus (AD0-AD7). The refresh address occurs during T3 of each M1 (opcode fetch) cycle. The internal refresh counter is

®Z80 is a registered trademark of Zilog, Inc.

incremented after each instruction cycle. This counter output can be employed by the user's dynamic RAM refresh circuits. The $\overline{PS}$ (Power Save) control input, when active, causes the CPU to stop all internal clocks at the end of the current instruction, which reduces power consumption. The on-chip oscillator and CLK remain active for any required external timing. The NSC800 leaves all buses unchanged during this time, which has the effect of reducing power consumption on other CMOS parts in the system since the buses are not changing states. All internal registers and status conditions are maintained, and when $\overline{PS}$ subsequently goes high, the opcode fetch cycle begins in a normal fashion.

Table 2 indicates the major differences between the NSC800 and the 8085 presented in tabular form for quick reference.

Table 2.

NSC800 vs. 8085 Comparison

Item	NSC800	8085
Power Consumption	50mW @ 5V	850mW @ 5V
Bus Drive Capacity	1 std. TTL (100 pF)	1 std. TTL (100 pF)
Dynamic RAM Refresh Counter	Yes, 8-bit	No
Automatic WAIT State on I/O	Yes	No
Number of instruction types	158	80
Number of Programmer Accessible Registers	22	10
Block I/O and Search	Yes	No
Minimum System (CPU, 2K ROM, 128 RAM, I/O) Power Required (Total)	(50+25+25) mW 100 mW @ 5V	(850+900+900) mW 2650 mW @ 5V

NSC800 vs. Z80

The NSC800 contains the same complement of internal registers as the Z80 and maintains instruction set and op-code compatibility.

Machine cycle timing for the standard speed version of the NSC800 compares directly with the Z80. The high speed version, the NSC800A, compares with the Z80A. Although the software execution speeds are comparable, the NSC800 offers architectural advantages.

The bus structures of the NSC800 and the Z80 are quite different. The NSC800 uses a multiplexed address/data bus. The Z80 has separate address and data buses. As stated earlier, the separate bus structure requires additional signal lines for interconnection and gives up some package pins which could be used for other purposes.

The main differences between the NSC800 and the Z80, in addition to the bus structures, are the refresh counter, on-chip clock generation, and the interrupt capability.

1. The NSC800 contains an 8-bit refresh counter as opposed to a 7-bit refresh counter in the Z80. (This enables refresh of a 64K dynamic RAM system memory.) The refresh timing of the NSC800 is functionally identical to that of the Z80.

2. The on-chip clock generation reduces the system component count. In place of an external clock generator chip, the NSC800 needs only a crystal or RC circuit to produce the system clock.

3. The NSC800 provides three interrupts that are not available on the Z80: $\overline{RSTA}$, $\overline{RSTB}$, $\overline{RSTC}$. This gives the NSC800 five levels of vectored, prioritized interrupts with no external logic. The general purpose interrupt ($\overline{INTR}$) and Non-maskable Interrupt ($\overline{NMI}$) are identical to the Z80. $\overline{INTR}$ has the same three modes of operation in both processors: Modes 0, 1, and 2. Upon initialization, the NSC800 is in mode 0 to maintain 8080 code compatibility. $\overline{NMI}$, when active, causes a restart to location X'66 as is the case with the Z80. Being a non-maskable interrupt, $\overline{NMI}$ cannot be disabled. The

additional interrupts $\overline{RSTA}$, $\overline{RSTB}$, and $\overline{RSTC}$ cause restarts to locations X'3C, X'34, and X'2C respectively. The priority levels of the five interrupts are: $\overline{NMI}$ (highest), $\overline{RSTA}$, $\overline{RSTB}$, $\overline{RSTC}$, and $\overline{INTR}$ (lowest). For the NSC800, Interrupt acknowledge ($\overline{INTA}$) is provided on a dedicated output pin and need not be decoded externally, as is the case with the Z80.

With the status outputs (S0, S1, IO/$\overline{M}$), early read/write information is obtainable. This is impossible to derive from the Z80.

Refer to *Table 3* for comparison of the major differences between the NSC800 and the Z80.

Table 3.

NSC800 vs. Z80 Comparison

Item	NSC800	Z80
Power Consumption	50mW @ 5V	750mW @ 5V
Instruction Execution (Minimum)	1 μs	1 μs
On-Chip Clock Generator	Yes	No
Number of On-Chip Vectored Interrupts	5	2
Early Read/Write Status	Yes	No
Dynamic RAM Refresh Counter	Yes, 8-bit	Yes, 7-bit

NSC800 Family Devices (P^2CMOS)

MM74PC00 Quad 2-Input NAND gate
MM74PC02 Quad 2-Input NOR gate
MM74PC04 Hex inverter
MM74PC08 Quad 2-Input AND gate
MM74PC32 Quad 2-Input OR gate
MM74PC74 Dual D flip-flop
MM74PC138 3-Line to 8-Line Decoder/Demultiplexer
MM82PC08 8-Bit Bidirectional Transceiver
MM82PC12 Input/Output Port

NOTE: *The above devices are pin for pin and function compatible with the standard TTL, CMOS or NMOS versions currently available.*

SUMMARY

National's NSC800 has a Z80 compatible instruction set, which is more powerful than the 8085. NSC800 external hardware requirements are less because of on-chip automatic wait state insertion, clock generation and five levels of vectored prioritized interrupts.

The 8085 and the NSC800 have similar bus structures, and timing. The key advantages of the NSC800 over the 8085 are the larger instruction set, more registers accessible to programmers, low power consumption, and a dynamic RAM refresh counter.

The main advantages of the NSC800 compared to the Z80 are the multiplexed address/data bus, an 8-bit refresh counter for dynamic RAMs, on-chip clock generation, and five interrupts. The speed of the NSC800 and Z80 is the same but, the NSC800 has very low power consumption.

REFERENCE

Series 800 Microprocessor Family Handbook
National Semiconductor Corporation, Santa Clara, CA.

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Elsenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya. Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor.
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

Software Comparison NSC800 vs. 8085, Z80®

National Semiconductor
Application Brief
Keith Winter
January 1981

Introduction

The NSC800 is an 8-bit parallel microprocessor fabricated Using National's P²CMOS process. This process allows fabrication of a microprocessor family that has the performance of silicon gate NMOS along with the low power inherent to CMOS. The NSC800 instruction set is a superset of the 8080's instruction set. It comprises 696 operation codes falling into 158 instruction types. The instruction categories are:

- Load and Exchange
- Arithmetic and Logic
- Rotate and Shift
- Jump and Call
- Input/Output
- Bit manipulation (set, test, reset)
- Block Transfer and Search
- CPU control

The load instructions allow the movement of data into and out of the CPU, between internal registers, plus the capability to load immediate data into internal registers. The exchange instructions allow swapping of data between two registers.

The arithmetic and logic instructions operate on the data in the accumulator (primary working register) and in the other registers. Status flags are set or reset depending on the result of the particular operation executed. This group includes 8-bit and 16-bit operations.

The rotate and shift instructions allow any register or memory location to be rotated or shifted, left or right, with or without carry. These can be either an arithmetic or logic type.

The jump and call group includes several different types: one byte calls, two byte relative jumps, conditional branching, and three byte calls and jumps, which can reach any location in memory. Calls push the current contents of the Program Counter onto the stack before branching to the new program address to facilitate subroutine execution.

Input/Output instructions allow communication between the NSC800 and external peripheral devices. There are 255 (location X'BB is used for an interrupt mask) unique peripheral I/O locations available to the NSC800. I/O instructions can move data between any memory location or internal register and any I/O location. There are also block I/O instructions which allow moving data blocks of up to 256 bytes directly from memory to any peripheral location or from any peripheral location to a block of memory.

Bit manipulation instructions can set, test or reset any bit in the accumulator, any general purpose register or any memory location.

The block transfer instructions allow a single instruction to move any size block of memory to any other location in memory. Through the use of the block search instructions, any size block of memory can be searched for a particular byte of data.

Finally, the CPU control group allows user control over the various modes of CPU operation, such as enabling and disabling interrupts or setting modes of interrupt response.

The following sections will compare the instruction set of the NSC800 with those of the 8085 and the Z80.

NSC800 vs. 8085

The 8085 instruction set consists of 246 op codes falling into 80 instruction types. With the exception of RIM and SIM, the NSC800 is instruction and op code compatible with the 8085. The RIM and SIM instructions are not supported because the NSC800 does not have the SID and SOD serial I/O lines. The interrupt mask on the NSC800 is accessable by writing the mask word to I/O location X'BB. The bit positions for the interrupt enables are shown below:

Location X'BB Bit Assignments

Bit	Interrupt Enable for
7	N/A
6	N/A
5	N/A
4	N/A
3	$\overline{\text{RSTA}}$
2	$\overline{\text{RSTB}}$
1	$\overline{\text{RSTC}}$
0	$\overline{\text{INTR}}$

N/A = not used: a don't care bit

*Z80 is a registered trademark of Zilog, Inc.

As an example, to enable interrupts on the $\overline{RSTA}$ input, a logic '1' is written into bit 3 of I/O location X'BB. If the master interrupt enable has been set by executing the Enable Interrupt (EI) instruction, interrupts will now be accepted on $\overline{RSTA}$ only.

Other than the method of enabling and disabling individual interrupts and the RIM and SIM instructions themselves, the NSC800 instruction set is a superset of the 8085's instruction set.

The following benchmark demonstrates the code reduction and throughput improvement obtained by using one of the special NSC800 instructions over the same function implemented with the limited 8085 instruction set. The function is to move a 512-byte block of data from one section of memory to another. The CPUs used for this demonstration are the NSC800A (250 ns cycle) and the 8085A-2 (200 ns cycle).

8085

BYTES		MNEMONICS		CYCLES
3		LXI	H,SOURCE	10
3		LXI	D,DEST	10
3		LXI	B,COUNT	10
1	LOOP:	MOV	A,M	7
1		STAX	D	7
1		INX	H	6
1		INX	D	6
1		DCX	B	6
1		MOV	A,C	4
1		ORA	B	4
3		JNZ	LOOP	10
Total: 19				Total: 80

NSC800

BYTES	MNEMONICS		CYCLES
3	LD	HL, SOURCE	10
3	LD	DE, DEST	10
3	LD	BC, COUNT	10
2	LDIR		21
Total: 11			Total: 51

The use of the LDIR instruction of the NSC800 results in a 47.5% increase in throughput and a 42% decrease in the number of bytes required to implement the function when compared with the 8085 implementation. The time required to make the move is approximately 2.69 ms for the NSC800 and approximately 5.12 ms for the 8085. Note that even though the 8085 runs at a faster cycle time (200 ns vs. 250 ns), the improved instruction set of the NSC800 produces an increase in system performance.

The NSC800 includes all 8085 flags plus some additional flags. The flag formats for the NSC800 and 8085 are:

NSC800 Flags (Z80 Flags)

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

8085 Flags

7	6	5	4	3	2	1	0
S	Z	X	AC	X	P	X	CY

The differences between the flag registers on the NSC800 and the 8085 are identified below:

1. Bit position D1 (additional on the NSC800) contains an add/subtract flag that is used internally for proper operation of BCD instructions.

2. In the NSC800, the P/V flag will not match the 8085's P flag after an 8-bit arithmetic operation, since it acts as an overflow bit for the NSC800, but acts as a parity bit for these operations in the 8085.

3. Bit position D2 (changed for the NSC800) is a dual purpose flag; it indicates the parity of the result in the accumulator when logical operations are performed and also represents overflow when signed two's complement arithmetic operations are performed. An overflow occurs when the result of a two's complement operation within the accumulator is out of range.

4. For general Compare operations, the NSC800 uses the P/V flag as an overflow bit, while the 8085 uses the P flag for parity.

5. The H flag (bit position D4) on the NSC800 is functionally the same as the auxiliary carry on the 8085.

6. For Double Precision Addition, the NSC800 leaves the H flag undefined, while the 8085 does not affect the AC flag for this operation (DAD).

7. For Rotate operations, the NSC800 resets the H flag, while the 8085 leaves the AC flag unaffected for these operations.

8. When Complementing the Accumulator, the NSC800 sets the H flag (H = 1), while the 8085 leaves the AC flag unaffected.

9. When Complementing Carry, the NSC800 leaves the H flag undefined, while the 8085 leaves the AC flag unaffected.

10. When Setting the Carry, the NSC800 clears the H flag (H = 0), while the 8085 leaves the AC flag unaffected.

NSC800 vs. Z80

The instruction set and op codes of the NSC800 are identical to those of the Z80. Software written for the Z80 will run on the NSC800 without change, unless I/O location X'BB is used. Another location should be assigned since location X'BB is an on-chip write-only register used for the interrupt mask. Since the NSC800 executes code at the same cycle time as the Z80, any software timing loops will also remain the same, and no change is necessary. The NSC800 expanded interrupt capability is transparent to the user unless specifically evoked by the user software.

The NSC800 has 8-bit refresh rather than the 7-bit refresh scheme of the Z80. Therefore, the state of the 8th bit will be indeterminate since it is part of the R Register and so included in refresh operations.

The status flags on the NSC800 are identical to those on the Z80. There is no difference between the positions of the individual bits in the flag register, nor in the manner in which the flags are set or reset due to an arithmetic or logical operation. Testing of the flags is also the same.

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408)737-5000
TWX: (910)339-9240

National Semiconductor GmbH
Eisenheimerstrasse 61/11
8000 Munchen 21
West Germany
Tel: (089)576091
Telex: 05-22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-ku 160
Tokyo, Japan
Tel: (03)355-3711
TWX: 232-2015 NSCJ—J

National Semiconductor (Hong Kong) Ltd.
8th Floor,
Cheung Kong Electronic Bldg
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex:1121008 Cabine Sao Paulo

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel: (408) 737-5000
TWX: (910) 339-9240

National Semiconductor
District Sales Office
345 Wilson Avenue, Suite 404
Downsview, Ontario M3H 5W1
Canada
Tel: (416) 635-7260

Mexicana de Electronica
Industrial S. A.
Tiacoquemecatl No. 139-401
Esquina Adolfo Prieto
Mexico 12, D.F.
Tel: 575-78-68, 575-79-24

NS Electronics Do Brasil
Avada Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
Sao Paulo, Brasil
Telex: 1121008 CABINE SAO PAULO

National Semiconductor GmbH
Elsenheimerstrasse 61/2
8000 Munchen 21
West Germany
Tel: (089) 576091
Telex: 05-22772

National Semiconductor (UK) Ltd.
301 Harpur Centre
Horne Lane
Bedford MK40 1TR
United Kingdom
Tel: 0234-47147
Telex: 826 209

National Semiconductor
789 Ave. Houba de Strooper
1020 Bruxelles
Belgium
Tel: (02) 4783400
Telex: 61007

National Semiconductor Ltd.
Vodroffsvej 44
1900 Copenhagen V.
Denmark
Tel: (01) 356533
Telex: 15179

National Semiconductor France
Expansion 10000
28, rue de la Redoute
92-260 Fontenay aux Roses
France
Tel: 660 81 40
Telex: NSF250956

National Semiconductor SRL
Via Alberto Mario 26
20149 Milano
Italy
Tel (02) 469 28 64/469 24 31
Telex: 332835

National Semiconductor Sweden
Algrytevagen 23
S-127 32 Skarholmen
Sweden
Tel: (8) 970835
Telex: 10731

NS International Inc., Japan
Miyake Building
1-9 Yotsuya, Shinjuku-Ku 160
Tokyo, Japan
Tel: (04) 355-5711
TWX: 232-2015 JSCJ-J

National Semiconductor (Hong Kong) Ltd.
8th Floor
Cheung Kong Electronic Bldg.
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Tel: 3-899235
Telex: 73866 NSEHK HX
Cable: NATSEMI

NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia
Tel: 03-729-6333
Telex: 32096

National Semiconductor (Pty.) Ltd.
No. 1100 Lower Delta Road
Singapore 3
Tel: 2700011
Telex: NAT SEMI RS 21402

National Semiconductor (Taiwan) Ltd.
3rd Floor, Apollo Bldg.
No. 218-7 Chung Shiao E. Road
Sec. 4, Taipei, Taiwan
P. O. Box 68-332 Taipei
Tel: 7310393-4 7310465-6
Telex: 22837 NSTW
Cable: NSTW Taipei

National Semiconductor (Hong Kong) Ltd.
Korea Liaison Office
6th Floor, Kunwon Bldg.
1-2 Mookjung-Dong
C. P. O. Box 7941
Choong-Ku, Seoul
C.P.O. Box 7941 Seo
Tel: 267-9473
Telex: K24942