AMPRO

COMPUTERS, INCORPORATED

# LITTLE BOARD/PLUS

## TECHNICAL MANUAL

P/N: A74010– B

AMPRO COMPUTERS, INC.
P.O. Box 390427
Mountain View, California 94039
– * –
(415) 962–0230
Telex: 4940302

## PREFACE

This manual is for integrators of systems based on Little Board/PLUS. It contains information on hardware requirements and interconnection, and details of how to use the system. There are five chapters, organized as follows:

Chapter1- GENERAL DESCRIPTION: General information pertaining to the Little Board/PLUS, its major features, and a brief functional description of the board.

Chapter 2 - INTEGRATING A SYSTEM: Descriptions of the external components necessary to construct a CP/M-based system with floppy and hard disk drives. Included are tables listing the pinouts of each of the six board connectors, as well as features and special considerations concerning the board's peripheral interfaces.

Chapter 3 - USING THE SYSTEM: System power-up procedures, and operation with AMPRO's enhanced CP/M operating system and utilities.

Chapter 4 - THEORY OF OPERATION: Detailed technical information on Little Board/PLUS hardware.

Chapter 5 - PROGRAMMER'S REFERENCE: I/O port addresses and programming information regarding custom programming of Little Board/PLUS.

Only brief descriptions and instructions regarding the Little Board/PLUS system software are provided in this manual. For full details on the AMPRO-supplied system software, please refer to the Z80 System Software User's Manual (AMPRO part number A74006).

## PLEASE NOTE

Specifications are subject to change without notice. The contents of this document are believed to be accurate. If errors are found, please notify AMPRO COMPUTERS INC., at the address shown on the title page of this document.

The following is a list of trademarks used within this document:

Z80: Zilog, Inc.
CP/M: Digital Research, Inc.
IBM: International Business Machines
ZCPR3: Richard L. Conn
LITTLE BOARD, LITTLE Board/PLUS, SCSI/PLUS: AMPRO Computers, Inc.

No part of this document may be reproduced in any form, for commercial purposes without the express written consent of AMPRO Computers, Inc.

**Copyright (C) 1985, AMPRO COMPUTERS INCORPORATED**

## TABLE OF CONTENTS


**CHAPTER 1 - GENERAL DESCRIPTION**

1.1 Introduction 1-1
1.2 Features 1-1
1.3 Functional Description 1-1
1.3.1 CPU, Memory, and Timing 1-2
1.3.2 Serial I/O Ports 1-2
1.3.3 Parallel I/O Port 1-2
1.3.4 Floppy Disk Controller 1-2
1.3.5 SCSI/PLUS Multi-Master Bus 1-3
1.4 Specifications 1-4

**CHAPTER 2 - INTEGRATING A SYSTEM**

2.1 Introduction 2-1
2.2 What Is Needed 2-1
2.2.1 DC Power 2-4
2.2.2 Reset, Power LED 2-4
2.2.3 Parallel Printer 2-5
2.2.4 Serial Ports 2-4
2.2.5 Floppy Disk Interface 2-6
2.2.6 SCSI/PLUS Interface 2-7
2.2.7 ID Input Port 2-8
2.3 Board Jumper Configurations 2-8
2.4 Booting the System 2-11
2.5 Troubleshooting 2-12

**CHAPTER 3 - USING THE SYSTEM**

3.1 Introduction 3-1
3.1.1 Software References 3-1
3.1.2 Conventions 3-1
3.2 First Time Use 3-2
3.2.1 Making Backup Disks 3-2
3.2.2 Customizing Your System Disk 3-3
3.3 Operating System Features 3-4
3.3.1 CP/M 2.2 BDOS 3-4
3.3.2 AMPRO Custom BIOS 3-4
3.3.3 ZCPR3 Command Processor 3-5
3.4 AMPRO-Supplied Utilities 3-9
3.4.1 CP/M Utilitiy Programs 3-9
3.4.2 AMPRO Utilities 3-10
3.4.3 ZCPR3 Utilities 3-11
3.4.4 Public Domain Programs 3-12
3.5 Generating Different Systems 3-12

**CHAPTER 4 - THEORY OF OPERATION**

4.1 Introduction 4-1
4.2 CPU, Memory, and Timing 4-1
4.3 Board Control Register 4-2
4.4 Serial Ports 4-2
4.5 Parallel Printer Port 4-3
4.6 Floppy Disk Interface 4-4
4.7 SCSI/PLUS Interface 4-4

**CHAPTER 5 - PROGRAMMERS' REFERENCE**


5.1 Introduction 5-1
5.2 Z80A CPU 5-1
5.3 Memory 5-1
5.4 Board Control Register 5-2
5.5 Counter/Timer Circuit 5-3
5.6 Serial Ports 5-3
5.7 Baud Rate Generation 5-4
5.8 Floppy Disk Interface 5-5
5.9 Parallel Printer Port 5-5
5.10 ID Input Port 5-5
5.11 SCSI/PLUS Interface 5-6
    5.11.1 SCSI (SASI) Programming 5-6
    5.11.2 Simple Bidirectional I/O 5-7

**APPENDIX A - BOARD DIAGRAM, PARTS LIST, AND SCHEMATIC**

**APPENDIX B - TYPICAL INTERFACE CABLES**

**APPENDIX C - USER GROUPS**

**APPENDIX D - COMPONENT DATA SHEETS**

# CHAPTER 1

# GENERAL DESCRIPTION

## 1.1 INTRODUCTION

This chapter provides an overview and functional description of the AMPRO Little Board/PLUS single board computer. It is intended to provide a basic understanding of the Little Board/PLUS, and how it forms the basis of a compact, powerful computer system.

## 1.2 FEATURES

Little Board/PLUS is a complete 8-bit, Z80-based single board microcomputer. It includes all the circuitry, software, and firmware necessary to construct a functional CP/M-based computer system. Some of the main features are:

- 4MHz Z80A 8-bit microprocessor
- 64K bytes dynamic RAM, 4K-32K EPROM
- Two spare counter/timer channels
- Floppy controller capable of controlling from one to four single- or double-sided, single- or double-density, 40- or 80-track mini or micro floppy drives
- Two RS232C serial ports
- One Centronics printer port
- SCSI/PLUS multi-master I/O expansion bus:
  - SASI Disk/Tape controller compatible
  - ANSC X3T9.2 (SCSI) compatible
  - Multiple Little Board networking
  - Simple bi-directional I/O (17 lines)
- Mounts directly to a 5-1/4" disk drive
- Minimum external components
- Power connector and voltages compatible with 5-1/4 inch disk drives

## 1.3 FUNCTIONAL DESCRIPTION

The following paragraphs briefly describe the Little Board/PLUS single board computer. More detailed information can be found in Chapter 4, Theory of Operation.

### 1.3.1 CPU, Memory, and Timing

The heart of the Little Board/PLUS is a Z80A 8-bit microprocessor operating at 4 MHz. All system functions are based on a single 16 MHz master clock. System RESET is provided in two ways: upon power-up and via an external RESET switch.

Two types of memory are present: EPROM and RAM. A single 28-pin EPROM socket provides from 4K to 32K bytes of firmware space. Jumpers are used to program the socket for a 2732, 2764, 27128, or 27256 type EPROM. The EPROM can be enabled and disabled by software.

System RAM consists of eight 64k x 1 bit dynamic RAM devices. Control circuitry for the RAM memory is entirely digital (no one-shots or R-C components) and provides a high degree of reliability.

A Z80 Counter Timer Circuit (CTC) provides four programmable counter or timer channels. Two of the CTC channels provide the baud rate clocks used by the two serial I/O ports. The other two CTC channels are available for use as programmable timers in applications programs, for real-time clock functions, etc.

### 1.3.2 Serial Ports

A Z-80 Dual Asynchronous Receiver/Transmitter (DART) provides two fully programmable serial I/O ports. Each channel has four of the standard RS232C signals: TxD, RxD, RTS, and CTS. These signals are sufficient for interfacing most serial printers, modems, and terminals. In those cases where other interface signals are required for one of the serial ports, handshaking signals can be borrowed from the second port (if not needed by that port). Polarity and use of the handshaking signals is defined by the software.

Programmable baud rate clocks are supplied by the CTC for baud rates up to 9600 baud. Additional circuitry provides baud rates of 19.2K and 38.4K baud, for Port A only. Since the two serial ports are otherwise identical, either can be programmed as a terminal, modem, serial printer, or other RS232C interface.

### 1.3.3 Parallel Printer Port

The parallel printer port provides the 10 essential signals of a Centronics-type printer interface: Data Bits 1-8, Data Strobe, and Busy. Both the Data Strobe (output) and Busy (input) handshake protocols are defined by software.

### 1.3.4 Floppy Disk Controller

A Western Digital 1772 floppy disk controller device provides all of the functions required to interface with standard 5-1/4 inch "mini" -- and most 3 to 4 inch "micro" -- floppy disk drives. The 1772 includes the following capabilities within a single LSI device:

- Digital phase locked loop
- Digital write precompensation
- Motor on start/stop delay
- Software controlled step rates

Timing for the floppy disk interface is derived directly from the 8 MHz system clock, without delay lines, R-C time constants, or one-shots. This again results in a very high degree of system reliability.

### 1.3.5 SCSI/PLUS Multi-Master Bus

A 50-pin "ribbon cable bus" interface which meets the specifications for the popular Small Computer System Interfaces (SCSI) -- formerly called "SASI" -- provides a general purpose multi-master I/O expansion bus. All SCSI Initiator and Target functions are fully supported, including bus arbitration and disconnect/reselect.

In addition, Little Board/PLUS supports the Initiator function of AMPRO's innovative SCSI/PLUS extension to SCSI. This allows connection of up to 64 SCSI/PLUS Target devices, rather than the usual eight device limit of SCSI.

Applications include both direct and shared use of a wide variety of controllers and devices, as well as tightly coupled Little Board networks. For example, one or more Little Boards, an SCSI Winchester controller, and modules providing calendar/clock, serial port expansion, RAM disk, etc., might all coexist on the same SCSI/PLUS bus.

The 17 bidirectional I/O signals of the SCSI/PLUS interface may also be used as general purpose software-controlled digital I/O lines, without SCSI compatibility. In this case the board's 8-bit SCSI bus ID input register can serve as an additional 8-bit input port.

## 1.4 LITTLE BOARD/PLUS SPECIFICATIONS

| | |
|---|---|
| CPU: | 4MHz Z80A, 8-bit microprocessor |
| MEMORY: | 64 kilobytes of dynamic RAM<br>4-32 kilobytes of EPROM |
| TIMER: | Z80A CTC (4 channels)<br>2 channels not used by AMPRO software |
| SERIAL I/O: | Z80A Dual Asynchronous Receiver/Transmitter (DART)<br>Two - RS232C compatible ports: DB-25 female conns.<br>Software-controlled baud rates:<br>Channel A - 75 to 38,400 baud<br>Channel B - 75 to 9600 baud<br>Four standard RS232C signals per port:<br>Data Out<br>Data In<br>Handshake Out<br>Handshake In<br>Two Ground pins |
| PARALLEL I/O: | Centronics-compatible printer port<br>10 signals supported:<br>Data Bits 1-8 - output<br>Data Strobe - output<br>Printer Busy - input<br>12 Ground pins |
| DISK I/O: | No. drives supported: 1 - 4<br>Disk Controller: WD1772<br>Data rate: 250k bps (MFM), 125K bps (FM)<br>Sector size: 128, 256, 512, or 1024 bytes<br>Phase locked loop: digital (8 mHz)<br>Write precompensation: software enabled<br>Drive capacity (formatted):<br>Type 1 (40 track, 1-sided) - 200K bytes<br>Type 2 (40 track, 2-sided) - 400K bytes<br>Type 3 (80 track, 1-sided) - 400K bytes<br>Type 4 (80 track, 2-sided) - 800K bytes |
| SCSI/PLUS BUS INTERFACE: | SASI Compatible<br>ANSI X3T9.2 (SCSI) compatible<br>SCSI/PLUS Initiator compatible<br>Uses NCR 5380 SCSI bus controller |
| POWER: | Same power connector and voltabes as mini and micro floppy disk drives.<br>Power Requirements: +5VDC at 0.95A<br>+9 to 12VDC at 0.05A |
| ENVIRONMENT: | Temperature: 0 to 32 degrees C, operating<br>Humidity: 5 to 95%, non-condensing<br>Altitude: 0-10,000 feet |

| | |
|---|---|
| SIZE: | 7.75 x 5.75 x 0.75 inches |
| SOFTWARE: | Boot program in 2732 EPROM<br>CP/M Version 2.2 and BIOS on diskette<br>Little Board/Plus system utilities<br>Option: FRIENDLY Integrated Operating Environment<br>Option: BIOS and Utilities source code |
| DOCUMENTATION: | Little Board/Plus Technical Manual<br>Little Board/Plus Software Manual<br>Option: SCSI/PLUS Technical Specification |

# CHAPTER 2

# INTEGRATING A SYSTEM

## 2.1 INTRODUCTION

This chapter describes what is required to build a floppy or hard disk based computer system using Little Board/PLUS as the heart of the machine. Details are provided concerning external device requirements, the boards connector pinouts, how to prepare the board for use, and connection of peripherals such as terminals, printers, and modems. Refer to Appendix B for tables.

## 2.2 WHAT IS NEEDED

A very minimum number of external parts will turn the Little Board/PLUS circuit board into a very powerful computer system. The items listed in Table 2-1 are easily available components.

**Table 2-1. System Components**

| Item | Description |
|---|---|
| Floppy drive(s) | 1 to 4 mini or micro floppy drives, 40 or 80 track, single- or double-sided |
| SCSI (SASI) hard disk controller (option) | Adaptec ACB-4000 or Xebec 1410(A), or equivalent SCSI (SASI) hard disk controller |
| Hard disk drive(s) (option) | 5 to 20 Megabyte drive(s); must be compatible with your specific SCSI (SASI) controller |
| Power supply | +12VDC @ 2.0A, +5VDC @ 2.5A (2 floppies); additional power, as required for hard disk options |
| Reset switch | SPST, normally open, w/LED indicator |
| Cables | Serial (2), parallel, floppy, SCSI, and power cables for Little Board/PLUS and drives |
| Cabinet | Housing for completed system |

Figure 2-1 shows the Little Board/PLUS interface connectors. All components can be housed in a very small box. The size is determined primarily by the disk drives, and perhaps the power supply. A space of approximately 7 x 8 x 11 inches is adequate for a dual floppy system. Tables 2-2 through 2-7 list the cable connector pinouts for the board's interface connectors. Table 2-2 provides suggested mating connectors and manufacturers' part numbers.

**Table 2-2. External Connector Part Numbers**

| Board Connector | Function | Part Number |
|---|---|---|
| J1 | Power Connector | Housing: AMP 1-480424-0<br>Contacts: AMP 60619-1<br>(4 req.) |
| J2 | Parallel Printer, Board end | 3M: 3399-6000<br>T&B: 609-2601M<br>Molex: 15-29-8262 |
| | Parallel Printer, Printer end | AMP: 57F-30360<br>3M: 3366-1001<br>T&B: 609-36M |
| J3,4 | Serial Ports A,B | Housing: Molex 22-01-2067<br>Contacts: Molex 08-50-0114<br>(6 req.) |
| J5 | RESET, Power LED | Housing: Molex 22-01-2047<br>Contacts: Molex 08-50-0114<br>(4 req.) |
| J6 | Floppy Disk Interface (Card edge connectors) | 3M: 3463-0001<br>T&B: 609-3415M<br>Molex: 15-29-0341 |
| J8 | SCSI/PLUS Interface | T&B: 609-5000M<br>Molex: 15-29-8502<br>Berg: 66902-150 |

**NOTE**

All Little Board/PLUS software is distributed on double-sided 48 tpi AMPRO format mini floppy diskettes. This may be directly read from, or booted, in either a 48 or 96 tpi mini floppy drive. The only special requirement for **first time** booting of the system is that you have a double-sided 48 or 96 tpi mini floppy drive capable of stepping at a 6 mS step rate, and a terminal capable of 9600 baud operation. Once you have initially booted the system, you may then generate a system disk with alternative defaults, e.g., single-sided, micro floppy, 12 mS step rate, 19.2K baud terminal data rate, etc.

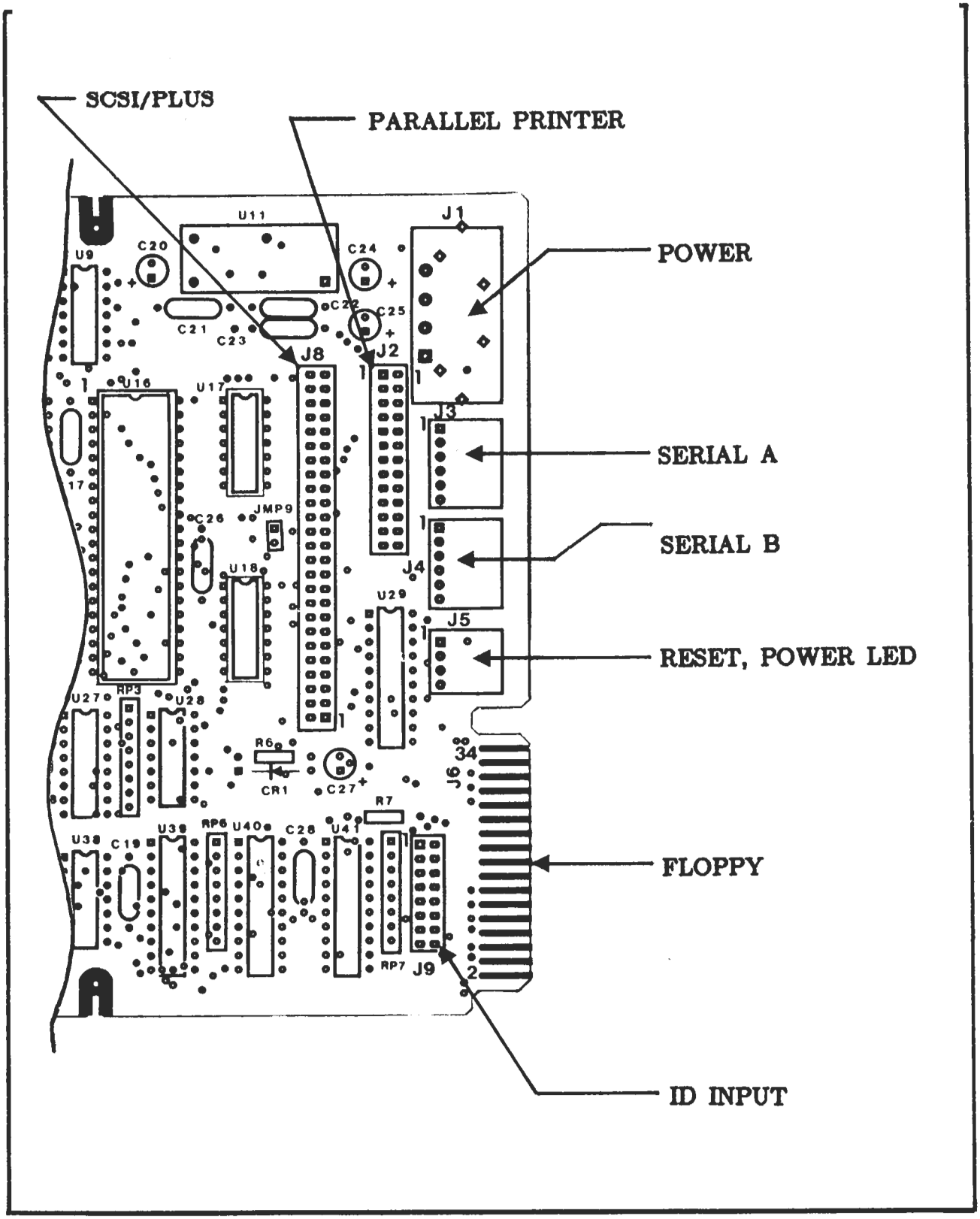



**Figure 2-1. Little Board/PLUS Connector Locations.**

### 2.2.1 DC Power

The power connector (J1) pinout is identical with that of the power connectors on nearly all 5-1/4 inch floppy disk drives. Note that pin 1 on J1 is reversed from the other connectors on the same end of the board. Refer to Table 2-3 for power connections.

**CAUTION**

BE SURE THE POWER PLUG IS CORRECTLY WIRED BEFORE ATTEMPTING TO APPLY POWER TO THE BOARD.

**Table 2-3. Power Connections (J1)**

| Pin | Signal Name | Function |
|---|---|---|
| 1 | +12VDC | +6 to + 15VDC |
| 2 | Ground | Ground return |
| 3 | Ground | Ground return |
| 4 | +5VDC | +5VDC +/- 5% |

### 2.2.2 RESET, Power LED

This connector is for connection to an external SPST switch, to provide the master RESET signal. In addition, a 15 mA current source provides power to an LED power-on indicator. Refer to Table 2-4 for the pinout of connector J5.

**Table 2-4. RESET, Power LED Connector (J5)**

| Pin | Signal Name | Function |
|---|---|---|
| 1 | Ground | To LED Cathode |
| 2 | LED | To LED Anode |
| 3 | Ground | To one side of RESET switch |
| 4 | RESET | To other side of RESET switch |

### 2.2.3 Serial Ports

Table 2-5 lists the connector pinout and signal definitions for each of the two RS232C serial I/O ports. Serial Port A is board connector J3, and Serial Port B is board connector J4. Appendix B gives typical cable wiring for connection to terminals, serial printers, and modems.

**Table 2-5. Serial Connectors (J3/J4)**

| Pin | Signal Name | Function | in/out | DB-25 Pin (DCE) |
|---|---|---|---|---|
| 1 | Ground | Protective Ground | -- | 1 |
| 2 | Ground | Signal Ground | -- | 7 |
| 3 | TxD | Data Output | out | 3 |
| 4 | HSO | Hand Shake Out (RTS) | out | 5 |
| 5 | RxD | Data Input | in | 2 |
| 6 | HSI | Hand Shake In (CTS) | in | 20 |

### 2.2.4 Parallel Printer

The Little Board/PLUS parallel printer connector has a pinout that allows the use of flat ribbon cable between the J2 header and the first 26 lines of a 36 pin male Centronics-type connector at the printer end.

Refer to Table 2-6 for the printer connector's pinout and signal definitions. Note that the pin numbering convention for the board's header connector (J2) differs from that of Centronics connectors, even though the required interconnection cable is straight through. To clarify this, each signal's corresponding Centronics connector pin number has been included in Table 2-6.

**Table 2-6. Parallel Printer Connector (J2)**

| J2 Pin | Signal Name | Function | in/out | Printer Pin |
|---|---|---|---|---|
| 1 | -DS | Data Strobe | out | 1 |
| 3 | Data 1 | LSB of printer data | out | 2 |
| 5 | Data 2 | : | : | 3 |
| 7 | Data 3 | : | : | 4 |
| 9 | Data 4 | : | : | 5 |
| 11 | Data 5 | : | : | 6 |
| 13 | Data 6 | : | : | 7 |
| 15 | Data 7 | : | : | 8 |
| 17 | Data 8 | MSB of printer data | out | 9 |
| 19 | --- | (Not used) | | |
| 21 | BUSY | Printer BUSY | in | 11 |
| 23 | --- | (Not used) | | |
| 25 | --- | (Not used) | | |
| 2-22 | (even) | Signal grounds | | 19-29 |
| 24,26 | --- | (Not used) | | 30,31 |

Appendix B provides a wiring table for a cable which will work with most Centronics compatible printers. The cable required is identical to that used with the Tandy (Radio Shack) TRS-80 Model 100 portable computer. The Tandy cable part number is 26-1409.

### 2.2.5 Floppy Disk Interface

Table 2-7 lists the floppy disk drive interface connector (J6) pinout and signals. A single 34-conductor PC edgecard connector goes at the Little Board end, while there can be from 1 to 4 PC card edge connectors for connecting mini floppy disk drives. Micro floppy drives generally use 34-conductor header connectors, instead.

**Table 2-7. Floppy Disk Interface Connector (J6)**

| Pin | Signal Name | Function | in/out |
|---|---|---|---|
| 2 | -LOW SPEED | Speed select (option) | out |
| 4 | --- | (Not used) | -- |
| 6 | -DRIVE DEL 4 | Drive Select 4 | out |
| 8 | -INDEX | Index pulse | in |
| 10 | -DRIVE SEL 1 | Drive Select 1 | out |
| 12 | -DRIVE SEL 2 | Drive Select 2 | out |
| 14 | -DRIVE SEL 3 | Drive Select 3 | out |
| 16 | -MOTOR ON | Motor on control | out |
| 18 | -DIR SEL | Direction select | out |
| 20 | -STEP | Step | out |
| 22 | -WRITE DATA | Write data | out |
| 24 | -WRITE GATE | Write gate control | out |
| 26 | -TRACK 00 | Track 00 | in |
| 28 | -WRITE PRT | Write protect | in |
| 30 | -READ DATA | Read data | in |
| 32 | -SIDE ONE | Side select | out |
| 34 | -READY | Drive ready (option) | in |
| 1-33 | (all odd pins) | Signal grounds | -- |

Nearly any type of soft-sectored, single- or double-sided, 40- or 80-track, mini (5 1/4") or micro (3" to 4") floppy disk drive is usable with the Little Board. Naturally, the higher the quality of the drives you use, the better your system's reliability. Here are some things that may be helpful for you to know:

- As indicated earlier in this chapter, the default system parameters programmed on the standard Little Board/PLUS system disk, as shipped, require that drive A be a double-sided 48 or 96 tpi mini floppy drive, with 6 mS (or faster) stepping rate. Once you have booted that disk, you can create a new system disk with alternative defaults, allowing you to subsequently boot from any other system-compatible drive.

- The drives used must be compatible with the AMPRO floppy disk interface (see Table 2-7). This interface is generally referred to as SA-450 compatible.

- Use of high quality floppy disk drives is recommended; preferably having DC servo-controlled direct drive motors.

- The drives must be capable of 12 mS (or faster) stepping.

- More than one type of floppy disk drive, up to four, can be present in the system, and in any mix.

- Each disk drive must be jumpered for a specific Drive Select value, through . Drive A is , B is , etc.

- Resistive terminations should be installed only on the drive connected to the last interface cable connector (farthest from the computer).

- When using drives with a Head Load option, jumper the drives for "head load with motor on" rather than "head load with drive select" jumpering.

### 2.2.6 SCSI/PLUS Interface

The SCSI/PLUS interface (J8) uses a 50-pin male header connector to interface with SCSI-compatible peripherals. Table 2-8 shows the signal names and pin numbers for the SCSI/PLUS bus interface connector. Refer to your disk controller documentation, or the ANSI SCSI (X3T9.2) specification, for information on the signal functions.

**Table 2-8. SCSI Interface Connector (J8)**

| Pin | Signal | Function |
|---|---|---|
| 1 - 49 | (All odd pins) | Signal grounds |
| 2 | -DB0 | Data Bit 0 (LSB) |
| 4 | -DB1 | " " 1 |
| 6 | -DB2 | " " 2 |
| 8 | -DB3 | " " 3 |
| 10 | -DB4 | " " 4 |
| 12 | -DB5 | " " 5 |
| 14 | -DB6 | " " 6 |
| 16 | -DB7 | " " 7 (MSB) |
| 18 | -DBP | Data Parity |
| 20,22,24 | Ground | Signal Ground |
| 25 | --- | (Not used) |
| 26 | TERMPWR | Termination +5VDC |
| 28,30 | Ground | Signal Ground |
| 32 | -ATN | Attention |
| 34 | Ground | Signal Ground |
| 36 | -BSY | Busy |
| 38 | -ACK | Transfer Acknowledge |
| 40 | -RST | Reset |
| 42 | -MSG | Message |
| 44 | -SEL | Select |
| 46 | -C/D | Control/Data |
| 48 | REQ | Transfer Request |
| 50 | -I/O | Data direction |

This interface can serve a variety of purposes, including connection of hard disk controllers, tape controllers, printer and communication servers, etc. In addition, the interface signals may be used as direct input/output lines, allowing the connection of TTL-level interfacing devices and sensors, etc. The output signals are open collector, 48mA drivers, and may be enabled and disabled under software control. On-board removable resistor networks provide bus termination.

### 2.2.7 ID Input Port

Eight pairs of jumper pins (J7) provides an ID Input Port, which can be used by SCSI-related software to determine the board's SCSI bus ID for bus arbitration (or other) purposes. If not required for this purpose the ID Input Port can be used as a general purpose 8-bit input port connector, with a flat ribbon cable connector plugged onto J7. Table 2-9 shows the signal names and pin numbers for J7.

**Table 2-9. ID Input Port (J7)**

| Pin | Signal | Function |
|---|---|---|
| 1 | -ID7 | ID Bit 7 (MSB) |
| 3 | -ID6 | " " 6 |
| 5 | -ID5 | " " 5 |
| 7 | -ID4 | " " 4 |
| 9 | -ID3 | " " 3 |
| 11 | -ID2 | " " 2 |
| 13 | -ID1 | " " 1 |
| 15 | -ID0 | " " 0 (LSB) |
| 2-16 | Ground | Signal Ground |

## 2.3 BOARD JUMPER CONFIGURATIONS

Little Board/PLUS contains nine sets of jumpers, which may be used to customize the board's operation. The jumper sets consist of either two or three pins, with pin 1 a square pad; each jumper set is outlined with white boxes on the component side of the board. The options available through these jumpers are described in the following paragraphs.

**NOTE**

Do not solder wires to the jumper pins, as this may void your board's warranty. Use wire wrap, or shorting plugs instead.

For normal operation, no jumper setup is required. All of the jumper settings are factory-set, by means of shorting plugs on the pins, or traces on the bottom side of the board. Use of J7 (SCSI bus ID) varies, according to software requirements. AMPRO's standard hard disk software does not require the use of J7.

**JMP1 - Clock**

This jumper, when shorted, connects the system 16 mHz master clock bus to the 16 mHz hybrid oscillator (U2). It is intended for test purposes only.

FACTORY SETTING: shorted

**JMP2 - CTC CLK/TRG3**

This jumper connects the CTC's CLK/TRG3 input (JMP2-2) to either the 1772 FDC's interrupt request output signal (JMP2-3) or to the CTC's ZC/TC2 output signal (JMP2-1).

FACTORY SETTING: both options open

**JMP3 - CTC CLK/TRG2**

This jumper, when shorted, connects the 5380 SCSI controller's interrupt request output signal (JMP3-2) to the CTC'S CLK/TRG2 input (JMP3-1).

FACTORY SETTING: open

**JMP4 - U6 Option**

This jumper is shorted for 8 mHz-only operation of the 1772 FDC device. NOTE: If this jumper is shorted, the clock multiplexor IC, U6 must not be present; if U6 is present, this jumper must be open.

FACTORY SETTING: open (U6 present)

**JMP5,6 - EPROM Type**

These jumpers are used to program the board for various types of EPROM devices. These two jumpers are set as shown in Table 2-10.

**Table 2-10. EPROM Jumper Configurations**

| EPROM Type | JMP5: 1-2 | JMP5: 2-3 | JMP6 |
|---|---|---|---|
| 2732 | open | short | * |
| 2764 | * | * | open |
| 27128 | short | open | open |
| 27256 | short | open | short |
| Factory setting: | open | short | open |

**Notes**

(1) Items indicated by * can be open or shorted; however do not short both JMP5 options simultaneously.

(2) When a 2732 24-pin EPROM is used, it must be plugged into the **lower** 24 pins of the 28-pin EPROM socket, so that the 2732's pin 1 plugs into pin 3 of the socket.

**JMP7 - DART RIA**

This jumper, when shorted, connects the 1772 FDC's DRQ output singal (JMP7-2) to the DART's RIA input (JMP7-1).

FACTORY SETTING: open

**JMP8 - DART DCDA**

This jumper, when shorted, connects the 5380 SCSI controller's DRQ output singal (JMP8-1) to the DART's RIA input (JMP8-2).

FACTORY SETTING: open

**JMP9 - SCSI Termination Power**

This jumper, when shorted, connects the board's +5V DC power to the SCSI bus signal line provided for optional external termination.

FACTORY SETTING: open

### JMP10,11 - 1772 Reset

These jumpers provide a choice of hardware or software control of the 1772 FDC's master reset input. Shorting JMP10 connects the board's hardware -RESET signal (JMP10-1) to the 1772's reset input (JMP10-2). Shorting JMP11 connects the DART's DTRB output (JMP11-2) to the 1772's reset input (JMP11-1).

FACTORY SETTING: JMP10 shorted, JMP11 open (hardware RESET)

### U17,22 - SCSI Bus Termination

These two 14-pin sockets contain resistive termination networks for SCSI bus termination. Every SCSI bus should have one set of termination at each end of the bus. These networks should be removed from the board if the bus is terminated by two other devices. If the SCSI interface will not be required, power may be saved by removing these termination networks, along with the 5380 SCSI controller, from the board. For non-SCSI use, the 220/330 termination networks may also be replaced with alternate values.

**NOTE**

> The on-board resistive termination networks (U17, U22) should be present on two, and only two, SCSI bus devices. Be sure that the board's SCSI bus is terminated in at least one place (generally on the board); a non-terminated SCSI bus may "hang" the system up due to indeterminate signal levels.

## 2.4 BOOTING THE SYSTEM

With the system completed, only connection to a terminal and a source of power for the computer are required. Typical cable wirings for connection to terminals, serial printers, and modems appear in Appendix B.

To boot from the standard distribution diskette, set your terminal as follows:

| | |
|---|---|
| Baud Rate: | 9600 |
| Data Bits: | 8 |
| Parity: | off |
| Stop Bits: | 1 |

If your terminal provides the choice, set it so that the data bit 8 is transmitted as a 0 ("low" or "space"). Some terminals do not provide such an option, automatically sending a zero for data bit #8 when parity is off. The AMPRO BIOS ordinarily masks the MSB when 8 bit transmission is selected, but you may choose to do otherwise. (Requires BIOS source code, available from AMPRO).

Connect the terminal to Serial Port A. If a modem is being connected, use Serial Port B. First time booting of the system requires that you connect a serial terminal capable of meeting the above specifications. As will be explained in Chapter 3, system boot parameters can be customized, but initially the system comes up with Serial Port A set for 9600 baud.

With a terminal connected and turned on, the system is ready to boot. When power is applied, a program in the EPROM will attempt to read the operating system from disk. If no disk is in the drive, the system will wait until a disk is in place, and drive latch closed. The system will then read the CP/M operating system from the disk in drive A.

**IMPORTANT**

See note in Table 2-1 concerning floppy drive requirements for booting the distribution system diskette.

If the drive's LED lights but nothing else happens, try inserting the flip-side of the disk and pressing RESET. If this doesn't help, refer to the next section for troubleshooting information.

## 2.5 TROUBLESHOOTING

It is possible that the completed system does not work the first time. If you have to troubleshoot it, here are some suggestions:

- Recheck all wiring, soldered connections.
- Check that power is available from the power supply.
- Be certain that the drives are working, and are jumpered correctly. NOTE: IBM PC drives are not jumpered in a "standard" manner; the drive cable has swapped drive select wires. For use with Little Board/PLUS, be sure drive A is jumpered as Drive Select 0, B as 1, etc.
- If both drive indicators light during power-up, with drive handles closed (across slot), the drive signal cable is probably reversed on the board's floppy interface connector (J6). Switch the computer OFF and reverse the drive cable connector at the Little Board/PLUS.
- Check the drive termination resistor pack(s) for proper location. Normally, this will be located at the drive connected at the end of the drive cable.
- If you have the debugging Monitor EPROM option, you can verify some of the system functions using the debugger and other tools in the Monitor.

In the event that your system still does not boot after following all of these instructions, contact AMPRO customer service for assistance.

# CHAPTER 3

# USING THE SYSTEM

## 3.1 INTRODUCTION

Assuming you have successfully booted CP/M as described in Chapter 2, you will probably want to take advantage of the flexibility designed into the Little Board/PLUS CP/M BIOS and utilities to customize your system diskette.

This chapter explains how to install your operating system software for various system configurations, making use of standard CP/M and ZCPR3 utilities and the Little Board/PLUS utilities software supplied on the system software diskette.

### 3.1.1 Software References

Only brief references are made in this manual to the use and operation of the required software utilities. Whenever a software utility is mentioned, it will either be called an AMPRO utility, a CP/M utility, or a ZCPR3 utility. This way you will know where to obtain further information on the program's use.

Full descriptions and operating instructions for the AMPRO software utilities are found in the AMPRO Z80 System Software User's Manual, AMPRO part number A74006.

Some recommended CP/M and ZCPR3 references are:

CP/M Primer, by Stephen M. Murtha and Mitchell Waite (Howard W. Sams)

CP/M Handbook, by Rodnay Zaks (Sybex Inc.)

CP/M Revealed, by Jack D. Dennon (Haydon Publishing)

ZCPR3: The Manual, by R. L. Conn (Echelon Inc. - Phone: 415/948-3820)

### 3.1.2 Conventions

In the descriptions of the use of software utilities, terminal keyboard inputs which you will make to the system are shown underlined. This has been done to make it easy for you to distinguish between the computer's prompts and the operator's keystrokes. For example:

A0>DIR <RETURN>

Also, certain keys on your terminal's keyboard have special uses. The control key, generally labeled CTRL, is meant to be pressed at the same time as one other key. The required control key combination will be represented as follows: <CTRL-C> = control key pressed along with C key.

Two other special keys are the "escape" key, indicated by <ESC> and the "return" key (also called the "carriage return" or "enter" key), indicated by <RETURN>. In general, all commands you enter from the CP/M (or ZCPR3) command prompt require you to press <RETURN> key to begin the operation, as in the example above.

## 3.2 FIRST TIME USE

Chapter 2 discussed connecting the completed system to terminal, a modem, and initial power-up. Assuming the computer works, there are two things it is recommended that you do immediately:

1. Make a backup copy of the disks included with the Little Board/PLUS.

2. Customize your system diskette.

### 3.2.1 Making Backup Disks

It is always a good idea to have at least one backup copy of all floppy disks. This is especially true of your master system disks. The exact procedure you use to make backup disks depends on your system configuration. Here are three methods:

**Method 1: Two identical format types.**

When making a backup in which the source and destination disks will be the same floppy format (i.e., 48 to 48 tpi, or 96 to 96 tpi), the backup can easily be made with the Copy function of the AMPRO AMPRODSK utility. The only catch is that AMPRODSK requires the source and destination to be the same floppy format and drive type. One exception is that double-sided drives can be used to copy from or to single-sided floppy formats. The AMPRODSK Copy funtion even formats the destination disk for you. Simply follow the instructions given by the program when you run it.

**NOTE**

AMPRODSK can not read 48 tpi disks in 96 tpi drives. Use Method 2.

**Method 2: Two different drive types.**

You can backup a source disk onto a different floppy format (e.g. 96 tpi backup of 48 tpi disk or visa versa), as follows:

1. Use the AMPRODSK utility's Format function to format a fresh disk having the desired destination format, in the destination drive.

2. Use a file copy utility (CP/M PIP, ZCPR3 DISK7, ZCPR3 MCOPY, AMPRO FRIENDLY, etc.) to copy all files from the source to the destination disk.

3. If the source disk is a system disk, use the AMPRO SYSGEN utility to copy the source system tracks to the destination system tracks.

**Method 3: Single-drive Systems.**

If you have a system with only one disk drive, you can do nearly anything that can be done with two or more drives. For example, a backup of your system software diskette can be made as follows:

1. Use the AMPRO AMPRODSK utility's Format function to format a blank disk. The program will indicate what you need to do.

2. Use the AMPRO SWAPCOPY utility to copy all files from the source disk system disk to the backup disk. The required command is:

   A0>SWAPCOPY *.* <RETURN>

3. Use the AMPRO SYSGEN utility to copy the system tracks from the source disk to the backup disk. The program will indicate what to do.

### 3.2.2 Customizing Your System Disk

When using your system for the first time, some of the system initial default values are probably not perfect for your system configuration. The AMPRO CONFIG utility allows you to easily modify the serial port setups (baud rates, handshaking, etc.), printer port assignment (serial or parallel), floppy drive step rates, and automatic powerup/reset command.

**NOTE**

> Any modifications to the system parameters should only be performed using your backup disks. **Do not** use the disks shipped with your Little Board/PLUS.

One important parameter to be sure to set correctly is the floppy disk drive step rate. Initially, the system disk is set up for, and boots with, a step rate of 6 mS. Check your drive's step rate specification, and set the default step rate to the one that is closest to the drive's specification. The CONFIG utility will guide you through its use.

You will also want to customize the terminal characteristics definition file, MYTERM.Z3T, so that it matches your terminal. This is done with the ZCPR3 TCSELECT or TCMAKE utilities. The menu which displays when you boot your system will look better, and write to the screen faster, once you have created a MYTERM.Z3T file for your terminal.

## 3.3 OPERATING SYSTEM FEATURES

The operating system included with the Little Board/PLUS is an enhanced version of standard CP/M version 2.2. One major difference is that the user command line interface (Console Command Processor, CCP) is replaced by an enhanced Z80 Command Processor Replacement called ZCPR3. The Little Board/PLUS operating system consists of three portions:

| | |
|---|---|
| CP/M 2.2 BDOS: | Standard CP/M file and device management facility |
| AMPRO Custom BIOS: | Enhanced Basic I/O System |
| ZCPR3 CCP: | Z80 Command Processor Replacement version 3 |

### 3.3.1 CP/M 2.2 BDOS

The heart of the Little Board/PLUS operating system is the Basic Disk Operating System (BDOS), which is the normal CP/M 2.2 BDOS. This is the part of the operating system that interacts with programs. Because this is completely standard CP/M 2.2, all software programs designed to work with CP/M 2.2 will run without modification, provided they do not contain hardware-dependant routines.

### 3.3.2 AMPRO Custom BIOS

The CP/M Basic I/O System (BIOS) takes care of all hardware-dependant operating system functions. Many features of the Little Board/PLUS CP/M operating system are the result of a highly flexible, sophisticated BIOS implementation. Here are a few:

**Automatic Format Sensing** - Single- and double-sided, 40- and 80- track disks and drives may be intermixed freely. The BIOS senses what format is present, and adjusts accordingly. 40-track (48 tpi) diskettes may even be read (but not written) in 80-track (96 tpi) drives. NOTE: CP/M requires that you use a <CTRL-C> keystroke when you change diskettes.

**Alien Format Support** - One disk drive can be assigned as an "emulating" drive, so that you can read from or write to your choice of non-AMPRO format.

**Hard Disk Support** (Optional) - You can add one or more hard disk controllers and drives to your system. The BIOS contains generic SCSI (SASI) support, making it compatible with a wide variety of devices, and has been structured to maximize the flexibility of this function.

**Power-up Port Defaults** - You can easily alter the system power-up I/O port defaults (console baud rate, printer port assignment and setup, etc.) using the AMPRO configuration utility.

**Power-up Auto-Command** - A single command can be specified to run automatically on system power-up or reset. This is one of the options available through the AMPRO configuration utility.

**IOBYTE Implementation** - The IOBYTE can be changed by the CP/M STAT utility, to reassign logical I/O devices to physical I/O devices. Table 3-1 lists the standard CP/M logical-to-physical device assignments and choices, as supported in the Little Board/PLUS BIOS.

**Table 3-1. Logical-to-Physical I/O Assignments**

| Logical Device | Physical Device Choices | Default |
|---|---|---|
| CON: | CRT: or TTY: | CRT: |
| RDR: | TTY: (input) | TTY: |
| PUN: | TTY: (output) | TTY: |
| LST: | CRT: or TTY: or LPT: | LPT: |
| Where: | CRT: = Serial Port A<br>TTY: = Serial Port B<br>LPT: = Parallel Printer Port | |

As implemented, the IOBYTE allows two choices for console port (Serial Port A or Serial Port B), and two choices for printer port (Serial Port A, Serial Port B, or Parallel Printer Port).

In addition, the AMPRO CONFIG utility program allows you to set the IOBYTE either temporarily, or in the cold-boot defaults on the system tracks of a disk.

### 3.3.3 ZCPR3 Command Processor

The normal CP/M console command processor (CCP) has been replaced with the more powerful Z80 Command Processor Replacement, version 3 (ZCPR3). As indicated in Table 3-2, the ZCPR3 implementation differs slightly from standard CP/M, but can be used in the much same way you would use standard CP/M. If you wish to eliminate the ZCPR3 enhancement, you can do so through the use of the AMPRO MOVCPM utility. (Refer to the AMPRO Z80 System Software User's Manual for information on how to do this.)

Your system software includes only part of the full power of the ZCPR3 System; the full ZCPR3 System occupies several megabytes of disk! The following paragraphs cover the ZCPR3 features that are present in the standard Little Board/PLUS operating system software. Additional ZCPR3 options can be easily added, as discussed in the AMPRO Z80 System Software User's Manual. Contact Echelon Inc. (Phone: 415/948-3820) for additional ZCPR3 information and support. ZCPR3 utilities and information are also available at no charge through many CP/M and ZCPR3 user groups and bulletin board systems (see Appendix C).

**Table 3-2. ZCPR3/CCP Command Comparison**

| Function | ZCPR3 Command | CCP Command |
|---|---|---|
| Display all files | DIR | DIR |
| Display files in specific DU | DIR DU: | No equivalent |
| Erase specified file | ERA DU:afn | ERA D:afn |
| Erase with verify | ERA DU:afn V | No equivalent |
| Rename file | REN DU:ufn=ufn2 | REN D:ufn=ufn2 |
| Rename file over existing file | REN DU:ufn=ufn2 | No equivalent |
| Print file on console Without paging | TYPE DU:ufn P | TYPE D:ufn |
| Print file on console With paging | TYPE DU:ufn | No equivalent |
| Save memory into file | SAVE n DU:ufn | SAVE n D:ufn |
| Save memory into file and specify size in hex | SAVE nH DU:ufn | No equivalent |
| Save memory into file and specify number of blocks | SAVE n DU:ufn S or SAVE nH DU:ufn S | No equivalent |
| Change disk | D: | D: |
| Change user | U: | USER n |
| Change disk and user at same time | DU: | No equivalent |

| | |
|---|---|
| DU: - Drive number,User number | (e.g., A0:, B15:, C:, 13:) |
| ufn - Unambiguous file name | (e.g., MYFILE.TXT, DIR.COM) |
| afn - Ambiguous file name | (e.g., *.COM, MYFILE.*, M??ILE.T?T) |

**Sub-Directories**

Each floppy disk has a directory of files; each directory can contain up to 16 sub-directories (also called user areas), numbered 0 through 15. Normal CP/M uses the USER command to change between the 16 possible sub-directories, with the default being 0. ZCPR3 uses a directory label formed from the combination of the drive letter (A, B, etc.) and user area (0,1, etc.). This is called a drive-user, or "DU" expression. For example, A0 represents drive A user area 0, while B15 corresponds to drive B user area 15.

Using ZCPR3, the current drive and user area are displayed in the command prompt. Instead of using CP/M's USER command to change user areas, you do it in the same way that you change drives. In addition, whenever you use the DU expression, you may omit either the letter or number portion, if that part of the expression is the same as the current one. For example:

```
A0>B15:<RETURN>
B15>0:<RETURN>
B0:A:<RETURN>
A0>
```

In addition, functions such as directory (DIR), erase (ERA), rename (REN), etc., allow the DU form as destination and source directory designations.

Another powerful feature of ZCPR3 is the option of "named" sub-directories. When the named directory option is present, a directory name can be substituted for the DU expression in all command line inputs. This feature is not present in the AMPRO system software as shipped, but can be easily added. Please refer to the above-mentioned software references for further information.

### Directory Utility

In AMPRO's ZCPR3 implementation, the DIR utility is not an "intrinsic" (internal) function, but requires the presence of the ZCPR3 DIR.COM utility on disk. As you will notice as soon as you use this command, the DIR utility has quite a few nice features, such as alphabetical file sorting and direct access to any directory. For example

```
A0>DIR B5:<RETURN>
```

displays the directory of drive B, user area 5 (sorted alphabetically!).

Also, since the directory utility is disk-based rather than internal, you can select from a large assortment of public domain directory utilities -- simply rename your favorite one "DIR.COM".

### Multiple Commands per Line

With ZCPR3, multiple commands may be given on a single command line, with semi-colons (;) used as separators. For example, the sequence

```
A0>DIR;ERA *.BAK;DIR<RETURN>
```

runs the directory program, erases all files with the .BAK type, and then runs the directory program a second time.

### Command Search Path

ZCPR3 also uses an automatic command search path. This means that programs referenced on the command line may be located anywhere along a pre-defined command search path. You can be logged onto drive B, and execute a program on drive A, without typing the drive prefix for the program drive. The default search path is:

current drive, current user
current drive, user 0
drive A, current user
drive A, user 0
drive A, user 15
current drive, user 15

Since the search path covers both different drive letters and different user area numbers, you can "hide" programs and utilities in different user areas. This results in cleaner looking directories. A common practice is to "hide" COM files (programs) in user 15. Such files will not be visible from user 0, but will execute from user 0.

**NOTE**

> Some application programs must be run from the same drive letter and user area as the files they will be used with, or require additional programs, overlays, or files to be present in the same directory (drive and user area) as the program itself.

The ZCPR3 DISK7 and MCOPY, and the AMPRO FRIENDLY utilities can be used to copy files directly between user areas. The ZCPR3 PATH utility allows you to easily change the search path as needed.

### Intrinsic Commands

With the exception of the DIR and USER commands noted above, all standard CP/M version 2.2 intrinsic commands are implemented, as well as some additions. Table 3-2 lists the ZCPR3 commands versus those of the standard CP/M CCP.

### Aliases

One of the most powerful features of ZCPR3 is the use of aliases. This feature is made possible by the multiple command line capability. An "alias" is a disk-resident multiple command line. The alias has a command file name, such as FUNCTION.COM, but represents a pre-programmed set of commands. Whenever you run the alias, you get the set of commands. It is like a fast, memory-based submit, or batch, facility. By using an alias (usually STARTUP.COM) as the CONFIG auto-command, you can have a complex sequence of functions automatically initialize your system on power-up or reset.

### Shells

ZCPR3 also provides shell support. A "shell" is a substitute operating environment. Examples of ZCPR3-compatible shells are ZCPR3 MENU, VMENU, and VFILER, and AMPRO FRIENDLY. A shell is a program that always reloads following the execution of any program, rather than returning you to the command prompt. Once a shell is loaded, you might never see the A0> prompt again! ZCPR3 shells provide varying levels of isolation of the user from the

operating system, and can even completely replace the CCP interface. The powerful MENU shell program is included on your system diskette.

### Termcap Facility

ZCPR3 adds another powerful feature to CP/M which is lacking in most microcomputer operating systems: a termcap facility. The AMPRO CP/M implementation contains a special buffer area in memory which is used by ZCPR3 to standardize terminal display control codes. This allows application programs to be terminal-independent, providing the software is written to take advantage of the ZCPR3 termcap. The ZCPR3 utilities TCSELECT and TCMAKE are used to create a termcap file, usually called MYTERM.Z3T. The ZCPR3 utility LDR is used to load the appropriate termcap file into memory, for use by compatible programs.

## 3.4 AMPRO-SUPPLIED UTILITIES

A powerful set of software programs is sipplied with the Little Board/PLUS. This section contains brief descriptions of each. They include:

- the standard CP/M 2.2 software set
- the AMPRO Little Board/PLUS utilities
- several key ZCPR3 utilities
- several public domain programs

The programs described below are those included on the Little Board/PLUS system software diskette at the time of this writing (hard disk software is optional). Actual contents may vary. Complete descriptions and operating instructions are provided in the publications listed in the introduction to this chapter. Program and program description updates are available from AMPRO on an on-going basis, at nominal charge. Contact AMPRO for information.

### 3.4.1 CP/M Utility Programs

Included with the Little Board/PLUS are all the standard CP/M utility programs:

**ASM.COM** - Standard assembler for 8080 instructions. May be used to assemble Little Board/PLUS source code.

**DDT.COM** - Dynamic Debugging Tool: standard CP/M debugger.

**DUMP.COM** - Permits display of a file in hexadecimal values.

**ED.COM** - Standard CP/M line editor. May be used to edit Little Board/PLUS source code.

**LOAD.COM** - Converts .HEX file output of the ASM program to an executable .COM file.

**PIP.COM** - Permits single or multiple disk-to-disk file transfers. Also port-to-port and port-to/from-disk transfers.

**STAT.COM** - Status of disk and other I/O devices. Also may be used to set file attributes.

**SUBMIT.COM** - Permits execution of multiple commands and parameters stored in a disk file.

**XSUB.COM** - For use with SUBMIT.COM, to allow passing of parameters direct to programs.

### 3.4.2 AMPRO Utilities

The following programs are specific to Little Board/PLUS, and used for system customization, disk formatting, disk format translation, etc. Source code is available from AMPRO at nominal cost. The hard disk utilities are available in the optional Z80 Hard Disk Software package.

**AMPRODSK.COM** - Used to copy, format, and verify AMPRO-format disks.

**CONFIG.COM** - Used to modify or set your system's current or powerup default peripheral port characteristics according to your particular requirements. Lets you set serial port A and B baud rates, data characteristics, and hand-shaking, floppy drive step rates, printer port choice (serial or parallel), and command for power-up or reset automatic execution.

**DOS.COM** - Used to read and write files on MS-DOS format disks. Also used to read the directory and erase files.

**DOSFMT.COM** - Used to format MS-DOS disks in all standard formats.

**ESET.COM** - Permits reading and writing of data to and from disk formats other than those available with the MULTIDSK.COM utility. (See MULTIDSK.COM)

**HFORMAT.COM** (optional) - Hard disk formatting program.

**HINIT.COM** (optional) - Hard disk controller and system initialization utility.

**HPARK.COM** (optional) - Hard disk drive head parking utility. Moves the head to an unused area of the disk drive's surface prior to system shut down, to prevent data loss from head crash.

**MOVCPM.COM** - Configures the operating system for a user-definable memory size. Same as ZMOVCPM.COM, except contains the standard CP/M CCP. Used as part of the procedure for generating a hard disk system, if ZCPR3 is not desired.

**MULTIDSK.COM** - Provides compatibility with other computers' disk formats. After MULTIDSK is run, you can read from or write to the selected alien format by using the drive letter "E" instead of the drive's normal designation (A, B, etc.).

**MULTIFMT.COM** - Permits formating (and verifying) disks using non-AMPRO formats.

**SET.COM** - Allows setting of current serial port characteristics (baud rate, data characteristics, hand shaking) and assignment of printer port (serial or parallel). Similar to CONFIG.COM, but all parameters are given from the command line, thus allowing use with ALIASes, MENU lines, etc.

**SWAP.COM** - Re-assigns CP/M disk drive letters, swapping them in pairs.

**SYSGEN.COM** - Used to write the AMPRO CP/M operating system tracks onto a disk. Allows source of the system tracks to be either another disk's system tracks, a disk file, or a memory image (generally placed in memory by MOVCPM or ZMOVCPM).

**ZMOVCPM.COM** - Configures the operating system for a user-definable memory size. Same as MOVCPM.COM, except contains the standard ZCPR3 CCP replacement. Used as part of the procedure for generating a hard disk system.

### 3.4.3 ZCPR3 Utilities

The following ZCPR3 utilities are included on the Little Board/PLUS system disk. Source code is available from ECHELON Inc. (415/948-3820) at nominal cost.

**ALIAS.COM** - Used to create or modify multiple command line files (aliases).

**DIR.COM** - Displays contents of disk directories. Allows direct drive/user area (DU) access.

**DIFF.COM** - File compare utility. Checks two files for differences.

**DISK7.COM** - Easy to use disk file management utility. Includes a menu of single-keystroke commands for Copy, Rename, Delete, Length, and drive Status.

**LDR.COM** - Used to load terminal definition files (e.g. MYTERM.Z3T), system environments, and other system-resident ZCPR3 files.

**MCOPY.COM** - General purpose file copying program. Allows direct file movement between directories (e.g. A0 to B15).

**MENU.COM** - Powerful system menu shell program.

**PATH.COM** - Modifies command search path.

**TCMAKE.COM** - Used to create non-standard terminal definition files (e.g. MYTERM.Z3T).

**TCSELECT.COM** - Used to select a standard terminal definition file from a menu of standard terminals.

**UNERA.COM** - Recovers deleted disk files. Inverse of the ERA (erase) command.

**WHEEL.COM** - Sets user privileges. (Required to use PATH.COM to change command search paths.)

**Z3INS.COM** - ZCPR3 installation utility. Installs other ZCPR3 utilities for your operating system configuration. AMPRO-suplied ZCPR3 utilities do not require installation prior to use with the standard AMPRO-supplied operating system.

**ZEX.COM** - Memory-resident submit facility, similar to CP/M's SUBMIT utility, but more powerful.

### 3.4.4 Public Domain Programs

Several valuable public domain programs have also been included. Source code for these programs is available through CP/M user groups and bulletin board systems (see Appendix C).

**FINDBAD.COM** - Bad sector lockout program for use in mapping out hard disk drive surface defects. Creates a file [UNUSED].BAD containing all bad sectors.

**MDM740.COM** - General purpose, powerful communication program. Modified for use with the Little Board/PLUS serial port B. (AMPRO-specific overlay is contained in the file, M&-LB.ASM.) Allows direct computer-to-computer file transfer over Rs232, or may be used with a modem. Fearures include ASCII transfer or XMODEM protocol, auto dialing, stored phone library, and more.

**SD.COM** - Directory display utility alternative to DIR.COM. Options you may specify in the command line allow printing the directory, creating a file containing the directory, and inclusion of multiple user areas.

**SWAPCOPY.COM** - Single drive disk-to-disk copy utility. Modified for use with AMPRO foreign formats (allows copying from A to A, A to E, and E to A).

## 3.5 GENERATING DIFFERENT SYSTEMS

There are several reasons why you may wish to generate an alternate operating system:

1. Use of hard disk drives
2. Alternate ZCPR3 system configurations
3. Memory requirements of a modified BIOS or custom software
4. Substitution of standard CP/M CCP for ZCPR3 CCP
5. Generation of a larger TPA system, using the Version 1 BIOS

In the first three cases, additional buffer areas are required in high memory, above the operating system. This requires moving the operating system **down** in memory, and leaving room for the required functions. In the fourth case, the use of CP/M results in less memory required for the operating system, allowing the operating system bo be moved up in memory. In the fifth case, slignty more program area is made available by using a BIOS with a few less features.

The AMPRO utilities MOVCPM.COM or ZMOVCPM.COM are used to relocate or regenerate the operating system. Refer to the AMPRO Z80 System Software User's Manual for additional information on the generation of alternate CP/M configurations.

# CHAPTER 4

# THEORY OF OPERATION

## 4.1 INTRODUCTION

This chapter provides detailed information on the functional operation of Little Board/PLUS. No information on the internal operation of the LSI components is included. Please refer to the manufacturers' data sheets (Appendix D) for specific details. Figure 4-1 is a block diagram of Little Board/PLUS. Chapter 5 contains programming information, and indicates the assignment of programmable device pins to specific hardware signals and functions.

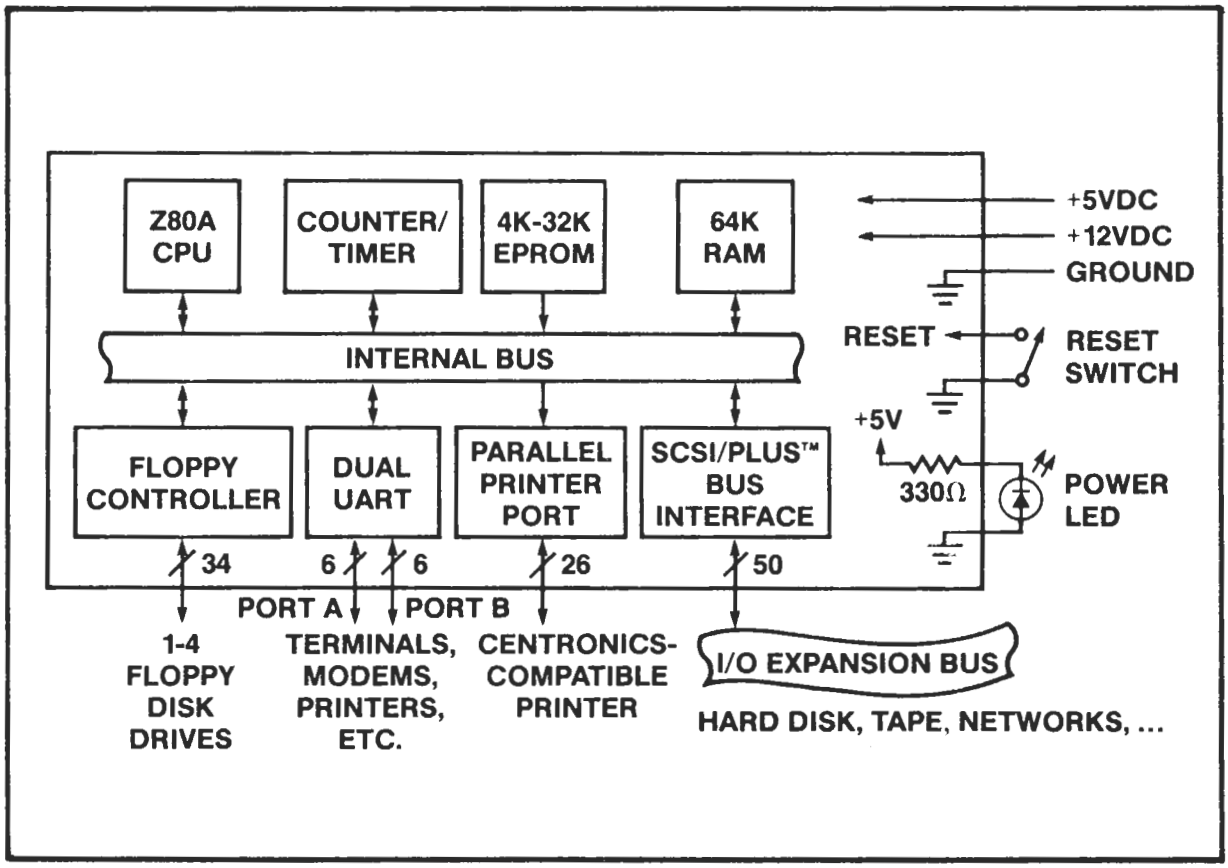



**Figure 4-1. Little Board/PLUS Block Diagram**

## 4.2 CPU, MEMORY, AND TIMING

The main system time base is provided by a 16 MHz oscillator module. A binary counter provides three system clocks: 8 MHz, 4 MHz, and 2 MHz. The 4 MHz signal is used by the Z80A, Counter Timer Circuit (CTC), and Dual Asynchronous Receiver/Transmitter (DART) devices. The 8 MHz signal provides the normal clock input to the 1772 Floppy Disk Controller (FDC).

The Z80A interrupt "daisy chain" is implemented in accordance with the standard Zilog protocol, using the peripheral devices' Interrupt Enable Input and Interrupt Enable Output signals. Several of the CTC and DART input lines have jumper options which allow those devices to optionally function as interrupt controllers for a number of floppy and SCSI interface signals. (See the jumpering information in Chapter 2.)

All control signals for the 64K dynamic RAM are derived from the system's 4 and 8 MHz clocks and the Z80A refresh output signal. RAM devices with access times up to 200 nS can be used. The Z80A generates the required 7-bit refresh addresses and timing required by the 64K by 1 bit dynamic RAM devices.

When a memory read or write occurs with address line A15 set to zero, and bit 6 of the Board Control Register is set to zero, memory address decoding logic selects the EPROM rather than RAM. In addition, a wait state generator becomes active whenever the EPROM is selected, permitting the use of EPROM device access times up to 450 nS.

A programmable array logic (PAL) device and a pair of two-to-four decoders generate the device select addresses for all of the Little Board's I/O devices. Table 4-1 shows the device select addresses in binary. Where X's are indicated in the table entries, the corresponding address bit may be a 1 or a 0. I/O address groups indicated as "unused" are available for I/O expansion via the Z80A CPU socket. There are 40 unused I/O addresses. A summary of the I/O ports, addresses and functions, are shown in Table 4-2.

**Table 4-1. I/O Device Addresses**

| Device Select | I/O Address (Binary) |
|---|---|
| Board Control Register | 0000 0X00 |
| Parallel Port Data Latch | 0000 0X01 |
| Parallel Port Strobe Set | 0000 0X10 |
| Parallel Port Strobe Clear | 0000 0X11 |
| (Unused) | 0000 1XXX |
| (Unused) | 0001 XXXX |
| 5380 Chep Select | 0010 0XXX |
| 5380 DMA Acknowledge | 0010 1XX0 |
| ID Input Port | 0010 1XX1 |
| (Unused) | 0011 XXXX |
| CTC | 01XX XXXX |
| DART | 10XX XXXX |
| FDC | 11XX XXXX |

**Table 4-2. Summary of I/O Ports**

| Address | Input/Output | Function |
|---|---|---|
| 00h | Output | Board Control Register |
| 01h | Output | Parallel Printer Data Register |
| 02h | Output | Parallel Printer Data Strobe Set |
| 03h | Output | Parallel Printer Data Strobe Clear |
| 20-27h | I/O | SCSIControllerInternal Registers |
| 28h | I/O | SCSI Controller "DMA" Read/Write |
| 29h | Input | ID Input Port |
| 40h | I/O | CTC Channel 0 |
| 50h | I/O | CTC Channel 1 |
| 60h | I/O | CTC Channel 2 |
| 70h | I/O | CTC Channel 3 |
| 80h | I/O | DART Channel A Data |
| 84h | I/O | DART Channel A Control |
| 88h | I/O | DART Channel B Data |
| 8Ch | I/O | DART Channel B Control |
| C0h | Output | FDC Command Register |
| C1h | Output | FDC Track Register |
| C2h | Output | FDC Sector Register |
| C3h | Output | FDC Data Register |
| C4h | Input | FDC Status Register |
| C5h | Input | FDC Track Register |
| C6h | Input | FDC Sector Register |
| C7h | Input | FDC Data Register |

**NOTE**

The I/O device addresses for these ports are not unabiguously decoded. Refer to Table 4-1.

## 4.3 BOARD CONTROL REGISTER

The Board Control Register (BCR) is a simple octal output latch which controls several board functions. Seven of the eight bits of the BCR control are associated with the floppy disk interface: drive selects (4), side select, density select, and 1772 clock select. These are discussed in the section about the floppy disk interface (below). One bit in the BCR also serves to enable or disable the EPROM device. All outputs of the BCR are cleared (to 0) by the board's RESET signal, selecting the EPROM at power-up or when the RESET signal is active.

## 4.4 SERIAL PORTS

A Z80A dual asynchronous receiver/transmitter (DART) forms the basis of two RS232C serial I/O ports. Baud rates for these ports are generated by the Z80A CTC. Channel A of the DART has an alternate baud rate clock source (614.385kHz), which is obtained by dividing the 16 MHz system clock by 104. This provides serial channel A with two additional baud rates: 19,200 and 38,400 baud, as well as 9600 baud.

Baud rate selection is accomplished by programming the CTC time constants, selecting the CTC channel mode (counter or timer), programming the DART prescale factor (16, 32, or 64), and, for serial I/O channel A, selecting either the high or low speed baud rate mode. The high/low baud rate mode of channel A is controlled using the DTRA output of the DART.

DART RTSA and RTSB output signals generate each channel's output handshake signal, while CTSA and CTSB provide the two channels' handshake inputs.

RS232C signal levels are converted to and from TTL levels by a 75188/1488 line driver and 75189/1489 line receiver. An on board -12 volt DC-to-DC converter provides the -12VDC power for the line driver.

Several of the DART's input and output signals are used for purposes not associated with serial communications:

- DCDA and RIA are available as optional interrupt sources
- DCDB is available for reading the optional floppy disk ready signal
- RIB is used to sence the parallel printer Busy signal

## 4.5 PARALLEL PRINTER PORT

An octal D-latch with a 24 mA output current sinking capacity is used to drive the eight parallel printer port data lines. In addition, there are two handshaking signals: Data Strobe (output) and Busy (input).

The Data Strobe output is generated by a flip-flop which is set and reset by software. This permits software controlled timing of data relative to strobe, and strobe polarity. The printer Busy input is sensed by the RIB input of the DART.

## 4.6 FLOPPY DISK INTERFACE

Nearly all of the logic required for the floppy disk interface is provided by the Western Digital 1772 Floppy Disk Controller (FDC) device. Only the drive and side-selection and interface input signal buffering require additional devices.

The Board Control Register (BCR) controls the state of the four drive select lines, the side select line, the 1772 density select input, and the 1772 master clock input rate. In principle, by switching the 1772 master clock rate to 16 mHz, single-density eight inch disk drive data rates (250 kbits/sec) are possible. However, the current version of the 1772 is not guaranteed to function properly with a 16 mHz input clock, so the use of this option is not recommended at this time.

The floppy disk interface Ready signal connects to the DART DCDB input signal. This can be optionally used with floppy disk drives which provide this signal. The AMPRO Little Board/PLUS BIOS does not use this option.

## 4.7 SCSI/PLUS INTERFACE

The SCSI/PLUS multi-master bus interface consists of an SCSI bus controller IC (NCR 5380) and an 8-bit jumperable bus ID input port. All of the signals of the SCSI/PLUS expansion bus interface connect directly to pins of the 5380 bus controller IC.

A 74LS244 is used as an ID input port, allowing the state of eight jumpers to be read under software control. Alternatively, this port may be used for general purpose input sensing.

The 5380 integrates approximately 20 components in a single 40-pin package. It allows full programmable control of 17 bi-directional bus signals, and provides both buffered (low leakage) bus inputs and high current (48 mA) bus output drive capacity. A pair of socketed 220/330 ohm termination networks provide optional on-board SCSI bus termination.

The 5380 is fully compatible with the ANSC X3T9.2 (SCSI) standard, including all roles and all phases. In addition, the 5380 can support the Initiator role of the proposed SCSI/PLUS enhancement to SCSI, but not the SCSI/PLUS Target role.

You can obtain a copy of the NCR 5380 SCSI Interface Chip Design Manual from:

NCR Microelectronics - (303) 596-5612 or (800) 525-2252

You may also wish to obtain a copy of the ANSI SCSI specification. Copies of this standard may be obtained by sending $20, and a self-addressed mailing label (for each copy desired) to:

The X3 Secretariat
Computer and Business Equipment Manufacturers Association
311 First Street, N.W. - Suite 500
Washington, DC 20001

In addition, the SCSI/PLUS Preliminary Technical Specification, which details AMPRO's proposed enhancement to SCSI to allow 64 (rather than 8) bus devices, is available from AMPRO.

# CHAPTER 5

# PROGRAMMER'S REFERENCE

## 5.1 INTRODUCTION

This chapter discusses programming techniques and peripheral device register addresses and requirements. Programming most Little Board/PLUS devices is straight forward. However, the floppy disk controller is relatively complex to program; we recommend that you modify the BIOS and utilities source code (available from AMPRO) rather than create custom floppy disk drivers. For more complete information on device functions not covered here, refer to the data sheets in Appendix D of this manual.

## 5.2 Z80A CPU

The Z80A 8-bit CPU operates at a 4.00 MHz clock rate. No wait states occur during RAM access; one wait state occurs for each EPROM access.

The Z80A's non-maskable interrupt (NMI) input is not connected; only maskable interrupts can occur. All maskable interrupt modes are supported. The Z80 interrupt priority daisy chain is fully implemented, with the following prioritization:

- The CTC device has the highest interrupt priority, with each channel sub-prioritized: channel 0 is highest, channel 3 lowest.
- DART Channel A (middle priority)
- DART Channel B (lowest priority)

A number of jumper options on the board allow interrupts to be registered from the floppy and SCSI interface controller devices. These jumpers allow the CTC's channel 2 and 3 clock inputs, and the DART's external status inputs (RIA and DCDA) to be used as optional interrupt sources. This allows a variety of interrupt sources, each with its own prioritization and vectoring. Refer to the jumpering information in Chapter 2, and to the board schematic, for more details.

## 5.3 MEMORY

When the EPROM enable bit in the Board Control Register is low, the EPROM is enabled in the lower 32K bytes of RAM. When the EPROM is enabled, the EPROM contents are repeated reduntantly throughout the lower 32K bytes of memory. When the EPROM enable bit in the BCR is high, the EPROM disappears, leaving 64K bytes of RAM.

## 5.4 BOARD CONTROL REGISTER

An eight bit register Board Control Register (BCR) controls seven FDC related functions, and provides the EPROM enable/disable control. A write to I/O address 00h immediately changes the state of the BCR's eight output signals, which are defined in Table 5-1.

**Table 5-1. Board Control Register Programming.**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Signal | FDCHI | EPROM | DDEN | SIDE1 | DS4 | DS3 | DS2 | DS1 |

| Bit | Signal | Function |
|---|---|---|
| 7 | FDCHI | FDC Controller master clock select: 0 = 8 mHz; 1 = 16 mHz |
| 6 | -EPROM | EPROM Enable 0 = enabled; 1 = disabled |
| 5 | -DDEN | Double density enable: 0 = enabled |
| 4 | SIDE1 | Floppy drive side select 0 = side 0; 1 = side 1 |
| 3 | DS4 | Drive Select 4, 1 = select drive 4 |
| 2 | DS3 | Drive Select 3, 1 = select drive 3 |
| 1 | DS2 | Drive Select 2, 1 = select drive 2 |
| 0 | DS1 | Drive Select 1, 1 = select drive 1 |

**FDCHI (data bit 7)** - allows the master clock input to the 1772 to be switched between 8 MHz (FDCHI = 0) and 16 MHz (FDCHI = 1). This theoretically allows the 1772 to function at eight-inch floppy drives, when its clock is set to 16 MHz. However, the 1772 is currently not guaranteed to work reliably with a 16 MHz clock. It is therefore not recommended that you use this capability at this time. If you do attempt to switch the 1772's clock input, we suggest the following algorithm:

1. Read and store 1772 internal register contents. Note whether the motor on bit is active.
2. Deselect all drives.
3. Re-write BCR with new speed
4. Re-write BCR with desired drive select.
5. Restore 1772 register contents. Be sure to restore the condition of the motor activity (i.e., turn it on with a null seek, etc., if it was on previously.

The reading and restoring of the 1772 registers, and restoration of motor on status, protect against corruption during clock switches. The deselection of drives is done to protect media from accidental write pulses during clock switches.

The signals controlled by the BCR are utilized as follows:

**-EPROM (data bit 6)** - enables the EPROM and disables the lower 32K bytes of RAM, when a 0 is written to data bit 6 of the BCR, and visa versa, when a 1 is written to that bit.

**-DDEN (data bit 5)** - places the 1772 in double density operation when a 0 is written to data bit 5 of the BCR, and single density when a 1 is written to that bit.

**SIDE1 (data bit 4)** - selects side one of the floppy media when it is set to 1, and selects side 0 of the floppy media when it is set to 0.

**DS4-DS1 (data bits 3-0)** - select one of four drive units when the corresponding bit is set to 1. Do not set more than one drive select to 1 at any one time.

On power-up or RESET, all bits in the board control register are automatically cleared (set to 0's).

## 5.5 COUNTER/TIMER CIRCUIT (CTC)

The CTC device contains four independent counter/timers addressed at I/O addresses 40h, 50h, 60h, and 70h, as shown in Table 5-2.

**Table 5-2. CTC Register Addresses**

| Address | CTC Channel |
|---|---|
| 40H | 0 |
| 50H | 1 |
| 60H | 2 |
| 70H | 3 |

All Channels are read/write

The CTC master clock is the 4.00 MHz system clock. The Clock input for Channels 0 and 1 is 2.00 MHz. Each of the four addresses is both a read and a write register representing one of the CTC channels. It is through these locations that the CTC is programmed. The CTC has the following assigned channel functions and options:

**CTC Channels 0 and 1** - Baud rate generators for DART Channels A and B, respectively. The CLK/TRG inputs for these channels are connected to 2.00 MHz. Each of these channels can be used in either the Counter or Timer mode to generate a full range of baud rates (discussed below).

**CTC Channels 2 and 3** - Not normally used by Little Board/PLUS software. These channels are available for use in timing functions, and may be cascaded. In addition, they may be used as interrupt controllers for floppy or SCSI functions, etc. These options can be configured using the board's jumpers.

You may use CTC channels 2 or 3 as interrupt controllers by programing the channel in Counter mode, with a count of 1, triggerable on a rising or falling edge, as appropriate. Use the jumper options described in Chapter 2.

You may cascade channels 2 and 3, for accurate real time clock functions, by jumpering channel 2's ZC/TC2 output to channel 3's CLK/TRGB input (see jumpering information, Chapter 2).

## 5.6 SERIAL PORTS

A Z80 DART device forms the basis of board's two RS232C serial ports. In addition, three of the DART I/O signals support other on-board functions, and several more are available for use as general purpose interrupt sources.

The DART internal registers are accessed through four, non-consecutive I/O addresses, as shown in Table 5-3. Each register is both read and write.

**NOTE**

In order to read the current state of the DART external status signals (CTS, RI, DCD, etc), a Reset External Status command must first be sent to the associated DART channel.

**Table 5-3. DART Register I/O Addresses.**

| Address | Function |
|---|---|
| 80H | Channel A, Data |
| 84H | Channel A, Control |
| 88H | Channel B, Data |
| 8CH | Channel B, Control |

All are read/write.

### 5.6.1 DART Channel A Signals

DART Channel A input/output signals are utilized as shown in Table 5-4. Note that the high/low baud rate select for Channel A uses the DTRA signal, and that DCDA and RIA are available as optional interrupt sources.

**Table 5-4. DART Channel A Pin Assignment.**

| Signal Name | DART Pin | Function |
|---|---|---|
| Serial Port Functions: | | |
| Transmit Data | TXDA | output to RS232C |
| Receive Data | RXDA | input from RS232C |
| Handshake Out | RTSA | output to RS232C |
| Handshake In | CTSA | input from RS232C |
| Data Clock | RXCA, TXCA | input from CTC ZC/TC0 pin |
| Additional Functions: | | |
| Low Baud Select | DTRA | Serial Port A baud rate mode |
| (options) | DCDA, RIA | Optional interrupt sources |

### 5.6.2 DART Channel B Signals

DART Channel B input/output signals are assigned as shown in Table 5-5. Note the additional functions. The floppy drive ready signal is not used by the Little Board/PLUS BIOS.

**Table 5-5. DART Channel B Pin Assignment.**

| Signal Name | DART Pin | Function |
|---|---|---|
| Serial Port Functions: | | |
| Transmit Data | TXDB | output to RS232C |
| Receive Data | RXDB | input from RS232C |
| Handshake Out | RTSB | output to RS232C |
| Handshake Input | CTSB | input from RS232C |
| Data Clock | RXCB, TXCB | input from CTC ZC/TC1 pin |
| Additional Functions: | | |
| Printer BUSY* | RIB | input from printer interface |
| (option) | DCDB | input from FDC interface |

## 5.7 BAUD RATE GENERATION

Both serial ports use clocks provided by the CTC for baud rates up to 9600. You set the channel's baud rate input clock by setting the associated CTC channel mode and time constant, and programming the DART channel prescale factor (16, 32, or 64). In addition, Serial Port A can be placed in a high speed mode, in which a 615.385 kHz signal is used as the baud rate clock input to DART channel A. This allows Serial Port A baud rates of 9600, 19.2k, and 38.4k.

**NOTE**

The Little Board/PLUS CP/M BIOS contains tables of DART and CTC initialization parameters, along with a BIOS call which initializes the DART and CTC devices. It is recommended that you utilize the BIOS tables and initialization call, or at least update the contents of the tables after modifying a device's operational characteristics.

### 5.7.1 Below 9600 Baud

Serial Port A baud rates are determined by CTC channel 0 (in that port's low speed mode), and Serial Port B baud rates are determined by CTC channel 1. For Serial Port A low speed mode (baud rates of 9600 baud or less), program DART output DTRA as a 1 (active). Program each CTC and DART channel as shown in Tables 5-6 and 5-7. For information on programming of the CTC and DART devices, refer to the device data sheets (Appendix D). Table 5-7 lists the available baud rates through the CTC clock sources.

**Table 5-6. Baud Rate Programming, up to 9600 Baud.**

| Device Function | Setting |
|---|---|
| CTC Interrupt | Disable |
| CTC Mode | per Table 5-7 |
| CTC Prescaler | *16 |
| CTC CLK/TRIG edge | Either |
| CTC Timer Trigger | *Set to automatic |
| CTC Time Constant | per Table 5-7 |
| DART Scale Factor | per Table 5-7 |
| * = don't cares in Counter mode | |

**NOTE**

These settings assume Channel A's low baud rate mode, obtained by setting DART output DTRA to 1.

**Table 5-7. CTC and DART Scale Factors**

| Desired Baud Rate | CTC Time Constant | CTC Channel Mode | DART Scale Factor | Actual Baud Rate |
|---|---|---|---|---|
| 9600 | 13 | Counter | 16 | 9615 |
| 4800 | 26 | Counter | 16 | 4808 |
| 2400 | 52 | Counter | 16 | 2404 |
| 1200 | 104 | Counter | 16 | 1202 |
| 600 | 208 | Counter | 16 | 601 |
| 300 | 208 | Counter | 32 | 300 |
| 110 | 142 | Timer | 16 | 110 |

**NOTE**

Other combinations may be used to obtain higher, non-standard, baud rates -- up to 125K baud.

### 5.7.2 Above 9600 Baud

Serial Port A can also be programmed to standard baud rates of 9600, 19.2K, and 38.4K, using the high baud rate mode. Higher, "non-standard" baud rates are also possible, by using different CTC and DART scale factors than those indicated in Table 5-7. For example, using a CTC time constant of 1 (in counter mode), and a DART scale factor of 16, results in 124.8K baud.

To select Serial Port A's the high baud rate mode, CTC Channel 0 must be turned off with a software reset, and the DART DTRA output must be cleared (set to 0). The values shown in Table 5-8 represent the required DART Scale Factor to be written to the DART. To program Serial Port A for the higher baud rates:

- Issue a software reset to CTC Channel 0. (Write a 03H byte as a control word to CTC Channel 0.)
- Clear DART channel A's DTRA (set to 0).
- Set DART channel A's scale factor as indicated in Table 5-8.

**Table 5-8. DART Channel A Settings, High Baud Rate Mode**

| Desired Baud Rate | DART Scale Factor | Actual Baud Rate |
|---|---|---|
| 38400 | 16 | 38462 |
| 19200 | 32 | 19230 |
| 9600 | 64 | 9615 |

## 5.8 FLOPPY DISK INTERFACE

A Western Digital 1772 Floppy Disk Formatter/Controller (FDC) occupies I/O addresses C0H thru C7H. Since the A2 address line is connected to the -R/W input of the 1772, read and write registers in the FDC occupy unequal addresses, as shown in Table 5-9. (This differs from the 1772 data sheet description.) The floppy interface signals associated with devices other than the 1772 are shown in Table 5-10.

**Table 5-9. 1772 Register Addresses**

| Address | Function | Read/Write |
|---|---|---|
| C0H | Command register | Write |
| C1H | Track register | Write |
| C2H | Sector register | Write |
| C3H | Data register | Write |
| C4H | Status register | Read |
| C5H | Track register | Read |
| C6H | Sector register | Read |
| C7H | Data register | Read |

**Table 5-10. Non-1772 Floppy Signals**

| Interface Signal | Source/Destination |
|---|---|
| Drive select 4 | BCR, bit 3, output |
| Drive select 3 | BCR, bit 2, output |
| Drive select 2 | BCR, bit 1, output |
| Drive select 1 | BCR, bit 0, output |
| Drive Ready | DART DCDB input |

Due to the complexity of programming of the floppy disk interface, we recommend that you modify the standard Little Board/PLUS FDC drivers (source is available from AMPRO), rather than creating new custom routines.

## 5.9 PARALLEL PRINTER PORT

The parallel printer interface supports eight data output bits (D1 - D8), a Data Strobe (output), and a printer Busy signal (input). With the exception of Busy, these signals are controlled as shown in Table 5-11. The Data Strobe flip-flop is cleared (Data Strobe = 0) upon power-up or RESET.

**Table 5-11. Parallel Printer Port Addresses**

| Address | Function |
|---|---|
| 01H | 8-bit data register written to by CPU. CPU data bit 0 = printer D1; bit 1 = printer D2; ...; bit 7 = printer D8. |
| 02H | A write to this address sets the data strobe flip-flop. |
| 03H | A write to this address clears the data strobe flip-flop. |

The data strobe flip-flop is cleared on power-up or RESET.

The printer Busy signal is connected to the DART RIB input; the DART Channel B status register must be read for RIB status. NOTE: In order to read the current state of the printer Busy signal, DART Channel B must first be sent a Reset External Status command. Note also that the sense of this signal is inverted: when printer Busy = 1, RIB reads 0 (inactive); when printer Busy = 0, RIB reads 1 (active).

## 5.10 ID INPUT PORT

This port can either be used for SCSI bus ID, for general purpose jumper settings, or as an 8-bit general purpose data input port.

The ID input port is read by an input from I/O address 29h. The jumpering of the eight pairs of pins at location J7 on the board determines the data byte obtained. The input buffer is non-inverting: the data read directly reflects the level on the input pin. When a jumper is inserted, the corresponding data is low (0); when out, the data bit is high (1).

Jumper assignment is as follows: J7 pins 1 and 2 corresponds to data bit 7; pins 3 and 4 are data bit 6; ...; pins 15 and 16 are data bit 0.

## 5.11 SCSI/PLUS INTERFACE

The SCSI/PLUS interface is controlled by means of an NCR 5380 SCSI Protocol Controller device. The 5380 contains 8 readable and 8 writable registers. These are addressed as shown in Table 5-12.

**Table 5-12. 5380 Internal Registers**

| Read Registers | Address |
|---|---|
| Current SCSI Data | 20h |
| Initiator Command Register | 21h |
| Mode Register | 22h |
| Target Command Register | 23h |
| Current SCSI Bus Status | 24h |
| Bus & Status Register | 25h |
| Input Data Register | 26h |
| Reset Parity/Interrupt | 27h |
| **Write Registers** | **Address** |
| Output Data Register | 20h |
| Initiator Command Register | 21h |
| Mode Register | 22h |
| Target Command Register | 23h |
| Select Enable Register | 24h |
| Start DMA Send | 25h |
| Start DMA Target Receive | 26h |
| Start DMA Initiator Receive | 27h |

The SCSI/PLUS interface has a wide variety of applications, including:

- use with SCSI (SASI) disk controllers and devices
- use with up to 64 SCSI/PLUS Target devices
- use as a bidirectional I/O port
- use as a multi-master network bus

If you plan to use SCSI (SASI) devices not supported by the standard Little Board/PLUS BIOS, you can either use the SCSI BIOS call provided for that purpose, or modify the BIOS source code as required.

If you plan to use program the 5380 yourself, you will probably require a copy of the NCR 5380 SCSI Interface Chip Design Manual. Contact:

NCR Microelectronics - (303) 596-5612 or (800) 525-2252

You may also wish to obtain a copy of the ANSI SCSI specification. Copies of this standard may be obtained by sending $20 and a self-addressed mailing label (for each copy desired) to:

The X3 Secretariat
Computer and Business Equipment Manufacturers Association
311 First Street, N.W. - Suite 500
Washington, DC 20001

In addition, the SCSI/PLUS Preliminary Technical Specification, which details AMPRO's proposed enhancement to SCSI to allow 64 (rather than 8) bus devices, is available from AMPRO.

### 5.11.1 SCSI (SASI) Programming

When using the SCSI/PLUS interface with SCSI (SASI) disk controllers, special programming is not generally required; the AMPRO BIOS hard disk driver and AMPRO hard disk utilities accomodate many types of disk controllers and disk drives. Installation of the hard disk software is all that is generally required, providing you are using controller and drive types supported by the BIOS. (Refer to AMPRO Z80 System Software User's Manual.)

When using the 5380 in SCSI (SASI) applications, care must be taken to meet the specified timing constraints. For detailed timing information, consult your peripheral controller's technical manual, or the SCSI specification referenced above. The Little Board/PLUS BIOS source code (available from AMPRO) represents an excellent example of how to use the 5380 in SCSI applications.

### 5.11.2 Simple Bidirectional I/O

If you plan to program the 5380 yourself, you will need a copy of the NCR 5380 design manual mentioned above. The 5380 has 17 bidirectional I/O lines, which may be used as inputs or outputs under software control.

The 5380 has two operating modes: Initiator and Target modes. In Initiator mode, several conditions are required before data output to the I/O bus can be active. If the device is used in the Target mode, however, these special conditions are not applicable. This results in more straight forward programming of simple 8-bit I/O applications, and is recommended for simple bidirectional I/O.

The 5380 is placed in Target mode by writing 40h to the Mode Register. Once in Target mode, all 17 I/O signals except ACK and ATN may be used as both inputs and outputs. In Target mode, ACK and ATN are inputs only. The data lines (DB0-7,P) are outputs when bit 0 ("Assert Data Bus") of the Initiator Command Register is a 1, and inputs when bit 0 of that register is a 0.

Eight additional inputs are available via the ID Input Port, discussed above. Also, the parallel printer port can also provide an additional set of eight outputs and two handshake signals, if it not required as a printer interface.

# APPENDIX A

# BOARD DIAGRAM, PARTS LIST, AND SCHEMTAIC

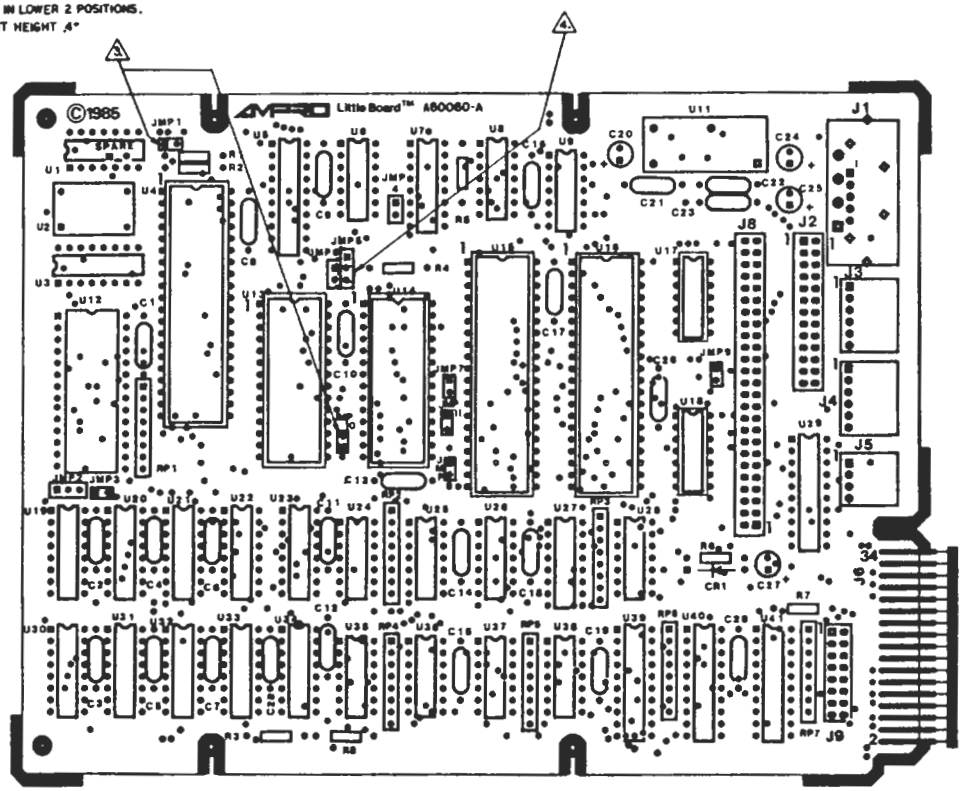
| SIZE D | DRAWING NUMBER | A60060-A |
|---|---|---|

**REVISIONS**

| LTR | ZONE | ECO NO | DESCRIPTION | APPD | DATE |
|---|---|---|---|---|---|
| A | ... | ... | INITIAL RELEASE | RK | 1-8-85 |

NOTES: UNLESS OTHERWISE SPECIFIED

1. Square pads indicate pin one (1).
2. CAUTION: Static sensitive devices. Handle with appropriate precautions to prevent damage from electrostatic discharge!
3. INSTALL CONNECTOR SHUNT.
4. INSTALL CON. SHUNT IN LOWER 2 POSITIONS.
5. MAX. INSTALLED SHUNT HEIGHT .4"




| ITEM | QTY | PART NUMBER | DESCRIPTION | SPECIFICATION |
|---|---|---|---|---|

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES XX ±.01 XXX ±.005

DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS

| | BY | DATE |
|---|---|---|
| DWG | | 1-8-85 |
| CHK | | 1-8-85 |
| APPD | | 1-8-85 |
| REL | | 1-8-85 |

APPLICATION

MATERIAL

CHEMICAL FINISH

SCALE 2X

DO NOT SCALE DRAWING

SHEET 1 OF 3

AMPRO COMPUTERS INCORPORATED

TITLE: CPU 1B ASSY

SIZE D | DRAWING NUMBER A60060-A

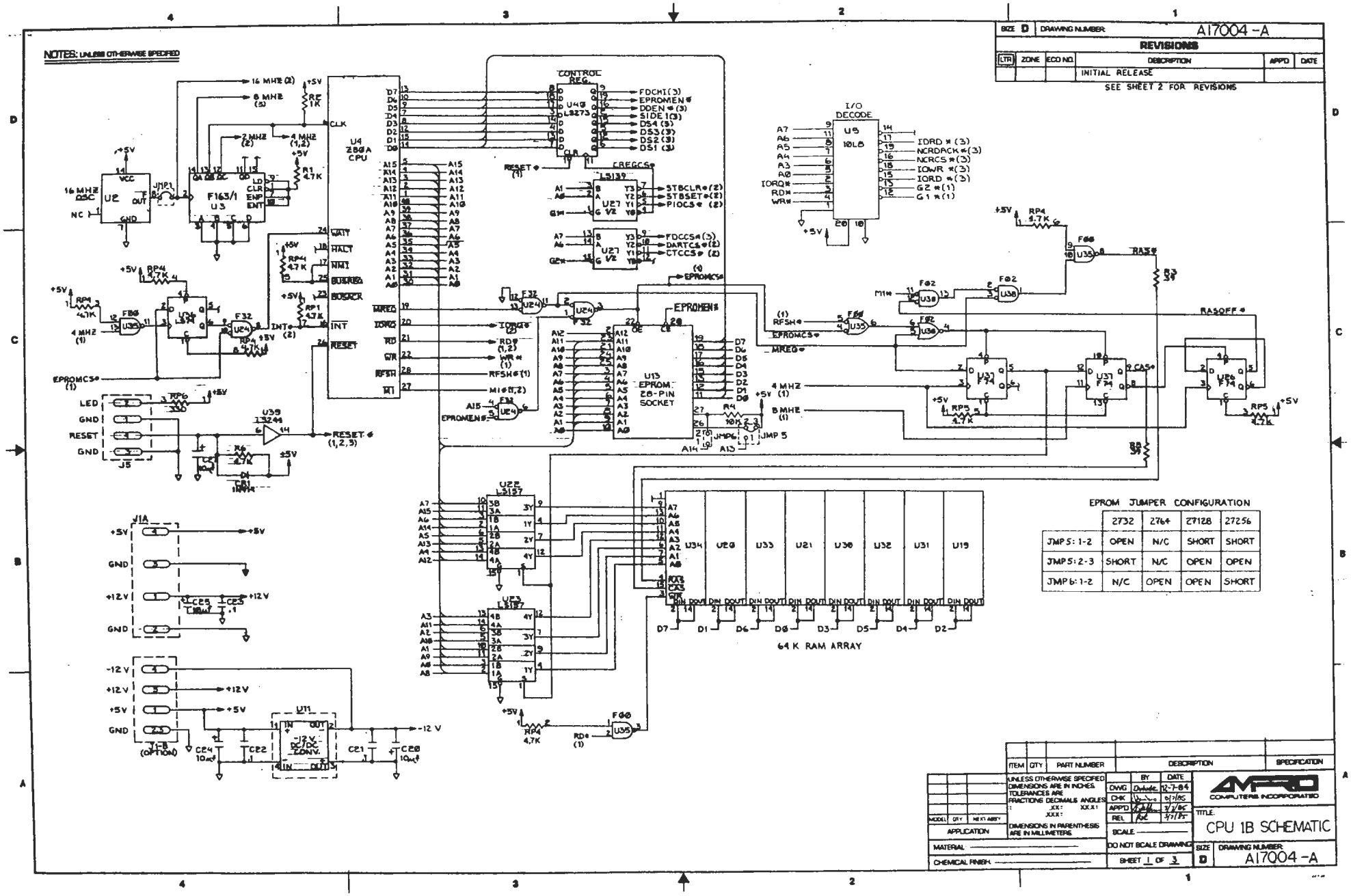
AMPRO COMPUTERS INCORPORATED - PARTS LIST

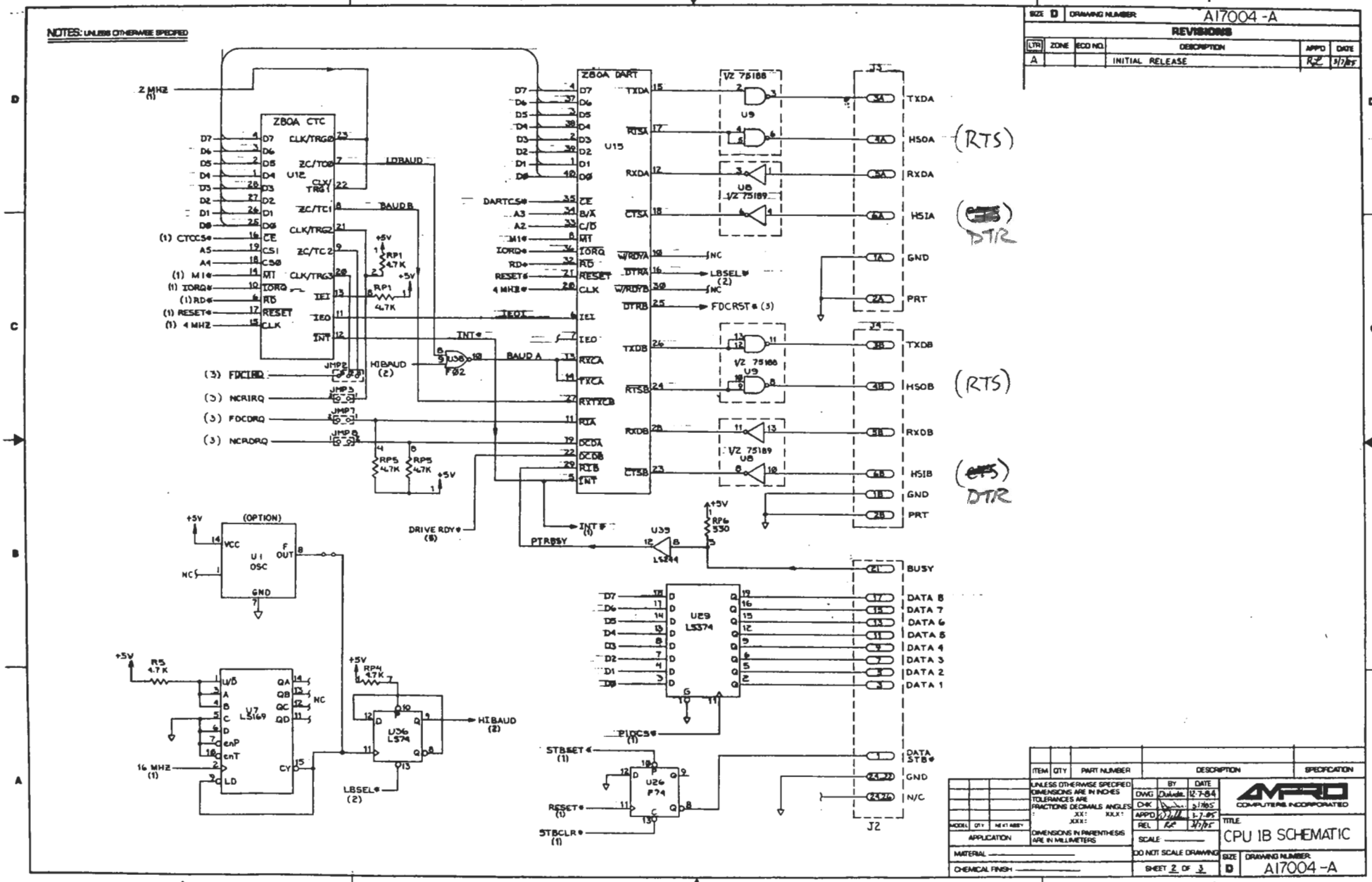
ASSEMBLY: A60060 - Little Board Plus REV: A DATE: 06/19/85

| DESIG. | QTY. | DESCRIPTION | VENDOR P/N | AMPRO P/N |
|---|---|---|---|---|
| A13004 | 1 | PCB, AMPRO SERIES 1B | | A13004 |
| C1-19,21-23, 26,28 | 24 | CAP CER AXIAL .1UF +80% -20% 50V | MRA RPA20-Z5U-104Z50<br>CRN CAC02Z5U104Z050A<br>UNI CGC104ZDZ | 90514-001<br>90514-001<br>90514-001 |
| C20,24,25,27 | 4 | CAP 10UF ELC RADIAL .100-OC 20% 25V | UCK LL25VB10-M<br>ALB LBRIE100S-M | 90522-001<br>90522-001 |
| CR1 | 1 | DIODE 1N4148 | | 90300-001 |
| J1 | 1 | CONN HDR 4POS SIL RT/AG POWER | MLX 8981-4R-1 | 90907-001 |
| J2 | 1 | CONN HDR 26POS .100-OC STR | SAE THD6926WIS | 90907-002 |
| J3,4 | 2 | CONN HDR 6POS SIL RT/AG | MLX 22-05-3061 | 90907-003 |
| J5 | 1 | CONN HDR 4POS SIL RT/AG | MLX 22-05-3041 | 90907-004 |
| J8 | 1 | CONN HDR 50POS .100-OC STR | SAE THD6950WIS | 90907-009 |
| J9 | 1 | CONN HDR 16POS .100-OC STR | SAE THD6916WIS | 90907-007 |
| JMP1,3,6,7,8, 9,10,11 | 8 | CONN HDR 2POS SIL .100-OC | MLX 22-10-2021<br>AMP 641122-2 | 90905-001<br>90905-001 |
| JMP5,2 | 2 | CONN HDR 3POS SIL .100-OC | | 90905-003 |
| JMP1,5,10 | 3 | CONN SHUNT 2POS .100-OC (.40 MAX HEIGHT) | MLX 15-38-1024 | 90905-002<br>90905-002 |
| R1,5,6,7 | 4 | RES CF 4700 5% 1/4W | | 90015-003 |
| R2 | 1 | RES CF 1K 5% 1/4W | | 90015-002 |
| R3,R8 | 2 | RES CF 39 5% 1/4W | | 90015-001 |
| R4 | 1 | RES CF 10K 5% 1/4W | | 90015-007 |
| RP1,4,5,7 | 4 | RES PK 8SIP 7-4700 5% | | 90014-003 |
| RP2,3 | 2 | RES PK 8SIP 7-1K 5% | | 90014-002 |
| RP6 | 1 | RES PK 8SIP 7-330 5% | | 90014-001 |
| U1 | SPARE | | | |
| U2 | 1 | OSCILLATOR, 16.000MHZ | SRX NCT070C16MHZ<br>NDK TD1114A-16.00 | 90824-001<br>90824-001 |
| U3 | 1 | IC, 74F163 OR 74F161 | | 90620-025 |
| U4 | 1 | IC, Z80A CPU | | 90670-002 |
| U5 | 1 | IC, 10L8/PROGRAMMED | | A75507 |
| U6,U35 | 2 | IC, 74F00 | | 90620-019 |
| U7 | 1 | IC, 74LS169 | ** NO TI | 90620-012 |
| U8 | 1 | IC, 75189A/1489A | | 90660-002 |
| U9 | 1 | IC, 75188/1488 | | 90660-001 |
| U10 | NOT USED | | | |
| U11 | 1 | HYBRID, -12VDC/DC CONVERTER | ELPAC/TDK CB3811 | 90702-001 |
| U12 | 1 | IC, Z80A CTC | | 90670-003 |
| U13 | 1 | IC, 2732 450nS (PROGRAMMED) | | A75508 |
| U14 | 1 | IC, WD1770 DISK CONTROLLER | WESTERN DIGITAL / TI | 90670-001 |
| U15 | 1 | IC, Z80A SIO/0 or Z80A DART | | 90670-008 |

ASSEMBLY: A60060 - Little Board Plus REV: A DATE: 06/19/85

| DESIG. | QTY. | DESCRIPTION | VENDOR P/N | AMPRO P/N |
|---|---|---|---|---|
| U16 | 1 | IC, NCR 5380 BUS CONTROLLER | | 90670-005 |
| U17,18 | 2 | RES N/W DIP 12 - 220/330 | | 90014-004 |
| U19-21,30-34 | 8 | IC, 4164 200nS 64K DYN RAM | ** NO TI | 90680-002 |
| U22,23 | 2 | IC, 74LS157 | | 90620-010 |
| U24 | 1 | IC, 74F32 | | 90620-016 |
| U25 | 1 | IC, 7406 | | 90620-003 |
| U26,37 | 2 | IC, 74F74 | | 90620-021 |
| U27 | 1 | IC, 74LS139 | | 90620-009 |
| U28 | 1 | IC, 7438 | | 90620-018 |
| U29 | 1 | IC, 74LS374 | | 90620-015 |
| U36 | 1 | IC, 74LS74 | | 90620-007 |
| U38 | 1 | IC, 74F02 | | 90620-026 |
| U39,41 | 2 | IC, 74LS244 | | 90620-013 |
| U40 | 1 | IC, 74LS273 | | 90620-014 |
| U4,15,16 | 3 | IC SOCKET 40 POS DUAL WIPE | JNE J23-5040 | 90800-002 |
| | | | AMP 2-640379-3 | 90800-002 |
| U13,14 | 2 | IC SOCKET 28 POS DUAL WIPE | JNE J23-5028 | 90800-004 |
| U17,18 | 2 | IC SOCKET 14 POS DUAL WIPE | JNE J23-5014 | 90800-003 |


NOTES: UNLESS OTHERWISE SPECIFIED
SIZE D DRAWING NUMBER A17004-A
REVISIONS
LTR ZONE ECO NO. DESCRIPTION APP'D DATE
INITIAL RELEASE
SEE SHEET 2 FOR REVISIONS
CONTROL REG.
U4 Z80A CPU
I/O DECODE
U5 10L8
U13 EPROM 28-PIN SOCKET
64 K RAM ARRAY
EPROM JUMPER CONFIGURATION
2732 2764 27128 27256
JMP5: 1-2 OPEN N/C SHORT SHORT
JMP5: 2-3 SHORT N/C OPEN OPEN
JMP6: 1-2 N/C OPEN OPEN SHORT
ITEM QTY PART NUMBER DESCRIPTION SPECIFICATION
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
APPLICATION
MATERIAL
CHEMICAL FINISH
DO NOT SCALE DRAWING
SHEET 1 OF 3
COMPUTERS INCORPORATED
TITLE: CPU 1B SCHEMATIC
SIZE D DRAWING NUMBER A17004-A


NOTES: UNLESS OTHERWISE SPECIFIED
SIZE D DRAWING NUMBER A17004-A
REVISIONS
LTR ZONE ECO NO. DESCRIPTION APPD DATE
A INITIAL RELEASE RZ 9/7/84
Z80A CTC
Z80A DART
U15
U12
U1 OSC
U7 LS169
U36 LS74
U26 F74
U29 LS374
U39 LS244
VZ 75188
VZ 75189
U9
U8
J3
J2
2 MHZ (1)
LDBAUD
BAUDB
BAUD A
HIBAUD (2)
INT#
IEO1
(1) CTCCS#
(1) M1#
(1) IORQ#
(1) RD#
(1) RESET#
(1) 4 MHZ
DARTCS#
4 MHZ#
RESET#
(3) FDCIRQ
(3) NCRIRQ
(3) FDCDRQ
(3) NCRDRQ
JMP2
JMP5
JMP7
JMP8
RP1 4.7K
RP5 4.7K
RP4 4.7K
R5 4.7K
RP6 330
+5V
(OPTION)
NC
GND
DRIVE RDY# (6)
PTRBSY
INT# (1)
FDCRST# (3)
LBSEL# (2)
PIOCS# (1)
STBSET# (1)
RESET# (1)
STBCLR# (1)
16 MHZ (1)
TXDA
HSOA
RXDA
HSIA
GND
PRT
TXDB
HSOB
RXDB
HSIB
BUSY
DATA 8
DATA 7
DATA 6
DATA 5
DATA 4
DATA 3
DATA 2
DATA 1
DATA STB#
N/C
(RTS)
DTR
ITEM QTY PART NUMBER DESCRIPTION SPECIFICATION
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES : XX: XXX: .XXX:
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
APPLICATION
MATERIAL
CHEMICAL FINISH
DWG Dukate 12-7-84
CHK 2/1165
APPD 1-7-85
REL RZ 3/7/85
SCALE
DO NOT SCALE DRAWING
SHEET 2 OF 3
AMPRO COMPUTERS INCORPORATED
TITLE CPU 1B SCHEMATIC
SIZE D DRAWING NUMBER A17004-A

NOTES: UNLESS OTHERWISE SPECIFIED

1. NUMBERS IN PARENTHESIS INDICATE SHEET NUMBER OF SCHEMATIC.

| SIZE D | DRAWING NUMBER A17004-A |
|---|---|

**REVISIONS**

| LTR | ZONE | ECO NO. | DESCRIPTION | APPD | DATE |
|---|---|---|---|---|---|
| A | | | INITIAL RELEASE | | |

SEE SHEET 2 FOR REVISIONS

# FLOPPY DISK CONTROLLER

OPTION

(1) 16 MHZ — F00 U6 — (1) FDCMI — (1) 8MHZ

U144 1770 (SOCKET)

D7 D6 D5 D4 D3 D2 D1 D0 — DAL7 DAL6 DAL5 DAL4 DAL3 DAL2 DAL1 DAL0

(1) FDCCS* — CS

A1 A0 A2 — A1 A0 R/W

WG WD MO DIRC STEP — RD IP TRK0 WPRT

DRQ DDEN INTRQ — (1) DDEN* — CLK

U25 7406 — U39 LS244

(1) DS4 (1) DS3 (1) DS2 (1) DS1 — 7438

SIDE 1

RP2 1K RP3 1K RP6 330 — +5V

J6

| | |
|---|---|
| FDX4 | WRITE GATE — |
| FDX2 | WRITE DATA — |
| FDX6 | MOTOR ON — |
| FDX8 | DIRECTION SEL — |
| FDXA | STEP — |
| FDX5 | DRIVE SEL 4 — |
| FDX3 | DRIVE SEL 3 — |
| FDX7 | DRIVE SEL 2 — |
| FDX9 | DRIVE SEL 1 — |
| FDXB | SIDE ONE — |
| FDXC | LOW SPEED — 2 |
| FDX0 | READ DATA — |
| FDX8 | INDEX — |
| FDXA | TRACK 00 — |
| FDXB | WRITE PROTECT — |
| FDXD | READY — 34 |

FDCHI (1) — U39

JMP10 (1) RESET* — JMP11 (2) FDCRST*

JMP4

FDCIRQ (2)

FDCDRQ (2)

DRIVE RDY* (2)

GND — ALL ODD

SCSI/PLUS (TM) BUS CONTROLLER

U16 5380 (SOCKET)

RP7 4.7K — R7 4.7K — J9

U41 LS244 — (1) DRD* — (1) DB0-7

(1) A2 (1) A1 (1) A0

(1) RESET* (1) NCRCS* (1) NCRDACK* (1) IORD* (1) IOWR*

RP5 4.7K

RESET CS DACK IOR IOW EOP READY DRQ IRQ

U18 220 330

J8

| | |
|---|---|
| 2 | DB0* |
| 4 | DB1* |
| 6 | DB2* |
| 8 | DB3* |
| 10 | DB4* |
| 12 | DB5* |
| 14 | DB6* |
| 16 | DB7* |
| 18 | DBP* |
| 32 | ATN* |
| 36 | BSY* |
| 38 | ACK* |
| 40 | RST* |
| 44 | SEL* |
| 42 | MSG* |
| 46 | C/D* |
| 48 | REQ* |
| 50 | I/O* |

JMP9 — J8

| | |
|---|---|
| 26 | TERMPWR |
| 25 | N/C |
| | GROUND |

20,22,24,28,30,34, ALL ODD (1-49) EXCEPT 25

NCRIRQ (2)

NCRDRQ (2)

| REF | GND | +5V | +12V | -12V |
|---|---|---|---|---|
| U1 | | | SPARE | |
| U2 | 7 | 14 | | 2 |
| U3 | 8 | 16 | | |
| U4 | 29 | 11 | | |
| U5 | 10 | 20 | | |
| U6 | 7 | 14 | | |
| U7 | 8 | 16 | | |
| U8 | 7 | 14 | | |
| U9 | 7 | | 14 | 1 |
| U10 | | | | |
| U11 | 3,4 | 1 | | |
| U12 | 5 | 24 | | |
| U13 | 14 | 28 | | |
| U14 | 14 | 15 | | |
| U15 | 31 | 9 | | |
| U16 | 11 | 31 | | |
| U17 | 7 | 14 | | |
| U18 | 7 | 14 | | |
| U19 | 16 | 8 | | |
| U20 | 16 | 8 | | |
| U21 | 16 | 8 | | |
| U22 | 8 | 16 | | |
| U23 | 8 | 16 | | |
| U24 | 7 | 14 | | |
| U25 | 7 | 14 | | |
| U26 | 7 | 14 | | |
| U27 | 8 | 16 | | |
| U28 | 7 | 14 | | |
| U29 | 10 | 20 | | |
| U30 | 16 | 8 | | |
| U31 | 16 | 8 | | |
| U32 | 16 | 8 | | |
| U33 | 16 | 8 | | |
| U34 | 16 | 8 | | |
| U35 | 7 | 14 | | |
| U36 | 7 | 14 | | |
| U37 | 7 | 14 | | |
| U38 | 7 | 14 | | |

| REF | GND | +5V | +12V | -12V |
|---|---|---|---|---|
| U39 | 10 | 20 | | |
| U40 | 10 | 20 | | |
| U41 | 10 | 20 | | |

| ITEM | QTY | PART NUMBER | DESCRIPTION | SPECIFICATION |
|---|---|---|---|---|

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. TOLERANCES ARE: FRACTIONS DECIMALS ANGLES; .XX ±.01 XX.X ±1°; .XXX ±.005. DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS.

| | BY | DATE |
|---|---|---|
| DWG | Dukede | 12-7-84 |
| CHK | | 5-7-85 |
| APPD | | 1-7-85 |
| REL | | 3/1/85 |

APPLICATION — NEXT ASSY — MODEL — QTY

MATERIAL

CHEMICAL FINISH

SCALE — DO NOT SCALE DRAWING — SHEET 3 OF 3

AMPRO COMPUTERS, INCORPORATED

TITLE: CPU 1B SCHEMATIC

SIZE D — DRAWING NUMBER A17004-A

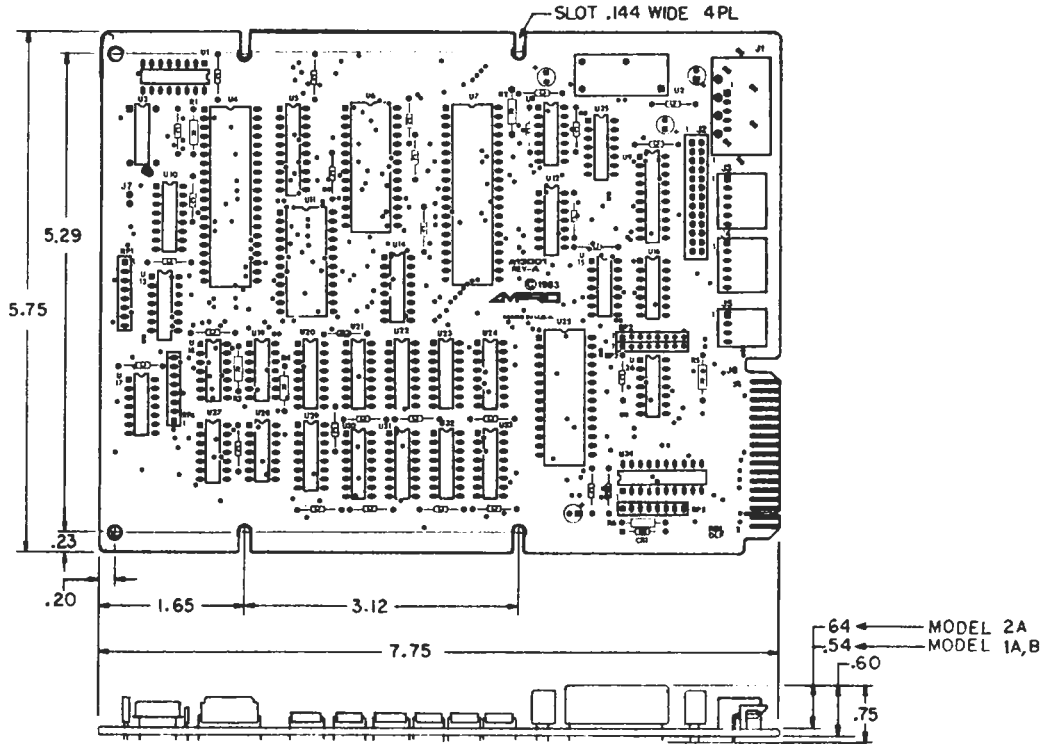
| SIZE C | DRAWING NUMBER: A17002-A |
|---|---|

**REVISIONS**

| LTR | ZONE | ECO NO | DESCRIPTION | APP'D | DATE |
|---|---|---|---|---|---|
| A | | | INITIAL RELEASE | | 2-1-84 |

**NOTES:** UNLESS OTHERWISE SPECIFIED

1. COMPONENT LAYOUT SHOWN IS CPU 1A.
2. MOUNTING DIMENSIONS ARE THE SAME FOR ALL MODELS.
3. J1,3,4,5,6 LOCATION IS THE SAME.




| ITEM | QTY | PART NUMBER | DESCRIPTION | SPECIFICATION |
|---|---|---|---|---|
| | | | | |

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ARE:
FRACTIONS DECIMALS ANGLES
± .XX±.02 XX X± – XXX±.010
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS.

| | BY | DATE |
|---|---|---|
| DWG | DJ | 2-7-84 |
| CHK | | |
| APP'D | | 2-8-84 |
| REL | | |

SCALE: 1 X
DO NOT SCALE DRAWING
SHEET 1 OF 1

MODEL | QTY | NEXT ASSY
APPLICATION
MATERIAL:
CHEMICAL FINISH:

AMPRO COMPUTERS INCORPORATED

TITLE: CPU MOUNTING

SIZE C | DRAWING NUMBER: A17002-A

## APPENDIX B

## TYPICAL INTERFACE CABLES

This Appendix contains wiring information for connection of the two Little Board/PLUS serial ports to typical terminals, modems, and serial printers. In the tables, signal directions are relative to Little Board/PLUS.

### TERMINAL CABLE

Table B-1 lists the pin connections generally used to connect to a terminal. NOTE: to reduce EMI radiation, the cable should be shielded, with shield connected to the connector shell. Suggested part numbers for the board connector are given in Table 2-2. The terminal connector can be either male or female, depending upon the specific terminal.

**Table B-1. Typical Terminal Cable Wiring**

| Board Connector (J3) | Signal Name | Function | Terminal Connector (DB-25) |
|---|---|---|---|
| 1 | Ground | Protective Ground | 1 |
| 5 | RxD | Data Input | 2 |
| 3 | TxD | Data Output | 3 |
| 4 | HSO | Handshake Signal Out | 5 |
| 2 | Ground | Signal Ground | 7 |
| 6 | HSI | Handshake Signal In | 20 |

### SERIAL PRINTER CABLE

Table B-2 lists the pin connections generally used to connect Serial Port B to a serial printer. NOTE: to reduce EMI radiation, the cable must be shielded, and the shield must be connected to the connector shell. Suggested part numbers for the board connector are given in Table 2-2. The printer connector can be either male or female, depending upon the specific printer.

**Table B-2. Typical Serial Printer Cable Wiring**

| Board Connector (J3) | Signal Name | Function | Printer Connector (DB-25) |
|---|---|---|---|
| 1 | Ground | Protective Ground | 1 |
| 5 | RxD | Data Input | 2 |
| 3 | TxD | Data Output | 3 |
| 4 | HSO | Hand Shake Out | 5 |
| 2 | Ground | Signal Ground | 7 |
| 6 | HSI | Hand Shake In | (11)* |

*** NOTE:**

"Handshake Signal In" must connect to the printer's "Busy" output, i.e., the signal which tells the computer to start/stop sending data to the printer. The specific printer connector pin required for "Handshake Signal In" may vary between printers, so be sure to consult your printer's instruction manual.

## MODEM CABLE

Table B-3 lists the pin connections generally used to connect to a modem. NOTE: to reduce EMI radiation, the cable must be shielded, and the shield connected to the connector shell. The connector for the computer end must be a male DB-25, while the modem connector can be either male or female (usually male), depending upon the specific modem.

**Table B-3. Typical Modem Cable Wiring**

| Board Connector (J3) | Signal Name | Function | Modem Connector (DB-25) |
|---|---|---|---|
| 1 | Ground | Protective Ground | 1 |
| 5 | RxD | Data Input | 3 |
| 3 | TxD | Data Output | 2 |
| 4 | HSO | Hand Shake Out | 20 |
| 2 | Ground | Signal Ground | 7 |
| 6 | HSI | Hand Shake In | 5 |

## APPENDIX C

## CP/M USER GROUPS

**AMPRO1: AMPRO USER'S BULLETIN BOARD**
(408) 258-8128
24 hrs/day, 7 days/week, 300/1200 baud

**AMPRO2: AMPRO USER'S BULLETIN BOARD**
(415) 962-9023
24 hrs/day, 7 days/week, 300/1200 baud

Both boards provide a focal point for dissemination of public domain software, and a place to exchange AMPRO-related hardware and software applications. The AMPRO1 bulletin board is not owned by AMPRO Computers, but is provided and maintained by AMPRO users for the benefit of AMPRO users. The AMPRO2 bulletin board is owned and operated by AMPRO Computers. Additional boards may become operational at any time. We suggest that you call into either of the above boards to check on the status of new boards in your area.

**ZCPR3 BULLETIN BOARD: "Z-NODE CENTRAL"**

(415) 489-9005
24 hrs/day, 7 days/week, 300/1200 baud

Filled with the latest ZCPR3 news, as well as the up-to-date versions of ZCPR3 utilities.

**CPMUG** - The CP/M User Group
1651 Third Avenue
New York, NY 10028

**SIG/M** - Special Interest Group for Microcomputers
P.O. Box 97
Iselin, NJ 08830

**PICONET** (a CP/M user group)
P.O. Box 391566
Mountain View, CA 94039-1566

**New York Amateur Computer Club**
P.O. Box 106
Church Street Station
New York, NY 10008

# APPENDIX D

# COMPONENT DATA SHEETS

# Z8400
# Z80® CPU Central Processing Unit

# Zilog

## Product Specification

September 1983



**Features**

- The instruction set contains 158 instructions. The 78 instructions of the 8080A are included as a subset; 8080A software compatibility is maintained.
- Eight MHz, 6 MHz, 4 MHz and 2.5 MHz clocks for the Z80H, Z80B, Z80A, and Z80 CPU result in rapid instruction execution with consequent high data throughput.
- The extensive instruction set includes string, bit, byte, and word operations. Block searches and block transfers together with indexed and relative addressing result in the most powerful data handling capabilities in the microcomputer industry.
- The Z80 microprocessors and associated family of peripheral controllers are linked by a vectored interrupt system. This system may be daisy-chained to allow implementation of a priority interrupt scheme. Little, if any, additional logic is required for daisy-chaining.
- Duplicate sets of both general-purpose and flag registers are provided, easing the design and operation of system software through single-context switching, background-foreground programming, and single-level interrupt processing. In addition, two 16-bit index registers facilitate program processing of tables and arrays.
- There are three modes of high-speed interrupt processing: 8080 similar, non-Z80 peripheral device, and Z80 Family peripheral with or without daisy chain.
- On-chip dynamic memory refresh counter.

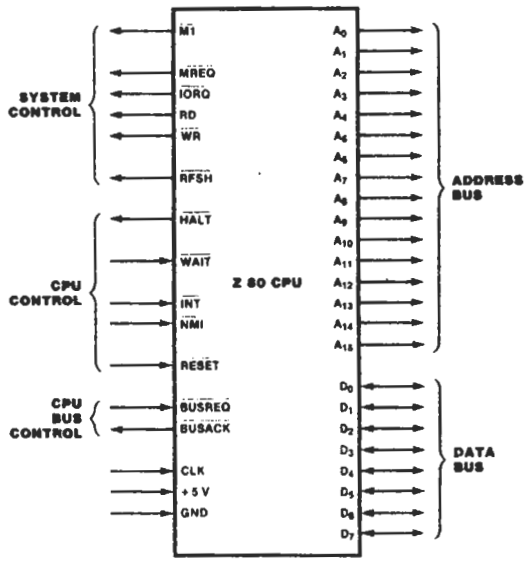



Figure 1. Pin Functions

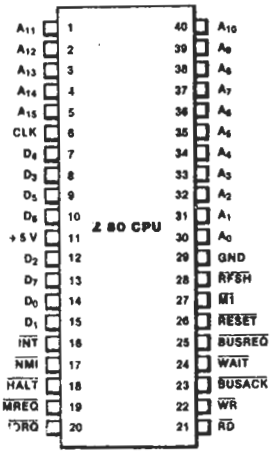



Figure 2. Pin Assignments



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**General Description**

The Z80, Z80A, Z80B, and Z80H CPUs are third-generation single-chip microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second-and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

The Z80 also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. The CPU is easy to incorporate into a system since it requires only a single +5 V power source. All output signals are fully decoded and timed to control standard memory or peripheral circuits, and it is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the Z80 processors. Subsequent text provides more detail on the Z80 I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

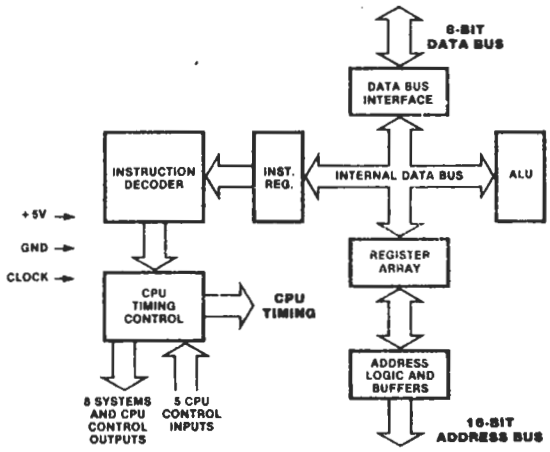



**Figure 3. Z80 CPU Block Diagram**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Z80 Microprocessor Family**

The Zilog Z80 microprocessor is the central element of a comprehensive microprocessor product family. This family works together in most applications with minimum requirements for additional logic, facilitating the design of efficient and cost-effective microcomputer-based systems.

Zilog has designed five components to provide extensive support for the Z80 microprocessor. These are:

- The PIO (Parallel Input/Output) operates in both data-byte I/O transfer mode (with handshaking) and in bit mode (without handshaking). The PIO may be configured to interface with standard parallel peripheral devices such as printers, tape punches, and keyboards.
- The CTC (Counter/Timer Circuit) features four programmable 8-bit counter/timers, each of which has an 8-bit prescaler. Each of the four channels may be configured to operate in either counter or timer mode.
- The DMA (Direct Memory Access) controller provides dual port data transfer operations and the ability to terminate data transfer as a result of a pattern match.
- The SIO (Serial Input/Output) controller offers two channels. It is capable of operating in a variety of programmable modes for both synchronous and asynchronous communication, including Bi-Sync and SDLC.
- The DART (Dual Asynchronous Receiver/Transmitter) device provides low cost asynchronous serial communication. It has two channels and a full modem control interface.

**Z80 CPU Registers**

Figure 4 shows three groups of registers within the Z80 CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set (designated by ' [prime], e.g., A'). Both sets consist of the Accumulator Register, the Flag Register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt Register), the R (Refresh Register), the IX and IY (Index Registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

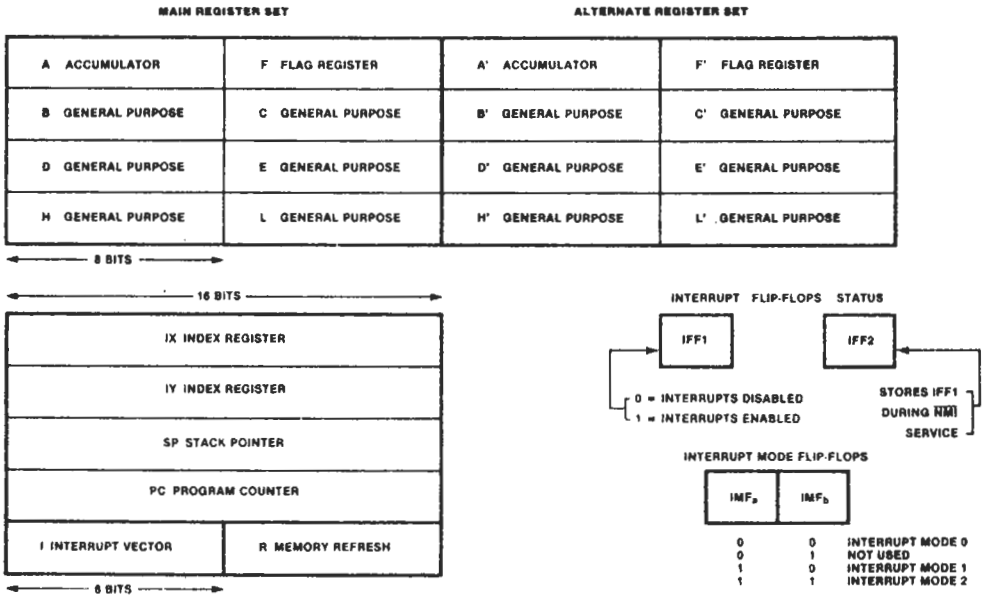



**Figure 4. CPU Registers**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Z80 CPU Registers** (Continued)

| Register | | Size (Bits) | Remarks |
|---|---|---|---|
| A, A' | Accumulator | 8 | Stores an operand or the results of an operation. |
| F, F' | Flags | 8 | See Instruction Set. |
| B, B' | General Purpose | 8 | Can be used separately or as a 16-bit register with C. |
| C, C' | General Purpose | 8 | See B, above. |
| D, D' | General Purpose | 8 | Can be used separately or as a 16-bit register with E. |
| E, E' | General Purpose | 8 | See D, above. |
| H, H' | General Purpose | 8 | Can be used separately or as a 16-bit register with L. |
| L, L' | General Purpose | 8 | See H, above. |
| | | | Note: The (B,C), (D,E), and (H,L) sets are combined as follows:<br>B — High byte C — Low byte<br>D — High byte E — Low byte<br>H — High byte L — Low byte |
| I | Interrupt Register | 8 | Stores upper eight bits of memory address for vectored interrupt processing. |
| R | Refresh Register | 8 | Provides user-transparent dynamic memory refresh. Lower seven bits are automatically incremented and all eight are placed on the address bus during each instruction fetch cycle refresh time. |
| IX | Index Register | 16 | Used for indexed addressing. |
| IY | Index Register | 16 | Same as IX, above. |
| SP | Stack Pointer | 16 | Holds address of the top of the stack. See Push or Pop in instruction set. |
| PC | Program Counter | 16 | Holds address of next instruction. |
| $IFF_1$-$IFF_2$ | Interrupt Enable | Flip-Flops | Set or reset to indicate interrupt status (see Figure 4). |
| IMFa-IMFb | Interrupt Mode | Flip-Flops | Reflect Interrupt mode (see Figure 4). |

**Table 1. Z80 CPU Registers**

**Interrupts: General Operation**

The CPU accepts two interrupt input signals: $\overline{NMI}$ and $\overline{INT}$. The $\overline{NMI}$ is a non-maskable interrupt and has the highest priority. $\overline{INT}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate. $\overline{INT}$ can be connected to multiple peripheral devices in a wired-OR configuration.

The Z80 has a single response mode for interrupt service for the non-maskable interrupt. The maskable interrupt, $\overline{INT}$, has three programmable response modes available. These are:

- Mode 0 — similar to the 8080 microprocessor.
- Mode 1 — Peripheral Interrupt service, for use with non-8080/Z80 systems.
- Mode 2 — a vectored interrupt scheme, usually daisy-chained, for use with Z80 Family and compatible peripheral devices.

The CPU services interrupts by sampling the $\overline{NMI}$ and $\overline{INT}$ signals at the rising edge of the last clock of an instruction. Further interrupt service processing depends upon the type of interrupt that was detected. Details on interrupt responses are shown in the CPU Timing Section.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Interrupts: General Operation** (Continued)

**Non-Maskable Interrupt ($\overline{NMI}$).** The non-maskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routing.

**Maskable Interrupt ($\overline{INT}$).** Regardless of the interrupt mode set by the user, the Z80 response to a maskable interrupt input follows a common timing cycle. After the interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{BUSREQ}$ is not active) a special interrupt processing cycle begins. This is a special fetch ($\overline{M1}$) cycle in which $\overline{IORQ}$ becomes active rather than $\overline{MREQ}$, as in normal $\overline{M1}$ cycle. In addition, this special $\overline{M1}$ cycle is automatically extended by two $\overline{WAIT}$ states, to allow for the time required to acknowledge the interrupt request.

**Mode 0 Interrupt Operation.** This mode is similar to the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.

**Mode 1 Interrupt Operation.** Mode 1 operation is very similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has a restart location of 0038H only.

**Mode 2 Interrupt Operation.** This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8-bits and the contents of the I register as the upper 8-bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 ($A_0$) must be a zero.

**Interrupt Priority (Daisy Chaining and Nested Interrupts).** The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain configuration. Each device in the chain has an interrupt enable input line (IEI) and an interrupt enable output line (IEO), which is fed to the next lower priority device. The first device in the daisy chain has its IEI input hardwired to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

The interrupting device disables its IEO line to the next lower priority peripheral until it has been serviced. After servicing, its IEO line is raised, allowing lower priority peripherals to demand interrupt servicing.

The Z80 CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

**Interrupt Enable/Disable Operation.** Two flip-flops, $IFF_1$ and $IFF_2$, referred to in the register description are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* and *Z80 Assembly Language Manual.*

| Action | $IFF_1$ | $IFF_2$ | Comments |
|---|---|---|---|
| CPU Reset | 0 | 0 | Maskable interrupt $\overline{INT}$ disabled |
| DI instruction execution | 0 | 0 | Maskable interrupt $\overline{INT}$ disabled |
| EI instruction execution | 1 | 1 | Maskable interrupt $\overline{INT}$ enabled |
| LD A,I instruction execution | • | • | $IFF_2$ → Parity flag |
| LD A,R instruction execution | • | • | $IFF_2$ → Parity flag |
| Accept $\overline{NMI}$ | 0 | $IFF_1$ | $IFF_1$ → $IFF_2$ (Maskable interrupt $\overline{INT}$ disabled) |
| RETN instruction execution | $IFF_2$ | • | $IFF_2$ → $IFF_1$ at completion of an $\overline{NMI}$ service routine. |

**Table 2. State of Flip-Flops**





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Instruction Set

The Z80 microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor. It includes such unique operations as a block move for fast, efficient data transfers within memory or between memory and I/O. It also allows operations on any bit in any location in memory.

The following is a summary of the Z80 instruction set and shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. The *Z80 CPU Technical Manual* (03-0029-01) and *Assembly Language Programming Manual* (03-0002-01) contain significantly more details for programming use.

The instructions are divided into the following categories:

- ☐ 8-bit loads
- ☐ 16-bit loads
- ☐ Exchanges, block transfers, and searches
- ☐ 8-bit arithmetic and logic operations
- ☐ General-purpose arithmetic and CPU control
- ☐ 16-bit arithmetic operations
- ☐ Rotates and shifts
- ☐ Bit set, reset, and test operations
- ☐ Jumps
- ☐ Calls, returns, and restarts
- ☐ Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

- ☐ Immediate
- ☐ Immediate extended
- ☐ Modified page zero
- ☐ Relative
- ☐ Extended
- ☐ Indexed
- ☐ Register
- ☐ Register indirect
- ☐ Implied
- ☐ Bit

## 8-Bit Load Group

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments r, r' | Reg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LD r, r' | r ← r' | • | • | X | • | X | • | • | • | 01 r r' | | 1 | 1 | 4 | 000 | B |
| LD r, n | r ← n | • | • | X | • | X | • | • | • | 00 r 110<br>← n → | | 2 | 2 | 7 | 001 | C |
| LD r, (HL) | r ← (HL) | • | • | X | • | X | • | • | • | 01 r 110 | | 1 | 2 | 7 | 010 | D |
| LD r, (IX + d) | r ← (IX + d) | • | • | X | • | X | • | • | • | 11 011 101<br>01 r 101<br>← d → | DD | 3 | 5 | 19 | 011 | E |
| LD r, (IY + d) | r ← (IY + d) | • | • | X | • | X | • | • | • | 11 111 101<br>01 r 110<br>← d → | FD | 3 | 5 | 19 | 100 | H |
| LD (HL), r | (HL) ← r | • | • | X | • | X | • | • | • | 01 110 r | | 1 | 2 | 7 | 101 | L |
| LD (IX + d), r | (IX + d) ← r | • | • | X | • | X | • | • | • | 11 011 101<br>01 110 r<br>← d → | DD | 3 | 5 | 19 | 111 | A |
| LD (IY + d), r | (IY + d) ← r | • | • | X | • | X | • | • | • | 11 111 101<br>01 110 r<br>← d → | FD | 3 | 5 | 19 | | |
| LD (HL), n | (HL) ← n | • | • | X | • | X | • | • | • | 00 110 110<br>← n → | 36 | 2 | 3 | 10 | | |
| LD (IX + d), n | (IX + d) ← n | • | • | X | • | X | • | • | • | 11 011 101<br>00 110 110<br>← d →<br>← n → | DD<br>36 | 4 | 5 | 19 | | |
| LD (IY + d), n | (IY + d) ← n | • | • | X | • | X | • | • | • | 11 111 101<br>00 110 110<br>← d →<br>← n → | FD<br>36 | 4 | 5 | 19 | | |
| LD A, (BC) | A ← (BC) | • | • | X | • | X | • | • | • | 00 001 010 | 0A | 1 | 2 | 7 | | |
| LD A, (DE) | A ← (DE) | • | • | X | • | X | • | • | • | 00 011 010 | 1A | 1 | 2 | 7 | | |
| LD A, (nn) | A ← (nn) | • | • | X | • | X | • | • | • | 00 111 010<br>← n →<br>← n → | 3A | 3 | 4 | 13 | | |
| LD (BC), A | (BC) ← A | • | • | X | • | X | • | • | • | 00 000 010 | 02 | 1 | 2 | 7 | | |
| LD (DE), A | (DE) ← A | • | • | X | • | X | • | • | • | 00 010 010 | 12 | 1 | 2 | 7 | | |
| LD (nn), A | (nn) ← A | • | • | X | • | X | • | • | • | 00 110 010<br>← n →<br>← n → | 32 | 3 | 4 | 13 | | |
| LD A, I | A ← I | ↕ | ↕ | X | 0 | X | IFF | 0 | • | 11 101 101<br>01 010 111 | ED<br>57 | 2 | 2 | 9 | | |
| LD A, R | A ← R | ↕ | ↕ | X | 0 | X | IFF | 0 | • | 11 101 101<br>01 011 111 | ED<br>5F | 2 | 2 | 9 | | |
| LD I, A | I ← A | • | • | X | • | X | • | • | • | 11 101 101<br>01 000 111 | ED<br>47 | 2 | 2 | 9 | | |
| LD R, A | R ← A | • | • | X | • | X | • | • | • | 11 101 101<br>01 001 111 | ED<br>4F | 2 | 2 | 9 | | |

NOTES: r, r' means any of the registers A, B, C, D, E, H, L
IFF the content of the interrupt enable flip-flop, (IFF) is copied into the P/V flag.
For an explanation of flag notation and symbols for mnemonic tables, see Symbolic Notation section following tables.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## 16-Bit Load Group

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LD dd, nn | dd ← nn | • | • | X | X | • | • | • | 00 dd0 001<br>← n →<br>← n → | 3 | 3 | 10 | dd Pair<br>00 BC<br>01 DE<br>10 HL<br>11 SP |
| LD IX, nn | IX ← nn | • | • | X | X | • | • | • | 11 011 101 DD<br>00 100 001 21<br>← n →<br>← n → | 4 | 4 | 14 | |
| LD IY, nn | IY ← nn | • | • | X | X | • | • | • | 11 111 101 FD<br>00 100 001 21<br>← n →<br>← n → | 4 | 4 | 14 | |
| LD HL, (nn) | H ← (nn + 1)<br>L ← (nn) | • | • | X | X | • | • | • | 00 101 010 2A<br>← n →<br>← n → | 3 | 5 | 16 | |
| LD dd, (nn) | $dd_H$ ← (nn + 1)<br>$dd_L$ ← (nn) | • | • | X | X | • | • | • | 11 101 101 ED<br>01 dd1 011<br>← n →<br>← n → | 4 | 6 | 20 | |
| LD IX, (nn) | $IX_H$ ← (nn + 1)<br>$IX_L$ ← (nn) | • | • | X | X | • | • | • | 11 011 101 DD<br>00 101 010 2A<br>← n →<br>← n → | 4 | 6 | 20 | |
| LD IY, (nn) | $IY_H$ ← (nn + 1)<br>$IY_L$ ← (nn) | • | • | X | X | • | • | • | 11 111 101 FD<br>00 101 010 2A<br>← n →<br>← n → | 4 | 6 | 20 | |
| LD (nn), HL | (nn + 1) ← H<br>(nn) ← L | • | • | X | X | • | • | • | 00 100 010 22<br>← n →<br>← n → | 3 | 5 | 16 | |
| LD (nn), dd | (nn + 1) ← $dd_H$<br>(nn) ← $dd_L$ | • | • | X | X | • | • | • | 11 101 101 ED<br>01 dd0 011<br>← n →<br>← n → | 4 | 6 | 20 | |
| LD (nn), IX | (nn + 1) ← $IX_H$<br>(nn) ← $IX_L$ | • | • | X | X | • | • | • | 11 011 101 DD<br>00 100 010 22<br>← n →<br>← n → | 4 | 6 | 20 | |
| LD (nn), IY | (nn + 1) ← $IY_H$<br>(nn) ← $IY_L$ | • | • | X | X | • | • | • | 11 111 101 FD<br>00 100 010 22<br>← n →<br>← n → | 4 | 6 | 20 | |
| LD SP, HL | SP ← HL | • | • | X | X | • | • | • | 11 111 001 F9 | 1 | 1 | 6 | |
| LD SP, IX | SP ← IX | • | • | X | X | • | • | • | 11 011 101 DD<br>11 111 001 F9 | 2 | 2 | 10 | |
| LD SP, IY | SP ← IY | • | • | X | X | • | • | • | 11 111 101 FD<br>11 111 001 F9 | 2 | 2 | 10 | |
| PUSH qq | (SP - 2) ← $qq_L$<br>(SP - 1) ← $qq_H$<br>SP ← SP - 2 | • | • | X | X | • | • | • | 11 qq0 101 | 1 | 3 | 11 | qq Pair<br>00 BC<br>01 DE<br>10 HL<br>11 AF |
| PUSH IX | (SP - 2) ← $IX_L$<br>(SP - 1) ← $IX_H$<br>SP ← SP - 2 | • | • | X | X | • | • | • | 11 011 101 DD<br>11 100 101 E5 | 2 | 4 | 15 | |
| PUSH IY | (SP - 2) ← $IY_L$<br>(SP - 1) ← $IY_H$<br>SP ← SP - 2 | • | • | X | X | • | • | • | 11 111 101 FD<br>11 100 101 E5 | 2 | 4 | 15 | |
| POP qq | $qq_H$ ← (SP + 1)<br>$qq_L$ ← (SP)<br>SP ← SP + 2 | • | • | X | X | • | • | • | 11 qq0 001 | 1 | 3 | 10 | |
| POP IX | $IX_H$ ← (SP + 1)<br>$IX_L$ ← (SP)<br>SP ← SP + 2 | • | • | X | X | • | • | • | 11 011 101 DD<br>11 100 001 E1 | 2 | 4 | 14 | |
| POP IY | $IY_H$ ← (SP + 1)<br>$IY_L$ ← (SP)<br>SP ← SP + 2 | • | • | X | X | • | • | • | 11 111 101 FD<br>11 100 001 E1 | 2 | 4 | 14 | |

NOTES: dd is any of the register pairs BC, DE, HL, SP.
qq is any of the register pairs AF, BC, DE, HL.
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively,
e.g., $BC_L$ = C, $AF_H$ = A.

## Exchange, Block Transfer, Block Search Groups

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EX DE, HL | DE ↔ HL | • | • | X | X | • | • | • | 11 101 011 EB | 1 | 1 | 4 | |
| EX AF, AF' | AF ↔ AF' | • | • | X | X | • | • | • | 00 001 000 08 | 1 | 1 | 4 | |
| EXX | BC ↔ BC'<br>DE ↔ DE'<br>HL ↔ HL' | • | • | X | X | • | • | • | 11 011 001 D9 | 1 | 1 | 4 | Register bank and auxiliary register bank exchange |
| EX (SP), HL | H ↔ (SP + 1)<br>L ↔ (SP) | • | • | X | X | • | • | • | 11 100 011 E3 | 1 | 5 | 19 | |
| EX (SP), IX | $IX_H$ ↔ (SP + 1)<br>$IX_L$ ↔ (SP) | • | • | X | X | • | • | • | 11 011 101 DD<br>11 100 011 E3 | 2 | 6 | 23 | |
| EX (SP), IY | $IY_H$ ↔ (SP + 1)<br>$IY_L$ ↔ (SP) | • | • | X | X | • | • | • | 11 111 101 FD<br>11 100 011 E3 | 2 | 6 | 23 | |
| LDI | (DE) ← (HL)<br>DE ← DE + 1<br>HL ← HL + 1<br>BC ← BC - 1 | • | • | X | 0 | X | ① 1 | 0 | • | 11 101 101 ED<br>10 100 000 A0 | 2 | 4 | 16 | Load (HL) into (DE), increment the pointers and decrement the byte counter (BC) |
| LDIR | (DE) ← (HL)<br>DE ← DE + 1<br>HL ← HL + 1<br>BC ← BC - 1<br>Repeat until<br>BC = 0 | • | • | X | 0 | X | ① 0 | 0 | • | 11 101 101 ED<br>10 110 000 B0 | 2<br>2 | 5<br>4 | 21<br>16 | If BC ≠ 0<br>If BC = 0 |

NOTE: ① P/V flag is 0 if the result of BC - 1 = 0, otherwise P/V = 1.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Exchange, Block Transfer, Block Search Groups (Continued)

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ① | | | | | | | |
| LDD | (DE) ← (HL)<br>DE ← DE − 1<br>HL ← HL − 1<br>BC ← BC − 1 | • | • | X 0 | X | ↕ | 0 | • | 11 101 101 ED<br>10 101 000 A8 | 2 | 4 | 16 | |
| | | | | | ② | | | | | | | |
| LDDR | (DE) ← (HL)<br>DE ← DE − 1<br>HL ← HL − 1<br>BC ← BC − 1<br>Repeat until<br>BC = 0 | • | • | X 0 | X | 0 | 0 | • | 11 101 101 ED<br>10 111 000 B8 | 2<br>2 | 5<br>4 | 21<br>16 | If BC ≠ 0<br>If BC = 0 |
| | | | ③ | | ① | | | | | | | |
| CPI | A − (HL)<br>HL ← HL + 1<br>BC ← BC − 1 | ↕ | ↕ | X ↕ | X | ↕ | 1 | • | 11 101 101 ED<br>10 100 001 A1 | 2 | 4 | 16 | |
| | | | ③ | | ① | | | | | | | |
| CPIR | A − (HL) | ↕ | ↕ | X ↕ | X | ↕ | 1 | • | 11 101 101 ED | 2 | 5 | 21 | If BC ≠ 0 and A ≠ (HL) |
| | HL ← HL + 1<br>BC ← BC − 1<br>Repeat until<br>A = (HL) or<br>BC = 0 | | | | | | | | 10 110 001 B1 | 2 | 4 | 16 | If BC = 0 or A = (HL) |
| | | | ③ | | ① | | | | | | | |
| CPD | A − (HL)<br>HL ← HL − 1<br>BC ← BC − 1 | ↕ | ↕ | X ↕ | X | ↕ | 1 | • | 11 101 101 ED<br>10 101 001 A9 | 2 | 4 | 16 | |
| | | | ③ | | ① | | | | | | | |
| CPDR | A − (HL) | ↕ | ↕ | X ↕ | X | ↕ | 1 | • | 11 101 101 ED | 2 | 5 | 21 | If BC ≠ 0 and A ≠ (HL) |
| | HL ← HL − 1<br>BC ← BC − 1<br>Repeat until<br>A = (HL) or<br>BC = 0 | | | | | | | | 10 111 001 B9 | 2 | 4 | 16 | If BC = 0 or A = (HL) |

NOTES: ① P/V flag is 0 if the result of BC − 1 = 0, otherwise P/V = 1.
② P/V flag is 0 at completion of instruction only.
③ Z flag is 1 if A = (HL), otherwise Z = 0.

## 8-Bit Arithmetic and Logical Group

| | | S | Z | H | P/V | N | C | Opcode 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD A, r | A ← A + r | ↕ | ↕ | X ↕ | X | V | 0 | ↕ | 10 000 r | 1 | 1 | 4 | r Reg.<br>000 B<br>001 C<br>010 D<br>011 E<br>100 H<br>101 L<br>111 A |
| ADD A, n | A ← A + n | ↕ | ↕ | X ↕ | X | V | 0 | ↕ | 11 000 110<br>← n → | 2 | 2 | 7 | |
| ADD A, (HL) | A ← A + (HL) | ↕ | ↕ | X ↕ | X | V | 0 | ↕ | 10 000 110 | 1 | 2 | 7 | |
| ADD A, (IX + d) | A ← A + (IX + d) | ↕ | ↕ | X ↕ | X | V | 0 | ↕ | 11 011 101 DD<br>10 000 110<br>← d → | 3 | 5 | 19 | |
| ADD A, (IY + d) | A ← A + (IY + d) | ↕ | ↕ | X ↕ | X | V | 0 | ↕ | 11 111 101 FD<br>10 000 110<br>← d → | 3 | 5 | 19 | |
| ADC A, s | A ← A + s + CY | ↕ | ↕ | X ↕ | X | V | 0 | ↕ | 001 | | | | s is any of r, n, (HL), (IX + d), (IY + d) as shown for ADD instruction. The indicated bits replace the 000 in the ADD set above. |
| SUB s | A ← A − s | ↕ | ↕ | X ↕ | X | V | 1 | ↕ | 010 | | | | |
| SBC A, s | A ← A − s − CY | ↕ | ↕ | X ↕ | X | V | 1 | ↕ | 011 | | | | |
| AND s | A ← A ∧ s | ↕ | ↕ | X 1 | X | P | 0 | 0 | 100 | | | | |
| OR s | A ← A ∨ s | ↕ | ↕ | X 0 | X | P | 0 | 0 | 110 | | | | |
| XOR s | A ← A ⊕ s | ↕ | ↕ | X 0 | X | P | 0 | 0 | 101 | | | | |
| CP s | A − s | ↕ | ↕ | X ↕ | X | V | 1 | ↕ | 111 | | | | |
| INC r | r ← r + 1 | ↕ | ↕ | X ↕ | X | V | 0 | • | 00 r 100 | 1 | 1 | 4 | |
| INC (HL) | (HL) ← (HL) + 1 | ↕ | ↕ | X ↕ | X | V | 0 | • | 00 110 100 | 1 | 3 | 11 | |
| INC (IX + d) | (IX + d) ← (IX + d) + 1 | ↕ | ↕ | X ↕ | X | V | 0 | • | 11 011 101 DD<br>00 110 100<br>← d → | 3 | 6 | 23 | |
| INC (IY + d) | (IY + d) ← (IY + d) + 1 | ↕ | ↕ | X ↕ | X | V | 0 | • | 11 111 101 FD<br>00 110 100<br>← d → | 3 | 6 | 23 | |
| DEC m | m ← m − 1 | ↕ | ↕ | X ↕ | X | V | 1 | • | 101 | | | | m is any of r, (HL), (IX + d), (IY + d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode. |



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## General-Purpose Arithmetic and CPU Control Groups

| Mnemonic | Symbolic Operation | S | Z | X | H | X | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DAA | Converts acc. content into packed BCD following add or subtract with packed BCD operands. | ↕ | ↕ | X | ↕ | X | P | • | ↕ | 00 100 111 | 27 | 1 | 1 | 4 | Decimal adjust accumulator. |
| CPL | $A \leftarrow \overline{A}$ | • | • | X | 1 | X | • | 1 | • | 00 101 111 | 2F | 1 | 1 | 4 | Complement accumulator (one's complement). |
| NEG | A ← 0 − A | ↕ | ↕ | X | ↕ | X | V | 1 | ↕ | 11 101 101<br>01 000 100 | ED<br>44 | 2 | 2 | 8 | Negate acc. (two's complement). |
| CCF | $CY \leftarrow \overline{CY}$ | • | • | X | X | X | • | 0 | ↕ | 00 111 111 | 3F | 1 | 1 | 4 | Complement carry flag. |
| SCF | CY ← 1 | • | • | X | 0 | X | • | 0 | 1 | 00 110 111 | 37 | 1 | 1 | 4 | Set carry flag. |
| NOP | No operation | • | • | X | • | X | • | • | • | 00 000 000 | 00 | 1 | 1 | 4 | |
| HALT | CPU halted | • | • | X | • | X | • | • | • | 01 110 110 | 76 | 1 | 1 | 4 | |
| DI * | IFF ← 0 | • | • | X | • | X | • | • | • | 11 110 011 | F3 | 1 | 1 | 4 | |
| EI * | IFF ← 1 | • | • | X | • | X | • | • | • | 11 111 011 | FB | 1 | 1 | 4 | |
| IM 0 | Set interrupt mode 0 | • | • | X | • | X | • | • | • | 11 101 101<br>01 000 110 | ED<br>46 | 2 | 2 | 8 | |
| IM 1 | Set interrupt mode 1 | • | • | X | • | X | • | • | • | 11 101 101<br>01 010 110 | ED<br>56 | 2 | 2 | 8 | |
| IM 2 | Set interrupt mode 2 | • | • | X | • | X | • | • | • | 11 101 101<br>01 011 110 | ED<br>5E | 2 | 2 | 8 | |

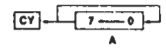
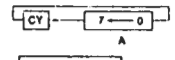
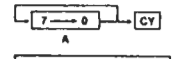
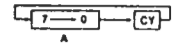
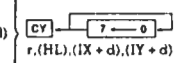
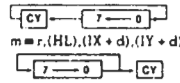
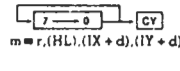
NOTES: IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop.
* indicates interrupts are not sampled at the end of EI or DI.

## 16-Bit Arithmetic Group

| Mnemonic | Symbolic Operation | S | Z | X | H | X | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD HL, ss | HL ← HL + ss | • | • | X | X | X | • | 0 | ↕ | 00 ss1 001 | | 1 | 3 | 11 | ss Reg.<br>00 BC<br>01 DE<br>10 HL<br>11 SP |
| ADC HL, ss | HL ← HL + ss + CY | ↕ | ↕ | X | X | X | V | 0 | ↕ | 11 101 101<br>01 ss1 010 | ED | 2 | 4 | 15 | |
| SBC HL, ss | HL ← HL − ss − CY | ↕ | ↕ | X | X | X | V | 1 | ↕ | 11 101 101<br>01 ss0 010 | ED | 2 | 4 | 15 | |
| ADD IX, pp | IX ← IX + pp | • | • | X | X | X | • | 0 | ↕ | 11 011 101<br>01 pp1 001 | DD | 2 | 4 | 15 | pp Reg.<br>00 BC<br>01 DE<br>10 IX<br>11 SP |
| ADD IY, rr | IY ← IY + rr | • | • | X | X | X | • | 0 | ↕ | 11 111 101<br>00 rr1 001 | FD | 2 | 4 | 15 | rr Reg.<br>00 BC<br>01 DE<br>10 IY<br>11 SP |
| INC ss | ss ← ss + 1 | • | • | X | • | X | • | • | • | 00 ss0 011 | | 1 | 1 | 6 | |
| INC IX | IX ← IX + 1 | • | • | X | • | X | • | • | • | 11 011 101<br>00 100 011 | DD<br>23 | 2 | 2 | 10 | |
| INC IY | IY ← IY + 1 | • | • | X | • | X | • | • | • | 11 111 101<br>00 100 011 | FD<br>23 | 2 | 2 | 10 | |
| DEC ss | ss ← ss − 1 | • | • | X | • | X | • | • | • | 00 ss1 011 | | 1 | 1 | 6 | |
| DEC IX | IX ← IX − 1 | • | • | X | • | X | • | • | • | 11 011 101<br>00 101 011 | DD<br>2B | 2 | 2 | 10 | |
| DEC IY | IY ← IY − 1 | • | • | X | • | X | • | • | • | 11 111 101<br>00 101 011 | FD<br>2B | 2 | 2 | 10 | |

NOTES: ss is any of the register pairs BC, DE, HL, SP.
pp is any of the register pairs BC, DE, IX, SP.
rr is any of the register pairs BC, DE, IY, SP.

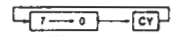
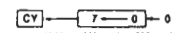
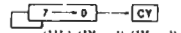
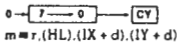
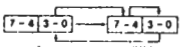
## Rotate and Shift Group

| Mnemonic | Symbolic Operation | S | Z | X | H | X | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RLCA |  | • | • | X | 0 | X | • | 0 | ↕ | 00 000 111 | 07 | 1 | 1 | 4 | Rotate left circular accumulator. |
| RLA |  | • | • | X | 0 | X | • | 0 | ↕ | 00 010 111 | 17 | 1 | 1 | 4 | Rotate left accumulator. |
| RRCA |  | • | • | X | 0 | X | • | 0 | ↕ | 00 001 111 | 0F | 1 | 1 | 4 | Rotate right circular accumulator. |
| RRA |  | • | • | X | 0 | X | • | 0 | ↕ | 00 011 111 | 1F | 1 | 1 | 4 | Rotate right accumulator. |
| RLC r | | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 11 001 011<br>00 000 r | CB | 2 | 2 | 8 | Rotate left circular register r. |
| RLC (HL) | | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 11 001 011<br>00 000 110 | CB | 2 | 4 | 15 | r Reg.<br>000 B<br>001 C<br>010 D<br>011 E<br>100 H<br>101 L<br>111 A |
| RLC (IX + d) |  <br>r,(HL),(IX + d),(IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 11 011 101<br>11 001 011<br>← d →<br>00 000 110 | DD<br>CB | 4 | 6 | 23 | |
| RLC (IY + d) | | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 11 111 101<br>11 001 011<br>← d →<br>00 000 110 | FD<br>CB | 4 | 6 | 23 | |
| RL m |  <br>m = r,(HL),(IX + d),(IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 010 | | | | | Instruction format and states are as shown for RLC's. To form new opcode replace 000 of RLC's with shown code. |
| RRC m |  <br>m = r,(HL),(IX + d),(IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 001 | | | | | |



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Rotate and Shift Group (Continued)

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RR m |  m = r, (HL), (IX + d), (IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 011 | | | | | |
| SLA m |  m = r, (HL), (IX + d), (IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 100 | | | | | |
| SRA m |  m = r, (HL), (IX + d), (IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 101 | | | | | |
| SRL m |  m = r, (HL), (IX + d), (IY + d) | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | 111 | | | | | |
| RLD | | ↕ | ↕ | X | 0 | X | P | 0 | • | 11 101 101<br>01 101 111 | ED<br>6F | 2 | 5 | 18 | Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected. |
| RRD |  | ↕ | ↕ | X | 0 | X | P | 0 | • | 11 101 101<br>01 100 111 | ED<br>67 | 2 | 5 | 18 | |

## Bit Set, Reset and Test Group

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BIT b, r | $Z \leftarrow r_b$ | X | ↕ | X | 1 | X | X | 0 | • | 11 001 011<br>01 b r | CB | 2 | 2 | 8 | r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A<br>b: 000 0, 001 1, 010 2, 011 3, 100 4, 101 5, 110 6, 111 7 |
| BIT b, (HL) | $Z \leftarrow \overline{(HL)_b}$ | X | ↕ | X | 1 | X | X | 0 | • | 11 001 011<br>01 b 110 | CB | 2 | 3 | 12 | |
| BIT b, (IX + d)b | $Z \leftarrow \overline{(IX+d)_b}$ | X | ↕ | X | 1 | X | X | 0 | • | 11 011 101<br>11 001 011<br>← d →<br>01 b 110 | DD<br>CB | 4 | 5 | 20 | |
| BIT b, (IY + d)b | $Z \leftarrow \overline{(IY+d)_b}$ | X | ↕ | X | 1 | X | X | 0 | • | 11 111 101<br>11 001 011<br>← d →<br>01 b 110 | FD<br>CB | 4 | 5 | 20 | |
| SET b, r | $r_b \leftarrow 1$ | • | • | X | • | X | • | • | • | 11 001 011<br>11 b r | CB | 2 | 2 | 8 | |
| SET b, (HL) | $(HL)_b \leftarrow 1$ | • | • | X | • | X | • | • | • | 11 001 011<br>11 b 110 | CB | 2 | 4 | 15 | |
| SET b, (IX + d) | $(IX+d)_b \leftarrow 1$ | • | • | X | • | X | • | • | • | 11 011 101<br>11 001 011<br>← d →<br>11 b 110 | DD<br>CB | 4 | 6 | 23 | |
| SET b, (IY + d) | $(IY+d)_b \leftarrow 1$ | • | • | X | • | X | • | • | • | 11 111 101<br>11 001 011<br>← d →<br>11 b 110 | FD<br>CB | 4 | 6 | 23 | |
| RES b, m | $m_b \leftarrow 0$<br>m = r, (HL), (IX + d), (IY + d) | • | • | X | • | X | • | • | • | 10 | | | | | To form new opcode replace 11 of SET b, s with 10. Flags and time states for SET instruction. |

NOTES: The notation $m_b$ indicates bit b (0 to 7) of location m.

## Jump Group

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 | Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| JP nn | PC ← nn | • | • | X | • | X | • | • | • | 11 000 011<br>← n →<br>← n → | C3 | 3 | 3 | 10 | |
| JP cc, nn | If condition cc is true PC ← nn, otherwise continue | • | • | X | • | X | • | • | • | 11 cc 010<br>← n →<br>← n → | | 3 | 3 | 10 | cc: 000 NZ non-zero, 001 Z zero, 010 NC non-carry, 011 C carry, 100 PO parity odd, 101 PE parity even, 110 P sign positive, 111 M sign negative |
| JR e | PC ← PC + e | • | • | X | • | X | • | • | • | 00 011 000<br>← e−2 → | 18 | 2 | 3 | 12 | |
| JR C, e | If C = 0, continue | • | • | X | • | X | • | • | • | 00 111 000<br>← e−2 → | 38 | 2 | 2 | 7 | If condition not met. |
| | If C = 1, PC ← PC + e | | | | | | | | | | | 2 | 3 | 12 | If condition is met. |
| JR NC, e | If C = 1, continue | • | • | X | • | X | • | • | • | 00 110 000<br>← e−2 → | 30 | 2 | 2 | 7 | If condition not met. |
| | If C = 0, PC ← PC + e | | | | | | | | | | | 2 | 3 | 12 | If condition is met. |
| JP Z, e | If Z = 0 continue | • | • | X | • | X | • | • | • | 00 101 000<br>← e−2 → | 28 | 2 | 2 | 7 | If condition not met. |
| | If Z = 1, PC ← PC + e | | | | | | | | | | | 2 | 3 | 12 | If condition is met. |
| JR NZ, e | If Z = 1, continue | • | • | X | • | X | • | • | • | 00 100 000<br>← e−2 → | 20 | 2 | 2 | 7 | If condition not met. |
| | If Z = 0, PC ← PC + e | | | | | | | | | | | 2 | 3 | 12 | If condition is met. |
| JP (HL) | PC ← HL | • | • | X | • | X | • | • | • | 11 101 001 | E9 | 1 | 1 | 4 | |
| JP (IX) | PC ← IX | • | • | X | • | X | • | • | • | 11 011 101<br>11 101 001 | DD<br>E9 | 2 | 2 | 8 | |



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Jump Group (Continued)

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| JP (IY) | PC ← IY | • | • | X | • | X | • | • | • | 11 111 101 FD<br>11 101 001 E9 | 2 | 2 | 8 | |
| DJNZ, e | B ← B − 1<br>If B = 0, continue | • | • | X | • | X | • | • | • | 00 010 000 10<br>← e − 2 → | 2 | 2 | 8 | If B = 0. |
| | If B ≠ 0, PC ← PC + e | | | | | | | | | | 2 | 3 | 13 | If B ≠ 0. |

NOTES: e represents the extension in the relative addressing mode.
e is a signed two's complement number in the range < −126, 129 >.
e − 2 in the opcode provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

## Call and Return Group

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CALL nn | (SP − 1) ← $PC_H$<br>(SP − 2) ← $PC_L$<br>PC ← nn | • | • | X | • | X | • | • | • | 11 001 101 CD<br>← n →<br>← n → | 3 | 5 | 17 | |
| CALL cc, nn | If condition cc is false continue, otherwise same as CALL nn | • | • | X | • | X | • | • | • | 11 cc 100<br>← n →<br>← n → | 3 | 3 | 10 | If cc is false. |
| | | | | | | | | | | | 3 | 5 | 17 | If cc is true. |
| RET | $PC_L$ ← (SP)<br>$PC_H$ ← (SP + 1) | • | • | X | • | X | • | • | • | 11 001 001 C9 | 1 | 3 | 10 | |
| RET cc | If condition cc is false continue, otherwise same as RET | • | • | X | • | X | • | • | • | 11 cc 000 | 1 | 1 | 5 | If cc is false. |
| | | | | | | | | | | | 1 | 3 | 11 | If cc is true. |
| RETI | Return from interrupt | • | • | X | • | X | • | • | • | 11 101 101 ED<br>01 001 101 4D | 2 | 4 | 14 | |
| RETN[1] | Return from non-maskable interrupt | • | • | X | • | X | • | • | • | 11 101 101 ED<br>01 000 101 45 | 2 | 4 | 14 | |
| RST p | (SP − 1) ← $PC_H$<br>(SP − 2) ← $PC_L$<br>$PC_H$ ← 0<br>$PC_L$ ← p | • | • | X | • | X | • | • | • | 11 t 111 | 1 | 3 | 11 | |

| cc | Condition |
|---|---|
| 000 | NZ non-zero |
| 001 | Z zero |
| 010 | NC non-carry |
| 011 | C carry |
| 100 | PO parity odd |
| 101 | PE parity even |
| 110 | P sign positive |
| 111 | M sign negative |

| t | p |
|---|---|
| 000 | 00H |
| 001 | 08H |
| 010 | 10H |
| 011 | 18H |
| 100 | 20H |
| 101 | 28H |
| 110 | 30H |
| 111 | 38H |

NOTE: [1]RETN loads $IFF_2$ → $IFF_1$

## Input and Output Group

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IN A, (n) | A ← (n) | • | • | X | • | X | • | • | • | 11 011 011 DB<br>← n → | 2 | 3 | 11 | n to $A_0$ ~ $A_7$<br>Acc. to $A_8$ ~ $A_{15}$ |
| IN r, (C) | r ← (C)<br>If r = 110 only the flags will be affected | ↕ | ↕ | X | ↕ | X | P | 0 | • | 11 101 101 ED<br>01 r 000 | 2 | 3 | 12 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| INI | (HL) ← (C)<br>B ← B − 1<br>HL ← HL + 1 | X | ↕ ① | X | X | X | X | 1 | X | 11 101 101 ED<br>10 100 010 A2 | 2 | 4 | 16 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| INIR | (HL) ← (C)<br>B ← B − 1<br>HL ← HL + 1<br>Repeat until B = 0 | X | 1 ② | X | X | X | X | 1 | X | 11 101 101 ED<br>10 110 010 B2 | 2 | 5<br>(If B≠0) | 21 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| | | | | | | | | | | | 2 | 4<br>(If B=0) | 16 | |
| IND | (HL) ← (C)<br>B ← B − 1<br>HL ← HL − 1 | X | ↕ ① | X | X | X | X | 1 | X | 11 101 101 ED<br>10 101 010 AA | 2 | 4 | 16 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| INDR | (HL) ← (C)<br>B ← B − 1<br>HL ← HL − 1<br>Repeat until B = 0 | X | 1 ② | X | X | X | X | 1 | X | 11 101 101 ED<br>10 111 010 BA | 2 | 5<br>(If B≠0) | 21 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| | | | | | | | | | | | 2 | 4<br>(If B=0) | 16 | |
| OUT (n), A | (n) ← A | • | • | X | • | X | • | • | • | 11 010 011 D3<br>← n → | 2 | 3 | 11 | n to $A_0$ ~ $A_7$<br>Acc. to $A_8$ ~ $A_{15}$ |
| OUT (C), r | (C) ← r | • | • | X | • | X | • | • | • | 11 101 101 ED<br>01 r 001 | 2 | 3 | 12 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| OUTI | (C) ← (HL)<br>B ← B − 1<br>HL ← HL + 1 | X | ↕ ① | X | X | X | X | 1 | X | 11 101 101 ED<br>10 100 011 A3 | 2 | 4 | 16 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| OTIR | (C) ← (HL)<br>B ← B − 1<br>HL ← HL + 1<br>Repeat until B = 0 | X | 1 ② | X | X | X | X | 1 | X | 11 101 101 ED<br>10 110 011 B3 | 2 | 5<br>(If B≠0) | 21 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |
| | | | | | | | | | | | 2 | 4<br>(If B=0) | 16 | |
| OUTD | (C) ← (HL)<br>B ← B − 1<br>HL ← HL − 1 | X | ↕ ① | X | X | X | X | 1 | X | 11 101 101 ED<br>10 101 011 AB | 2 | 4 | 16 | C to $A_0$ ~ $A_7$<br>B to $A_8$ ~ $A_{15}$ |

NOTE: ① If the result of B − 1 is zero the Z flag is set, otherwise it is reset.
② Z flag is set upon instruction completion only.





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Input and Output Group (Continued)

| Mnemonic | Symbolic Operation | S | Z | H | P/V | N | C | Opcode 76 543 210 Hex | No. of Bytes | No. of M Cycles | No. of T States | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OTDR | (C) ← (HL)<br>B ← B − 1<br>HL ← HL − 1<br>Repeat until B = 0 | X | 1 ① | X | X | X | 1 | 11 101 101 ED<br>10 111 011 | 2 | 5<br>(If B≠0) | 21 | C to $A_0$ – $A_7$<br>B to $A_8$ – $A_{15}$ |
| | | | | | | | | | 2 | 4<br>(If B=0) | 16 | |

NOTE ① Z flag is set upon instruction completion only.

## Summary of Flag Operation

| Instruction | $D_7$ S | Z | | H | | P/V | N | $D_0$ C | Comments |
|---|---|---|---|---|---|---|---|---|---|
| ADD A, s; ADC A, s | ↕ | ↕ | X | ↕ | X | V | 0 | ↕ | 8-bit add or add with carry. |
| SUB s; SBC A, s; CP s; NEG | ↕ | ↕ | X | ↕ | X | V | 1 | ↕ | 8 bit subtract, subtract with carry, compare and negate accumulator. |
| AND s | ↕ | ↕ | X | 1 | X | P | 0 | 0 | Logical operations. |
| OR s, XOR s | ↕ | ↕ | X | 0 | X | P | 0 | 0 | |
| INC s | ↕ | ↕ | X | ↕ | X | V | 0 | • | 8-bit increment. |
| DEC s | ↕ | ↕ | X | ↕ | X | V | 1 | • | 8-bit decrement. |
| ADD DD, ss | • | • | X | X | X | • | 0 | ↕ | 16-bit add. |
| ADC HL, ss | ↕ | ↕ | X | X | X | V | 0 | ↕ | 16-bit add with carry. |
| SBC HL, ss | ↕ | ↕ | X | X | X | V | 1 | ↕ | 16-bit subtract with carry. |
| RLA, RLCA, RRA, RRCA | • | • | X | 0 | X | • | 0 | ↕ | Rotate accumulator. |
| RL m; RLC m; RR m; RRC m; SLA m; SRA m; SRL m | ↕ | ↕ | X | 0 | X | P | 0 | ↕ | Rotate and shift locations. |
| RLD; RRD | ↕ | ↕ | X | 0 | X | P | 0 | • | Rotate digit left and right. |
| DAA | ↕ | ↕ | X | ↕ | X | P | • | ↕ | Decimal adjust accumulator. |
| CPL | • | • | X | 1 | X | • | 1 | • | Complement accumulator. |
| SCF | • | • | X | 0 | X | • | 0 | 1 | Set carry. |
| CCF | • | • | X | X | X | • | 0 | ↕ | Complement carry. |
| IN r (C) | ↕ | ↕ | X | 0 | X | P | 0 | • | Input register indirect. |
| INI, IND, OUTI; OUTD | X | ↕ | X | X | X | X | 1 | • | Block input and output. Z = 0 if B ≠ 0 otherwise Z = 0. |
| INIR; INDR, OTIR; OTDR | X | 1 | X | X | X | X | 1 | • | |
| LDI; LDD | X | X | X | 0 | X | ↕ | 0 | • | Block transfer instructions P/V = 1 if BC ≠ 0, otherwise P/V = 0. |
| LDIR; LDDR | X | X | X | 0 | X | 0 | 0 | • | |
| CPI; CPIR, CPD, CPDR | X | ↕ | X | X | X | ↕ | ↕ | • | Block search instructions. Z = 1 if A = (HL), otherwise Z = 0. P/V = 1 if BC ≠ 0, otherwise P/V = 0. |
| LD A, I, LD A, R | ↕ | ↕ | X | 0 | X | IFF | 0 | • | The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag. |
| BIT b, s | X | ↕ | X | 1 | X | X | 0 | • | The state of bit b of location s is copied into the Z flag. |

## Symbolic Notation

| Symbol | Operation |
|---|---|
| S | Sign flag. S = 1 if the MSB of the result is 1. |
| Z | Zero flag. Z = 1 if the result of the operation is 0. |
| P/V | Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V = 1 if the result of the operation is even, P/V = 0 if result is odd. If P/V holds overflow, P/V = 1 if the result of the operation produced an overflow. |
| H | Half-carry flag. H = 1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator. |
| N | Add/Subtract flag. N = 1 if the previous operation was a subtract. |
| H & N | H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format. |
| C | Carry/Link flag. C = 1 if the operation produced a carry from the MSB of the operand or result. |

| Symbol | Operation |
|---|---|
| ↕ | The flag is affected according to the result of the operation. |
| • | The flag is unchanged by the operation. |
| 0 | The flag is reset by the operation. |
| 1 | The flag is set by the operation. |
| X | The flag is a "don't care." |
| V | P/V flag affected according to the overflow result of the operation. |
| P | P/V flag affected according to the parity result of the operation. |
| r | Any one of the CPU registers A, B, C, D, E, H, L. |
| s | Any 8-bit location for all the addressing modes allowed for the particular instruction. |
| ss | Any 16-bit location for all the addressing modes allowed for that instruction. |
| ii | Any one of the two index registers IX or IY. |
| R | Refresh counter. |
| n | 8-bit value in range < 0, 255 >. |
| nn | 16-bit value in range < 0, 65535 >. |



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Pin Descriptions**

**$A_0$-$A_{15}$.** *Address Bus* (output, active High, 3-state). $A_0$-$A_{15}$ form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

**$\overline{BUSACK}$.** *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ have entered their high-impedance states. The external circuitry can now control these lines.

**$\overline{BUSREQ}$.** *Bus Request* (input, active Low). Bus Request has a higher priority than $\overline{NMI}$ and is always recognized at the end of the current machine cycle. $\overline{BUSREQ}$ forces the CPU address bus, data bus, and control signals $\overline{MREQ}$, $\overline{IORQ}$, RD, and WR to go to a high-impedance state so that other devices can control these lines. $\overline{BUSREQ}$ is normally wire-ORed and requires an external pullup for these applications. Extended $\overline{BUSREQ}$ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

**$D_0$-$D_7$.** *Data Bus* (input/output, active High, 3-state). $D_0$-$D_7$ constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.

**$\overline{HALT}$.** *Halt State* (output, active Low). $\overline{HALT}$ indicates that the CPU has executed a Halt instruction and is awaiting either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

**$\overline{INT}$.** *Interrupt Request* (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. $\overline{INT}$ is normally wire-ORed and requires an external pullup for these applications.

**$\overline{IORQ}$.** *Input/Output Request* (output, active Low, 3-state). $\overline{IORQ}$ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. $\overline{IORQ}$ is also generated concurrently with $\overline{M1}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

**$\overline{M1}$.** *Machine Cycle One* (output, active Low). $\overline{M1}$, together with $\overline{MREQ}$, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. M1, together with IORQ, indicates an interrupt acknowledge cycle.

**$\overline{MREQ}$.** *Memory Request* (output, active Low, 3-state). $\overline{MREQ}$ indicates that the address bus holds a valid address for a memory read or memory write operation.

**$\overline{NMI}$.** *Non-Maskable Interrupt* (input, negative edge-triggered). $\overline{NMI}$ has a higher priority than $\overline{INT}$. $\overline{NMI}$ is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

**$\overline{RD}$.** *Read* (output, active Low, 3-state). $\overline{RD}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

**$\overline{RESET}$.** *Reset* (input, active Low). $\overline{RESET}$ initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that $\overline{RESET}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

**$\overline{RFSH}$.** *Refresh* (output, active Low). $\overline{RFSH}$, together with $\overline{MREQ}$, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

**$\overline{WAIT}$.** *Wait* (input, active Low). $\overline{WAIT}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended $\overline{WAIT}$ periods can prevent the CPU from refreshing dynamic memory properly.

**$\overline{WR}$.** *Write* (output, active Low, 3-state). $\overline{WR}$ indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## CPU Timing

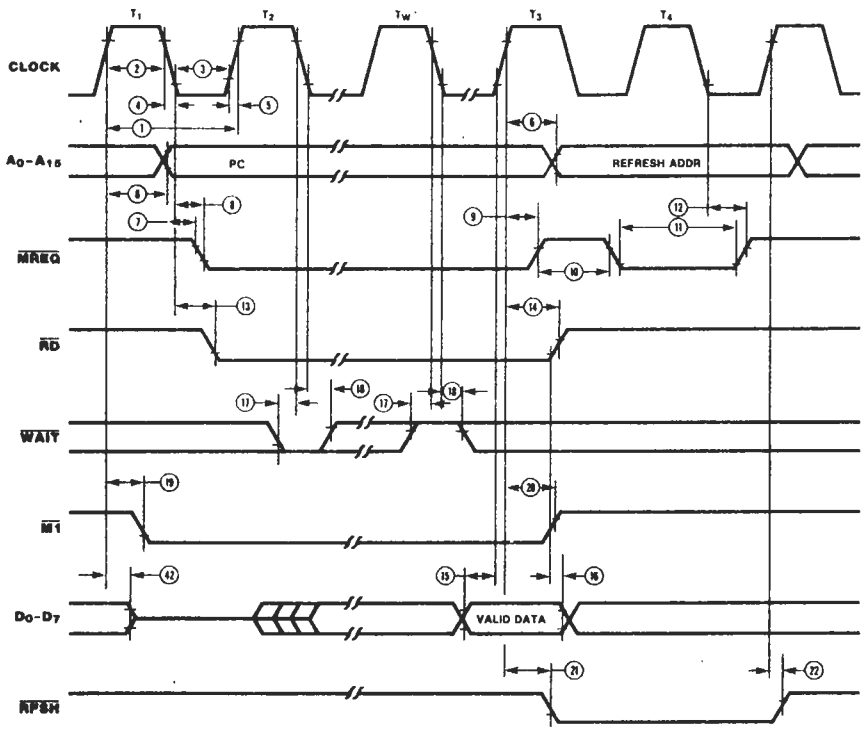
The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle, and three or more T cycles make up a machine cycle (M1, M2 or M3 for instance). Machine cycles can be extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

**Instruction Opcode Fetch.** The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Figure 5). Approximately one-half clock cycle later, $\overline{MREQ}$ goes active. When active, $\overline{RD}$ indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the $\overline{WAIT}$ input with the falling edge of clock state $T_2$. During clock states $T_3$ and $T_4$ of an $\overline{M1}$ cycle dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.




NOTE: $T_W$-Wait cycle added when necessary for slow ancilliary devices.

**Figure 5. Instruction Opcode Fetch**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**CPU Timing** (Continued)

**Memory Read or Write Cycles.** Figure 6 shows the timing of memory read or write cycles other than an opcode fetch ($\overline{M1}$) cycle. The $\overline{MREQ}$ and $\overline{RD}$ signals function exactly as in the fetch cycle. In a memory write cycle, $\overline{MREQ}$ also becomes active when the address bus is stable. The $\overline{WR}$ line is active when the data bus is stable, so that it can be used directly as an R/$\overline{W}$ pulse to most semiconductor memories.

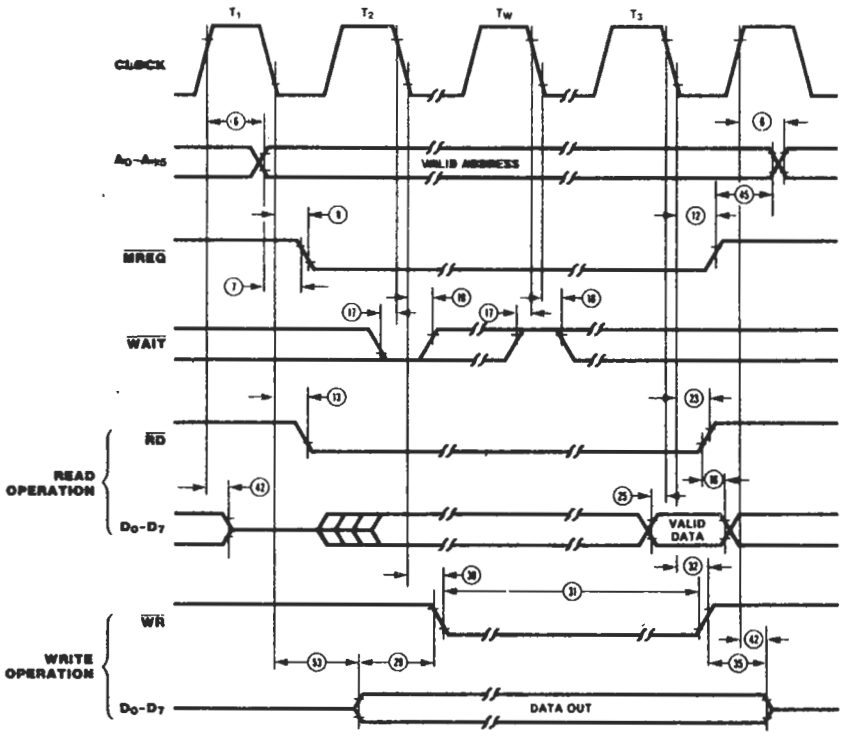


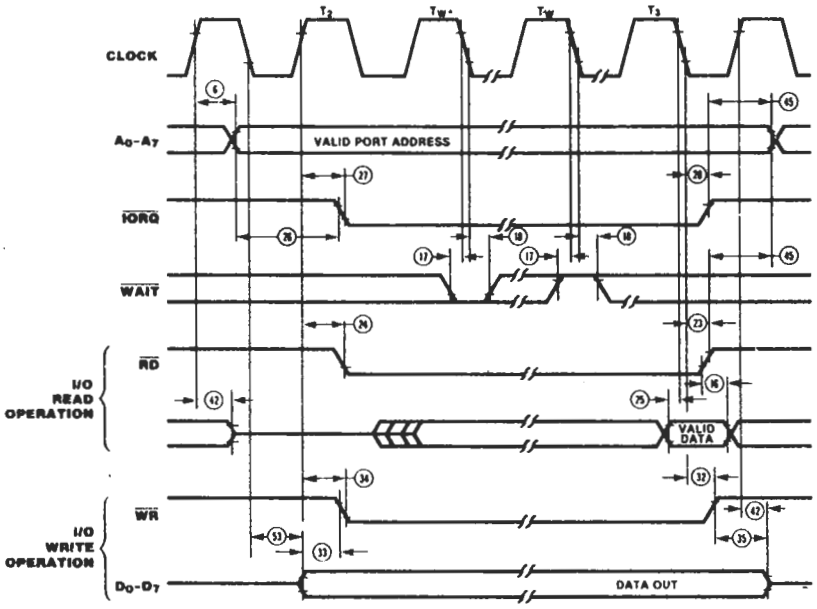

**Figure 6. Memory Read or Write Cycles**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**CPU Timing** (Continued)

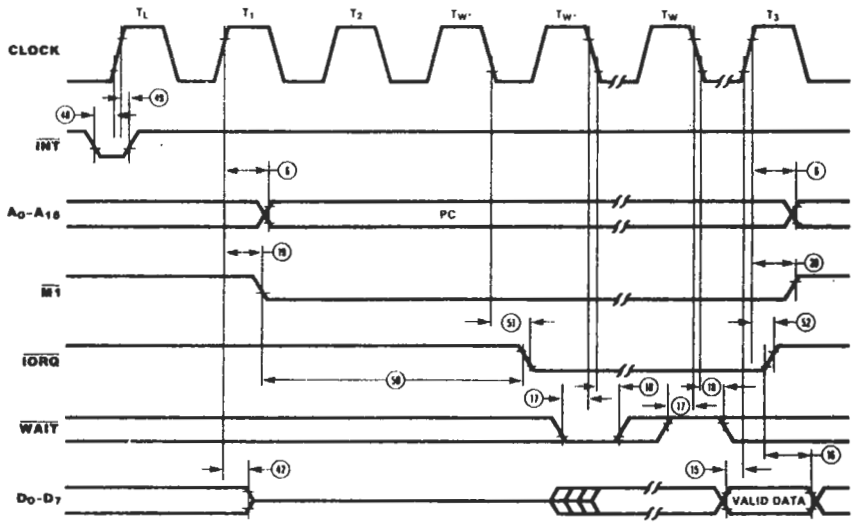
**Input or Output Cycles.** Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state ($T_W$). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.




NOTE: $T_W$* = One Wait cycle automatically inserted by CPU.

**Figure 7. Input or Output Cycles**

**Interrupt Request/Acknowledge Cycle.** The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction (Figure 8). When an interrupt is accepted, a special $\overline{M1}$ cycle is generated. During this $\overline{M1}$ cycle, $\overline{IORQ}$ becomes active (instead of $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.




NOTE. 1) $T_L$ = Last state of previous instruction. 2) Two Wait cycles automatically inserted by CPU(*).

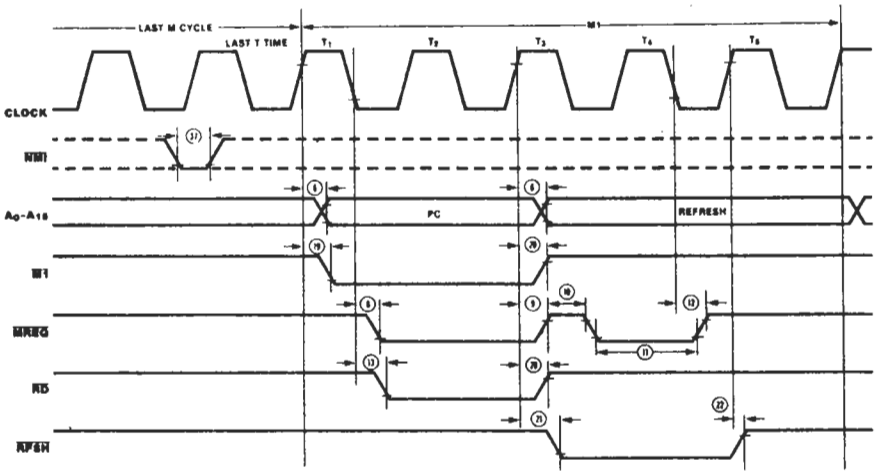
**Figure 8. Interrupt Request/Acknowledge Cycle**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**CPU Timing** (Continued)

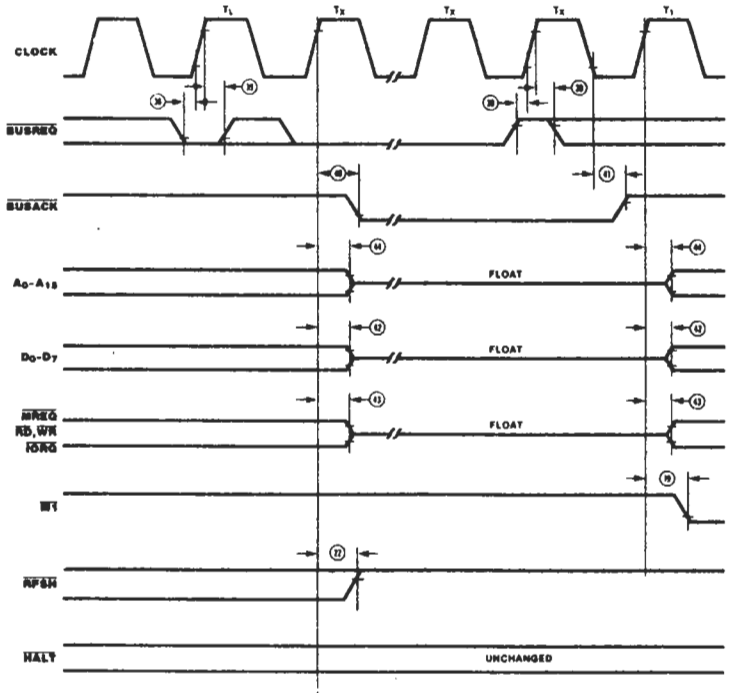
**Non-Maskable Interrupt Request Cycle.** $\overline{NMI}$ is sampled at the same time as the maskable interrupt input $\overline{INT}$ but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal instruction fetch except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the $\overline{NMI}$ service routine located at address 0066H (Figure 9).




*Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding $T_{LAST}$.

**Figure 9. Non-Maskable Interrupt Request Operation**

**Bus Request/Acknowledge Cycle.** The CPU samples $\overline{BUSREQ}$ with the rising edge of the last clock period of any machine cycle (Figure 10). If $\overline{BUSREQ}$ is active, the CPU sets its address, data, and $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ lines to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.




NOTE: $T_L$ = Last state of any M cycle. $T_X$ = An arbitrary clock cycle used by requesting device.

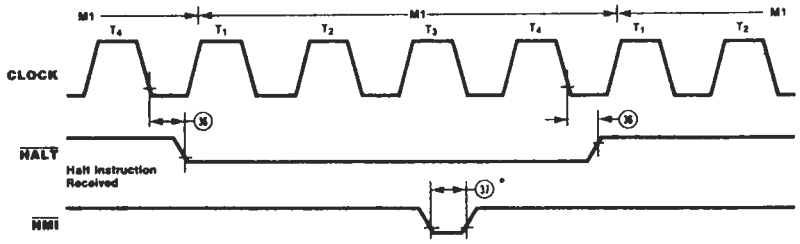
**Figure 10. Z-BUS Request/Acknowledge Cycle**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**CPU Timing** (Continued)

**Halt Acknowledge Cycle.** When the CPU receives a Halt instruction, it executes NOP states until either an $\overline{INT}$ or $\overline{NMI}$ input is received. When in the Halt state, the $\overline{HALT}$ output is active and remains so until an interrupt is received (Figure 11).




NOTE: $\overline{INT}$ will also force a Halt exit.

*See note, Figure 9.

**Figure 11. Halt Acknowledge Cycle**

**Reset Cycle.** $\overline{RESET}$ must be active for at least three clock cycles for the CPU to properly accept it. As long as $\overline{RESET}$ remains active, the address and data buses float, and the control outputs are inactive. Once $\overline{RESET}$ goes inactive, three internal T cycles are consumed before the CPU resumes normal processing operation. $\overline{RESET}$ clears the PC register, so the first opcode fetch will be to location 0000 (Figure 12).

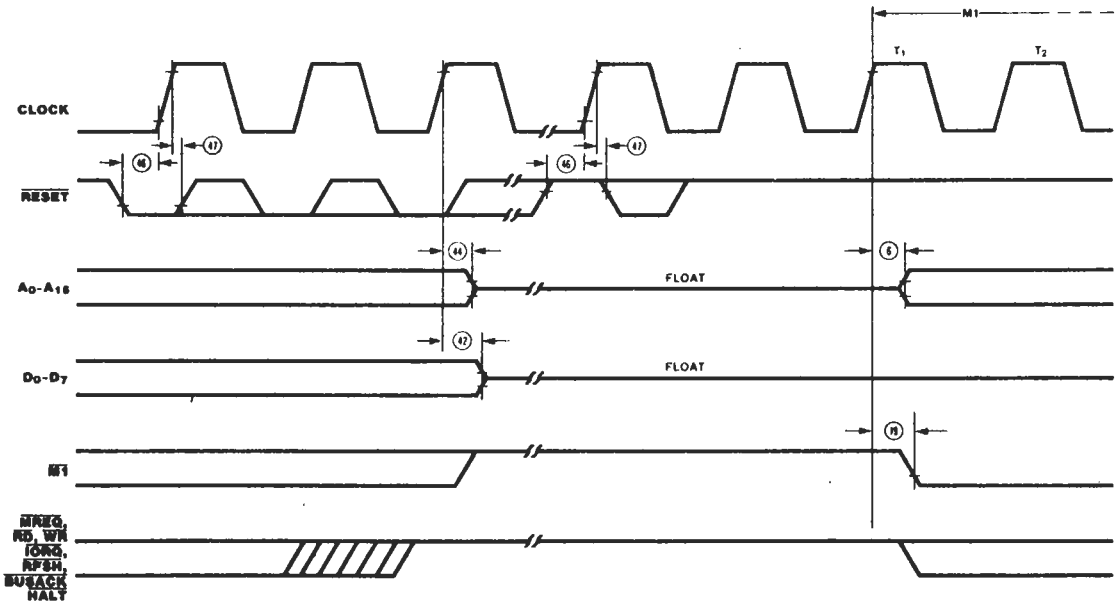



**Figure 12. Reset Cycle**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## AC Characteristics

| Number | Symbol | Parameter | Z80 CPU Min | Z80 CPU Max | Z80A CPU Min | Z80A CPU Max | Z80B CPU Min | Z80B CPU Max | Z80H CPU† Min | Z80H CPU† Max |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TcC | Clock Cycle Time | 400* | | 250* | | 165* | | 125* | |
| 2 | TwCh | Clock Pulse Width (High) | 180* | | 110* | | 65* | | 55* | |
| 3 | TwCl | Clock Pulse Width (Low) | 180 | 2000 | 110 | 2000 | 65 | 2000 | 55 | 2000 |
| 4 | TfC | Clock Fall Time | — | 30 | — | 30 | — | 20 | — | 10 |
| 5 | TrC | Clock Rise Time | — | 30 | — | 30 | — | 20 | — | 10 |
| 6 | TdCr(A) | Clock ↑ to Address Valid Delay | — | 145 | — | 110 | — | 90 | — | 80 |
| 7 | TdA(MREQf) | Address Valid to $\overline{MREQ}$ ↓ Delay | 125* | — | 65* | — | 35* | — | 20* | — |
| 8 | TdCf(MREQf) | Clock ↓ to $\overline{MREQ}$ ↓ Delay | — | 100 | — | 85 | — | 70 | — | 60 |
| 9 | TdCr(MREQr) | Clock ↑ to $\overline{MREQ}$ ↑ Delay | — | 100 | — | 85 | — | 70 | — | 60 |
| 10 | TwMREQh | $\overline{MREQ}$ Pulse Width (High) | 170* | | 110* | | 65* | | 45* | |
| 11 | TwMREQl | $\overline{MREQ}$ Pulse Width (Low) | 360* | — | 220* | — | 135* | — | 100* | — |
| 12 | TdCf(MREQr) | Clock ↓ to $\overline{MREQ}$ ↑ Delay | — | 100 | — | 85 | — | 70 | — | 60 |
| 13 | TdCf(RDf) | Clock ↓ to $\overline{RD}$ ↓ Delay | — | 130 | — | 95 | — | 80 | — | 70 |
| 14 | TdCr(RDr) | Clock ↑ to $\overline{RD}$ ↑ Delay | — | 100 | — | 85 | — | 70 | — | 60 |
| 15 | TsD(Cr) | Data Setup Time to Clock ↑ | 50 | | 35 | | 30 | | 30 | |
| 16 | ThD(RDr) | Data Hold Time to $\overline{RD}$ ↑ | — | 0 | — | 0 | — | 0 | — | 0 |
| 17 | TsWAIT(Cf) | $\overline{WAIT}$ Setup Time to Clock ↓ | 70 | — | 70 | — | 60 | — | 50 | — |
| 18 | ThWAIT(Cf) | $\overline{WAIT}$ Hold Time after Clock ↓ | — | 0 | — | 0 | — | 0 | — | 0 |
| 19 | TdCr(M1f) | Clock ↑ to $\overline{M1}$ ↓ Delay | — | 130 | — | 100 | — | 80 | — | 70 |
| 20 | TdCr(M1r) | Clock ↑ to $\overline{M1}$ ↑ Delay | — | 130 | — | 100 | — | 80 | — | 70 |
| 21 | TdCr(RFSHf) | Clock ↑ to $\overline{RFSH}$ ↓ Delay | — | 180 | — | 130 | — | 110 | — | 95 |
| 22 | TdCr(RFSHr) | Clock ↑ to $\overline{RFSH}$ ↑ Delay | — | 150 | — | 120 | — | 100 | — | 85 |
| 23 | TdCf(RDr) | Clock ↓ to $\overline{RD}$ ↑ Delay | — | 110 | — | 85 | — | 70 | — | 60 |
| 24 | TdCr(RDf) | Clock ↑ to $\overline{RD}$ ↓ Delay | — | 100 | — | 85 | — | 70 | — | 60 |
| 25 | TsD(Cf) | Data Setup to Clock ↓ during $M_2$, $M_3$, $M_4$ or $M_5$ Cycles | 60 | | 50 | | 40 | | 30 | |
| 26 | TdA(IORQf) | Address Stable prior to $\overline{IORQ}$ ↓ | 320* | — | 180* | — | 110* | — | 75* | — |
| 27 | TdCr(IORQf) | Clock ↑ to $\overline{IORQ}$ ↓ Delay | — | 90 | — | 75 | — | 65 | — | 55 |
| 28 | TdCf(IORQr) | Clock ↓ to $\overline{IORQ}$ ↑ Delay | — | 110 | — | 85 | — | 70 | — | 60 |
| 29 | TdD(WRf) | Data Stable prior to $\overline{WR}$ ↓ | 190* | — | 80* | — | 25* | — | 5* | — |
| 30 | TdCf(WRf) | Clock ↓ to $\overline{WR}$ ↓ Delay | — | 90 | — | 80 | — | 70 | — | 60 |
| 31 | TwWR | $\overline{WR}$ Pulse Width | 360* | — | 220* | — | 135* | — | 100* | — |
| 32 | TdCf(WRr) | Clock ↓ to $\overline{WR}$ ↑ Delay | — | 100 | — | 80 | — | 70 | — | 60 |
| 33 | TdD(WRf) | Data Stable prior to $\overline{WR}$ ↓ | 20* | — | -10* | — | -55* | — | 55* | — |
| 34 | TdCr(WRf) | Clock ↑ to $\overline{WR}$ ↓ Delay | — | 80 | — | 65 | — | 60 | — | 55 |
| 35 | TdWRr(D) | Data Stable from $\overline{WR}$ ↑ | 120* | | 60* | | 30* | | 15* | |
| 36 | TdCf(HALT) | Clock ↓ to $\overline{HALT}$ ↑ or ↓ | — | 300 | — | 300 | — | 260 | — | 225 |
| 37 | TwNMI | $\overline{NMI}$ Pulse Width | 80 | — | 80 | — | 70 | — | 60* | — |
| 38 | TsBUSREQ(Cr) | $\overline{BUSREQ}$ Setup Time to Clock ↑ | 80 | — | 50 | — | 50 | — | 40 | — |

*For clock periods other than the minimums shown in the table, calculate parameters using the expressions in the table on the following page.

† Units in nanoseconds (ns). All timings are preliminary and subject to change.





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**AC Characteristics** (Continued)

| Number | Symbol | Parameter | Z80 CPU Min | Z80 CPU Max | Z80A CPU Min | Z80A CPU Max | Z80B CPU Min | Z80B CPU Max | Z80H CPU† Min | Z80H CPU† Max |
|---|---|---|---|---|---|---|---|---|---|---|
| 39 | ThBUSREQ(Cr) | $\overline{BUSREQ}$ Hold Time after Clock ↑ | 0 | — | 0 | — | 0 | — | 0 | — |
| 40 | TdCr(BUSACKf) | Clock ↑ to $\overline{BUSACK}$ ↓ Delay | | 120 | | 100 | | 90 | — | 80 |
| 41 | TdCf(BUSACKr) | Clock ↓ to $\overline{BUSACK}$ ↑ Delay | — | 110 | — | 100 | — | 90 | — | 80 |
| 42 | TdCr(Dz) | Clock ↑ to Data Float Delay | — | 90 | — | 90 | — | 80 | — | 70 |
| 43 | TdCr(CTz) | Clock ↑ to Control Outputs Float Delay ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$) | — | 110 | — | 80 | — | 70 | — | 60 |
| 44 | TdCr(Az) | Clock ↑ to Address Float Delay | — | 110 | — | 90 | — | 80 | — | 70 |
| 45 | TdCTr(A) | $\overline{MREQ}$ ↑, $\overline{IORQ}$ ↑, $\overline{RD}$ ↑, and $\overline{WR}$ ↑ to Address Hold Time | 160* | | 80* | | 35* | | 20* | — |
| 46 | TsRESET(Cr) | $\overline{RESET}$ to Clock ↑ Setup Time | 90 | — | 60 | — | 60 | — | 45 | — |
| 47 | ThRESET(Cr) | $\overline{RESET}$ to Clock ↑ Hold Time | — | 0 | — | 0 | — | 0 | — | 0 |
| 48 | TsINTf(Cr) | $\overline{INT}$ to Clock ↑ Setup Time | 80 | — | 80 | — | 70 | — | 55 | — |
| 49 | ThINTr(Cr) | $\overline{INT}$ to Clock ↑ Hold Time | — | 0 | — | 0 | — | 0 | — | 0 |
| 50 | TdM1f(IORQf) | $\overline{M1}$ ↓ to $\overline{IORQ}$ ↓ Delay | 920* | | 565* | | 365* | | 270* | — |
| 51 | TdCf(IORQf) | Clock ↓ to $\overline{IORQ}$ ↓ Delay | — | 110 | — | 85 | — | 70 | — | 60 |
| 52 | TdCf(IORQr) | Clock ↓ to $\overline{IORQ}$ ↑ Delay | — | 100 | — | 85 | — | 70 | — | 60 |
| 53 | TdCf(D) | Clock ↓ to Data Valid Delay | — | 230 | — | 150 | — | 130 | — | 115 |

*For clock periods other than the minimums shown in the table, calculate parameters using the following expressions. Calculated values above assumed TrC = TfC = 20 ns.

† Units in nanoseconds (ns). All timings are preliminary and subject to change.

**Footnotes to AC Characteristics**

| Number | Symbol | Z80 | Z80A | Z80B |
|---|---|---|---|---|
| 1 | TcC | TwCh + TwCl + TrC + TfC | TwCh + TwCl + TrC + TfC | TwCh + TwCl + TrC + TfC |
| 2 | TwCh | Although static by design, TwCh of greater than 200 μs is not guaranteed | Although static by design, TwCh of greater than 200 μs is not guaranteed | Although static by design, TwCh of greater than 200 μs is not guaranteed |
| 7 | TdA(MREQf) | TwCh + TfC − 75 | TwCh + TfC − 65 | TwCh + TfC − 50 |
| 10 | TwMREQh | TwCh + TfC − 30 | TwCh + TfC − 20 | TwCh + TfC − 20 |
| 11 | TwMREQl | TcC − 40 | TcC − 30 | TcC − 30 |
| 26 | TdA(IORQf) | TcC − 80 | TcC − 70 | TcC − 55 |
| 29 | TdD(WRf) | TcC − 210 | TcC − 170 | TcC − 140 |
| 31 | TwWR | TcC − 40 | TcC − 30 | TcC − 30 |
| 33 | TdD(WRf) | TwCl + TrC − 180 | TwCl + TrC − 140 | TwCl + TrC − 140 |
| 35 | TdWRr(D) | TwCl + TrC − 80 | TwCl + TrC − 70 | TwCl + TrC − 55 |
| 45 | TdCTr(A) | TwCl + TrC − 40 | TwCl + TrC − 50 | TwCl + TrC − 50 |
| 50 | TdM1f(IORQf) | 2TcC + TwCh + TfC − 80 | 2TcC + TwCh + TfC − 65 | 2TcC + TwCh + TfC − 50 |

AC Test Conditions:

$V_{IH}$ = 2.0 V
$V_{IL}$ = 0.8 V
$V_{IHC}$ = $V_{CC}$ −0.6 V
$V_{ILC}$ = 0.45 V
$V_{OH}$ = 2.0 V
$V_{OL}$ = 0.8 V
FLOAT = ±0.5 V



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Absolute Maximum Ratings

Storage Temperature . . . . . -65°C to +150°C
Temperature under Bias . . . . . Specified operating range
Voltages on all inputs and outputs with respect to ground . -0.3 V to +7 V
Power Dissipation . . . . . 1.5 W

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

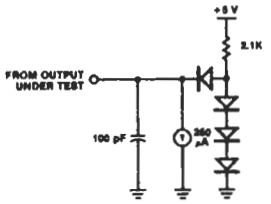
## Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0 V). Positive current flows into the referenced pin. Available operating temperature ranges are:

- S* = 0°C to +70°C, +4.75 V ≤ $V_{CC}$ ≤ +5.25 V
- E* = -40°C to +85°C, +4.75 V ≤ $V_{CC}$ ≤ +5.25 V
- M* = -55°C to +125°C, +4.5 V ≤ $V_{CC}$ ≤ +5.5 V

*See Ordering Information section for package temperature range and product number.

All ac parameters assume a load capacitance of 100 pF. Add 10 ns delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus and 100 pF for address and control lines.




## DC Characteristics

| Symbol | Parameter | Min | Max | Unit | Test Condition |
|---|---|---|---|---|---|
| $V_{ILC}$ | Clock Input Low Voltage | -0.3 | 0.45 | V | |
| $V_{IHC}$ | Clock Input High Voltage | $V_{CC}$-.6 | $V_{CC}$+.3 | V | |
| $V_{IL}$ | Input Low Voltage | -0.3 | 0.8 | V | |
| $V_{IH}$ | Input High Voltage | 2.0 | $V_{CC}$ | V | |
| $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{OL}$ = 1.8 mA |
| $V_{OH}$ | Output High Voltage | 2.4 | | V | $I_{OH}$ = -250 μA |
| $I_{CC}$ | Power Supply Current | | | | |
| | Z80 | | 150[1] | mA | |
| | Z80A | | 200[2] | mA | |
| | Z80B | | 200 | mA | |
| $I_{LI}$ | Input Leakage Current | | 10 | μA | $V_{IN}$ = 0 to $V_{CC}$ |
| $I_{LO}$ | 3-State Output Leakage Current in Float | -10 | 10[3] | μA | $V_{OUT}$ = 0.4 to $V_{CC}$ |

1. For military grade parts, $I_{CC}$ is 200 mA.
2. Typical rate for Z80A is 90 mA.
3. $A_{15}$-$A_0$, $D_7$-$D_0$, MREQ, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$.

## Capacitance

| Symbol | Parameter | Min | Max | Unit | Note |
|---|---|---|---|---|---|
| $C_{CLOCK}$ | Clock Capacitance | | 35 | pF | Unmeasured pins returned to ground |
| $C_{IN}$ | Input Capacitance | | 5 | pF | |
| $C_{OUT}$ | Output Capacitance | | 10 | pF | |

$T_A$ = 25°C, f = 1 MHz.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Ordering Information

| Product Number | Package/ Temp | Speed | Description | Product Number | Package/ Temp | Speed | Description |
|---|---|---|---|---|---|---|---|
| Z8400 | CE | 2.5 MHz | Z80 CPU (40-pin) | Z8400A | CMB | 4.0 MHz | Z80A CPU (40-pin) |
| Z8400 | CM | 2.5 MHz | Same as above | Z8400A | CS | 4.0 MHz | Same as above |
| Z8400 | CMB | 2.5 MHz | Same as above | Z8400A | DE | 4.0 MHz | Same as above |
| Z8400 | CS | 2.5 MHz | Same as above | Z8400A | DS | 4.0 MHz | Same as above |
| Z8400 | DE | 2.5 MHz | Same as above | Z8400A | PE | 4.0 MHz | Same as above |
| Z8400 | DS | 2.5 MHz | Same as above | Z8400A | PS | 4.0 MHz | Same as above |
| Z8400 | PE | 2.5 MHz | Same as above | Z8400B | CS | 6.0 MHz | Z80B CPU (40 pin) |
| Z8400 | PS | 2.5 MHz | Same as above | Z8400B | DS | 6.0 MHz | Same as above |
| Z8400A | CE | 4.0 MHz | Z80A CPU (40-pin) | Z8400B | PS | 6.0 MHz | Same as above |
| Z8400A | CM | 4.0 MHz | Same as above | | | | |

*NOTES: C = Ceramic, D = Cerdip, P = Plastic; E = -40°C to +85°C, M = -55°C to +125°C, MB = -55°C to +125°C with MIL-STD-883 Class B processing, S = 0°C to +70°C.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

# Z8430
# Z80® CTC Counter/ Timer Circuit

# Zilog

## Product Specification

September 1983



**Features**

- Four independently programmable counter/timer channels, each with a readable downcounter and a selectable 16 or 256 prescaler. Downcounters are reloaded automatically at zero count.
- Three channels have Zero Count/Timeout outputs capable of driving Darlington transistors.
- Selectable positive or negative trigger initiates timer operation.
- Standard Z-80 Family daisy-chain interrupt structure provides fully vectored, prioritized interrupts without external logic. The CTC may also be used as an interrupt controller.
- Interfaces directly to the Z-80 CPU or—for baud rate generation—to the Z-80 SIO.

**General Description**

The Z-80 CTC four-channel counter/timer can be programmed by system software for a broad range of counting and timing applications. The four independently programmable channels of the Z-80 CTC satisfy common microcomputer system requirements for event counting, interrupt and interval timing, and general clock rate generation.

System design is simplified because the CTC connects directly to both the Z-80 CPU and the Z-80 SIO with no additional logic. In larger systems, address decoders and buffers may be required.

Programming the CTC is straightforward: each channel is programmed with two bytes; a third is necessary when interrupts are enabled. Once started, the CTC counts down, reloads its time constant automatically, and resumes counting. Software timing loops are completely eliminated. Interrupt processing is simplified because only one vector need be specified; the CTC internally generates a unique vector for each channel.

The Z-80 CTC requires a single +5 V power supply and the standard Z-80 single-phase system clock. It is fabricated with n-channel silicon-gate depletion-load technology, and packaged in a 28-pin plastic or ceramic DIP.

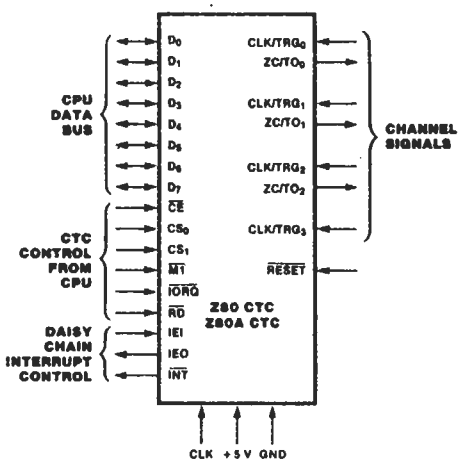



Figure 1. Pin Functions

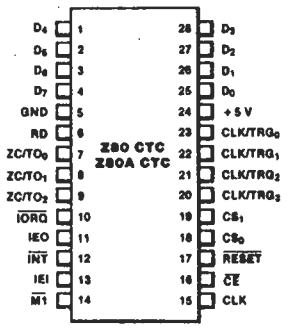



Figure 2. Pin Assignments



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Functional Description

The Z-80 CTC has four independent counter/timer channels. Each channel is individually programmed with two words: a control word and a time-constant word. The control word selects the operating mode (counter or timer), enables or disables the channel interrupt, and selects certain other operating parameters. If the timing mode is selected, the control word also sets a prescaler, which divides the system clock by either 16 or 256. The time-constant word is a value from 1 to 256.

During operation, the individual counter channel counts down from the preset time constant value. In counter mode operation the counter decrements on each of the CLK/TRG input pulses until zero count is reached. Each decrement is synchronized by the system clock. For counts greater than 256, more than one counter can be cascaded. At zero count, the down-counter is automatically reset with the time constant value.

The timer mode determines time intervals as small as 4 $\mu$s (Z-80A) or 6.4 $\mu$s (Z-80) without additional logic or software timing loops. Time intervals are generated by dividing the system clock with a prescaler that decrements a preset down-counter.

Thus, the time interval is an integral multiple of the clock period, the prescaler value (16 or 256) and the time constant that is preset in the down-counter. A timer is triggered automatically when its time constant value is programmed, or by an external CLK/TRG input.

Three channels have two outputs that occur at zero count. The first output is a zero-count/timeout pulse at the ZC/TO output. The fourth channel (Channel 3) does not have a ZC/TO output; interrupt request is the only output available from Channel 3.

The second output is Interrupt Request ($\overline{INT}$), which occurs if the channel has its interrupt enabled during programming. When the Z-80 CPU acknowledges Interrupt Request, the Z-80 CTC places an interrupt vector on the data bus.

The four channels of the Z-80 CTC are fully prioritized and fit into four contiguous slots in a standard Z-80 daisy-chain interrupt structure. Channel 0 is the highest priority and Channel 3 the lowest. Interrupts can be individually enabled (or disabled) for each of the four channels.

## Architecture

The CTC has four major elements, as shown in Figure 3.

- CPU bus I/O
- Channel control logic
- Interrupt logic
- Counter/timer circuits

**CPU Bus I/O.** The CPU bus I/O circuit decodes the address inputs, and interfaces the CPU data and control signals to the CTC for distribution on the internal bus.

**Internal Control Logic.** The CTC internal control logic controls overall chip operating functions such as the chip enable, reset, and read/write logic.

**Interrupt Logic.** The interrupt control logic ensures that the CTC interrupts interface properly with the Z-80 CPU interrupt system. The logic controls the interrupt priority of the CTC as a function of the IEI signal. If IEI is High, the CTC has priority. During interrupt

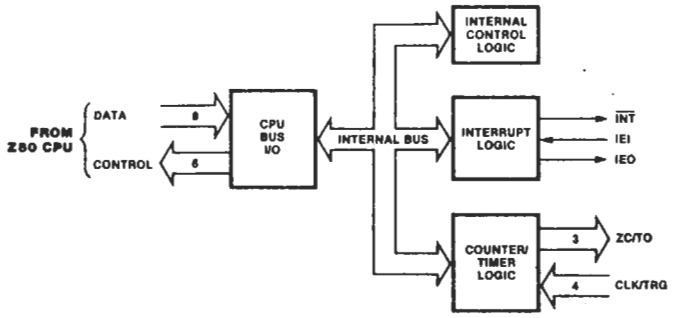



**Figure 3. Functional Block Diagram**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Architecture** (Continued)

processing, the interrupt logic holds IEO Low, which inhibits the interrupt operation on lower priority devices. If the IEI input goes Low, priority is relinquished and the interrupt logic drives IEO Low.

If a channel is programmed to request an interrupt, the interrupt logic drives IEO Low at the zero count, and generates an $\overline{INT}$ signal to the Z-80 CPU. When the Z-80 CPU responds with interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$), then the interrupt logic arbitrates the CTC internal priorities, and the interrupt control logic places a unique interrupt vector on the data bus.

If an interrupt is pending, the interrupt logic holds IEO Low. When the Z-80 CPU issues a Return From Interrupt (RETI) instruction, each peripheral device decodes the first byte ($ED_{16}$). If the device has a pending interrupt, it raises IEO (High) for one $\overline{M1}$ cycle. This ensures that all lower priority devices can decode the entire RETI instruction and reset properly.

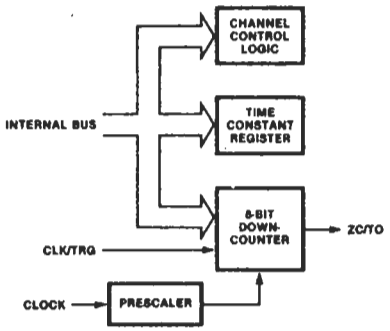



**Figure 4. Counter/Timer Block Diagram**

**Counter/Timer Circuits.** The CTC has four independent counter/timer circuits, each containing the logic shown in Figure 4.

**Channel Control Logic.** The channel control logic receives the 8-bit channel control word when the counter/timer channel is programmed. The channel control logic decodes the control word and sets the following operating conditions:

- Interrupt enable (or disable)
- Operating mode (timer or counter)
- Timer mode prescaler factor (16 or 256)
- Active slope for CLK/TRG input
- Timer mode trigger (automatic or CLK/TRG input)
- Time constant data word to follow
- Software reset

**Time Constant Register.** When the counter/timer channel is programmed, the time constant register receives and stores an 8-bit time constant value, which can be anywhere from 1 to 256 (0 = 256). This constant is automatically loaded into the down-counter when the counter/timer channel is initialized, and subsequently after each zero count.

**Prescaler.** The prescaler, which is used only in timer mode, divides the system clock frequency by a factor of either 16 or 256. The prescaler output clocks the down-counter during timer operation. The effect of the prescaler on the down-counter is a multiplication of the system clock period by 16 or 256. The prescaler factor is programmed by bit 5 of the channel control word.

**Down-Counter.** Prior to each count cycle, the down-counter is loaded with the time constant register contents. The counter is then decremented one of two ways, depending on operating mode:

- By the prescaler output (timer mode)
- By the trigger pulses into the CLK/TRG input (counter mode)

Without disturbing the down-count, the Z-80 CPU can read the count remaining at any time by performing an I/O read operation at the port address assigned to the CTC channel. When the down-counter reaches the zero count, the ZC/TO output generates a positive-going pulse. When the interrupt is enabled, zero count also triggers an interrupt request signal ($\overline{INT}$) from the interrupt logic.





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Programming**

Each Z-80 CTC channel must be programmed prior to operation. Programming consists of writing two words to the I/O port that corresponds to the desired channel. The first word is a control word that selects the operating mode and other parameters; the second word is a time constant, which is a binary data word with a value from 1 to 256. A time constant word must be preceded by a channel control word.

After initialization, channels may be reprogrammed at any time. If updated control and time constant words are written to a channel during the count operation, the count continues to zero before the new time constant is loaded into the counter.

If the interrupt on any Z-80 CTC channel is enabled, the programming procedure should also include an interrupt vector. Only one vector is required for all four channels, because the interrupt logic automatically modifies the vector for the channel requesting service.

A control word is identified by a 1 in bit 0. A 1 in bit 2 indicates a time constant word is to follow. Interrupt vectors are always addressed to Channel 0, and identified by a 0 in bit 0.

**Addressing.** During programming, channels are addressed with the channel select pins $CS_1$ and $CS_2$. A 2-bit binary code selects the appropriate channel as shown in the following table.

| Channel | $CS_1$ | $CS_0$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

**Reset.** The CTC has both hardware and software resets. The hardware reset terminates all down-counts and disables all CTC interrupts by resetting the interrupt bits in the control registers. In addition, the ZC/TO and Interrupt outputs go inactive, IEO reflects IEI, and $D_0$–$D_7$ go to the high-impedance state. All channels must be completely reprogrammed after a hardware reset.

The software reset is controlled by bit 1 in the channel control word. When a channel receives a software reset, it stops counting. When a software reset is used, the other bits in the control word also change the contents of the channel control register. After a software reset a new time constant word must be written to the same channel.

If the channel control word has both bits $D_1$ and $D_2$ set to 1, the addressed channel stops operating, pending a new time constant word. The channel is ready to resume after the new constant is programmed. In timer mode, if $D_3 = 0$, operation is triggered automatically when the time constant word is loaded.

**Channel Control Word Programming.** The channel control word is shown in Figure 5. It sets the modes and parameters described below.

*Interrupt Enable.* $D_7$ enables the interrupt, so that an interrupt output ($\overline{INT}$) is generated at zero count. Interrupts may be programmed in either mode and may be enabled or disabled at any time.

*Operating Mode.* $D_6$ selects either timer or counter mode.

*Prescaler Factor. (Timer Mode Only).* $D_5$ selects factor—either 16 or 256.

*Trigger Slope.* $D_4$ selects the active edge or slope of the CLK/TRG input pulses. Note that reprogramming the CLK/TRG slope during operation is equivalent to issuing an active edge. If the trigger slope is changed by a control word update while a channel is pending operation in timer mode, the result is the same as a CLK/TRG pulse and the timer starts. Similarly, if the channel is in counter mode, the counter decrements.

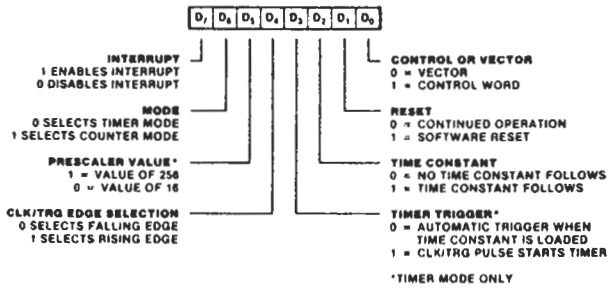



**Figure 5. Channel Control Word**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Programming** (Continued)

**Trigger Mode (Timer Mode Only).** $D_3$ selects the trigger mode for timer operation. When $D_3$ is reset to 0, the timer is triggered automatically. The time constant word is programmed during an I/O write operation, which takes one machine cycle. At the end of the write operation there is a setup delay of one clock period. The timer starts automatically (decrements) on the rising edge of the second clock pulse ($T_2$) of the machine cycle following the write operation. Once started, the timer runs continuously. At zero count the timer reloads automatically and continues counting without interruption or delay, until stopped by a reset.

When $D_3$ is set to 1, the timer is triggered externally through the CLK/TRG input. The time constant word is programmed during an I/O write operation, which takes one machine cycle. The timer is ready for operation on the rising edge of the second clock pulse ($T_2$) of the following machine cycle. Note that the first timer decrement follows the active edge of the CLK/TRG pulse by a delay time of one clock cycle if a minimum setup time to the rising edge of clock is met. If this minimum is not met, the delay is extended by another clock period. Consequently, for immediate triggering, the CLK/TRG input must precede $T_2$ by one clock cycle plus its minimum setup time. If the minimum time is not met, the timer will start on the third clock cycle ($T_3$).

Once started the timer operates continuously, without interruption or delay, until stopped by a reset.

**Time Constant to Follow.** A 1 in $D_2$ indicates that the next word addressed to the selected channel is a time constant data word for the time constant register. The time constant word may be written at any time.

A 0 in $D_2$ indicates no time constant word is to follow. This is ordinarily used when the channel is already in operation and the new channel control word is an update. A channel will not operate without a time constant value. The only way to write a time constant value is to write a control word with $D_2$ set.

**Software Reset.** Setting $D_1$ to 1 causes a software reset, which is described in the Reset section.

**Control Word.** Setting $D_0$ to 1 identifies the word as a control word.

**Time Constant Programming.** Before a channel can start counting it must receive a time constant word from the CPU. During programming or reprogramming, a channel control word in which bit 2 is set must precede the time constant word to indicate that the next word is a time constant. The time constant word can be any value from 1 to 256 (Figure 6). Note that $00_{16}$ is interpreted as 256.

In timer mode, the time interval is controlled by three factors:

- The system clock period ($\phi$)
- The prescaler factor (P), which multiplies the interval by either 16 or 256
- The time constant (T), which is programmed into the time constant register

Consequently, the time interval is the product of $\phi \times P \times T$. The minimum timer resolution is $16 \times \phi$ (4 $\mu$s with a 4 MHz clock). The maximum timer interval is $256 \times \phi \times 256$ (16.4 ms with a 4 MHz clock). For longer intervals timers may be cascaded.

**Interrupt Vector Programming.** If the Z-80 CTC has one or more interrupts enabled, it can supply interrupt vectors to the Z-80 CPU. To do so, the Z-80 CTC must be pre-programmed with the most-significant five bits of the interrupt vector. Programming consists of writing a vector word to the I/O port corresponding to the Z-80 CTC Channel 0. Note that $D_0$ of the vector word is always zero, to distinguish the vector from a channel control word. $D_1$ and $D_2$ are not used in programming the vector word. These bits are supplied by the interrupt logic to identify the channel requesting interrupt service with a unique interrupt vector (Figure 7). Channel 0 has the highest priority.

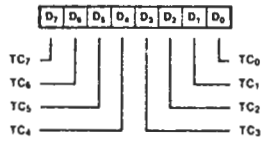



**Figure 6. Time Constant Word**

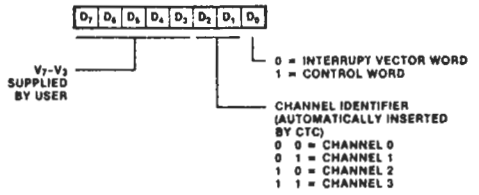



**Figure 7. Interrupt Vector Word**





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Pin Description**

**$\overline{CE}$.** *Chip Enable* (input, active Low). When enabled the CTC accepts control words, interrupt vectors, or time constant data words from the data bus during an I/O write cycle; or transmits the contents of the down-counter to the CPU during an I/O read cycle. In most applications this signal is decoded from the eight least significant bits of the address bus for any of the four I/O port addresses that are mapped to the four counter-timer channels.

**CLK.** *System Clock* (input). Standard single-phase Z-80 system clock.

**CLK/TRG$_0$-CLK/TRG$_3$.** *External Clock/Timer Trigger* (input, user-selectable active High or Low). Four pins corresponding to the four Z-80 CTC channels. In counter mode, every active edge on this pin decrements the down-counter. In timer mode, an active edge starts the timer.

**CS$_0$-CS$_1$.** *Channel Select* (inputs active High). Two-bit binary address code selects one of the four CTC channels for an I/O write or read (usually connected to $A_0$ and $A_1$).

**D$_0$-D$_7$.** *System Data Bus* (bidirectional, 3-state). Transfers all data and commands between the Z-80 CPU and the Z-80 CTC.

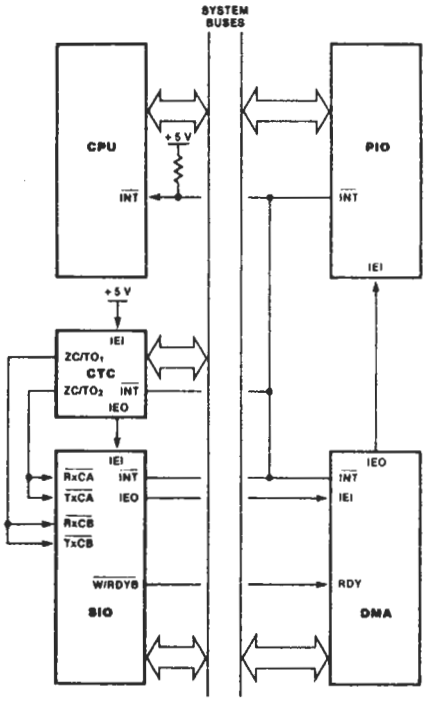



**Figure 8. A Typical Z-80 Environment**

**IEI.** *Interrupt Enable In* (input, active High). A High indicates that no other interrupting devices of higher priority in the daisy chain are being serviced by the Z-80 CPU.

**IEO.** *Interrupt Enable Out* (output, active High). High only if IEI is High and the Z-80 CPU is not servicing an interrupt from any Z-80 CTC channel. IEO blocks lower priority devices from interrupting while a higher priority interrupting device is being serviced.

**$\overline{INT}$.** *Interrupt Request* (output, open drain, active Low). Low when any Z-80 CTC channel that has been programmed to enable interrupts has a zero-count condition in its down-counter.

**$\overline{IORQ}$.** *Input/Output Request* (input from CPU, active Low). Used with $\overline{CE}$ and $\overline{RD}$ to transfer data and channel control words between the Z-80 CPU and the Z-80 CTC. During a write cycle, $\overline{IORQ}$ and $\overline{CE}$ are active and $\overline{RD}$ inactive. The Z-80 CTC does not receive a specific write signal; rather, it internally generates its own from the inverse of an active $\overline{RD}$ signal. In a read cycle, $\overline{IORQ}$, $\overline{CE}$ and $\overline{RD}$ are active; the contents of the down-counter are read by the Z-80 CPU. If $\overline{IORQ}$ and $\overline{M1}$ are both true, the CPU is acknowledging an interrupt request, and the highest priority interrupting channel places its interrupt vector on the Z-80 data bus.

**$\overline{M1}$.** *Machine Cycle One* (input from CPU, active Low). When $\overline{M1}$ and $\overline{IORQ}$ are active, the Z-80 CPU is acknowledging an interrupt. The Z-80 CTC then places an interrupt vector on the data bus if it has highest priority, and if a channel has requested an interrupt ($\overline{INT}$).

**$\overline{RD}$.** *Read Cycle Status* (input, active Low). Used in conjunction with $\overline{IORQ}$ and $\overline{CE}$ to transfer data and channel control words between the Z-80 CPU and the Z-80 CTC.

**$\overline{RESET}$.** *Reset* (input active Low). Terminates all down-counts and disables all interrupts by resetting the interrupt bits in all control registers; the ZC/TO and the Interrupt outputs go inactive; IEO reflects IEI; $D_0$-$D_7$ go to the high-impedance state.

**ZC/TO$_0$-ZC/TO$_2$.** *Zero Count/Timeout* (output, active High). Three ZC/TO pins corresponding to Z-80 CTC channels 2 through 0 (Channel 3 has no ZC/TO pin). In both counter and timer modes the output is an active High pulse when the down-counter decrements to zero.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Timing

**Read Cycle Timing.** Figure 9 shows read cycle timing. This cycle reads the contents of a down-counter without disturbing the count. During clock cycle $T_2$, the Z-80 CPU initiates a read cycle by driving the following inputs Low: $\overline{RD}$, $\overline{IORQ}$, and $\overline{CE}$. A 2-bit binary code at inputs $CS_1$ and $CS_0$ selects the channel to be read. $\overline{M1}$ must be High to distinguish this cycle from an interrupt acknowledge. No additional wait states are allowed.

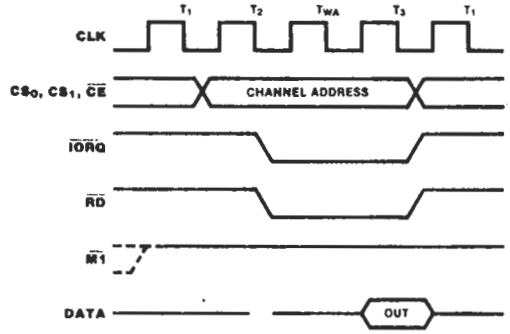



**Figure 9. Read Cycle Timing**

**Write Cycle Timing.** Figure 10 shows write cycle timing for loading control, time constant or vector words.

The CTC does not have a write signal input, so it generates one internally when the read ($\overline{RD}$) input is High during $T_1$. During $T_2$ $\overline{IORQ}$ and $\overline{CE}$ inputs are Low. $\overline{M1}$ must be High to distinguish a write cycle from an interrupt acknowledge. A 2-bit binary code at inputs $CS_1$ and $CS_0$ selects the channel to be addressed, and the word being written is placed on the Z-80 data bus. The data word is latched into the appropriate register with the rising edge of clock cycle $T_3$.

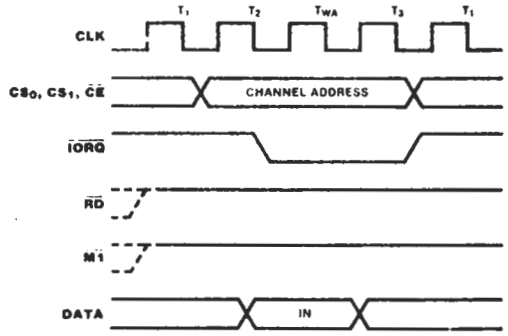



**Figure 10. Write Cycle Timing**

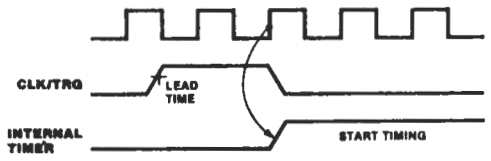



**Figure 11. Timer Mode Timing**

**Timer Operation.** In the timer mode, a CLK/TRG pulse input starts the timer (Figure 11) on the second succeeding rising edge of CLK. The trigger pulse is asynchronous, and it must have a minimum width. A minimum lead time (210 ns) is required between the active edge of the CLK/TRG and the next rising edge of CLK to enable the prescaler on the following clock edge. If the CLK/TRG edge occurs closer than this, the initiation of the timer function is delayed one clock cycle. This corresponds to the startup timing discussed in the programming section. The timer can also be started automatically if so programmed by the channel control word.

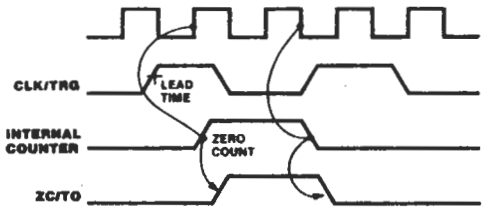



**Figure 12. Counter Mode Timing**

**Counter Operation.** In the counter mode, the CLK/TRG pulse input decrements the down-counter. The trigger is asynchronous, but the count is synchronized with CLK. For the decrement to occur on the next rising edge of CLK, the trigger edge must precede CLK by a minimum lead time as shown in Figure 12. If the lead time is less than specified, the count is delayed by one clock cycle. The trigger pulse must have a minimum width, and the trigger period must be at least twice the clock period.

The ZC/TO output occurs immediately after zero count, and follows the rising CLK edge.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Interrupt Operation

The Z-80 CTC follows the Z-80 system interrupt protocol for nested priority interrupts and return from interrupt, wherein the interrupt priority of a peripheral is determined by its location in a daisy chain. Two lines—IEI and IEO—in the CTC connect it to the system daisy chain. The device closest to the +5 V supply has the highest priority (Figure 13). For additional information on the Z-80 interrupt structure, refer to the *Z-80 CPU Product Specification* and the *Z-80 CPU Technical Manual.*

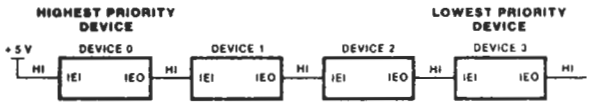



Figure 13. Daisy-Chain Interrupt Priorities

Within the Z-80 CTC, interrupt priority is predetermined by channel number: Channel 0 has the highest priority, and Channel 3 the lowest. If a device or channel is being serviced with an interrupt routine, it cannot be interrupted by a device or channel with lower priority until service is complete. Higher priority devices or channels may interrupt the servicing of lower priority devices or channels.

A Z-80 CTC channel may be programmed to request an interrupt every time its down-counter reaches zero. Note that the CPU must be programmed for interrupt mode 2. Some time after the interrupt request, the CPU sends an interrupt acknowledge. The CTC interrupt control logic determines the highest priority channel that is requesting an interrupt. Then, if the CTC IEI input is High (indicating that it has priority within the system daisy chain) it places an 8-bit interrupt vector on the system data bus. The high-order five bits of this vector were written to the CTC during the programming process; the next two bits are provided by the CTC interrupt control logic as a binary code that identifies the highest priority channel requesting an interrupt; the low-order bit is always zero.

**Interrupt Acknowledge Timing.** Figure 14 shows interrupt acknowledge timing. After an interrupt request, the Z-80 CPU sends an interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$). All channels are inhibited from changing their interrupt request status when $\overline{M1}$ is active—about two clock cycles earlier than $\overline{IORQ}$. $\overline{RD}$ is High to distinguish this cycle from an instruction fetch.

The CTC interrupt logic determines the highest priority channel requesting an interrupt. If the CTC interrupt enable input (IEI) is High, the highest priority interrupting channel within the CTC places its interrupt vector on the data bus when $\overline{IORQ}$ goes Low. Two wait states ($T_{WA}$) are automatically inserted at this time to allow the daisy chain to stabilize. Additional wait states may be added.

**Return from Interrupt Timing.** At the end of an interrupt service routine the RETI (Return From Interrupt) instruction initializes the daisy chain enable lines for proper control of nested priority interrupt handling. The CTC decodes the 2-byte RETI code internally and determines whether it is intended for a channel being serviced. Figure 15 shows RETI timing.

If several Z-80 peripherals are in the daisy chain, IEI settles active (High) on the chip currently being serviced when the opcode $ED_{16}$ is decoded. If the following opcode is $4D_{16}$, the peripheral being serviced is released and its IEO becomes active. Additional wait states are allowed.

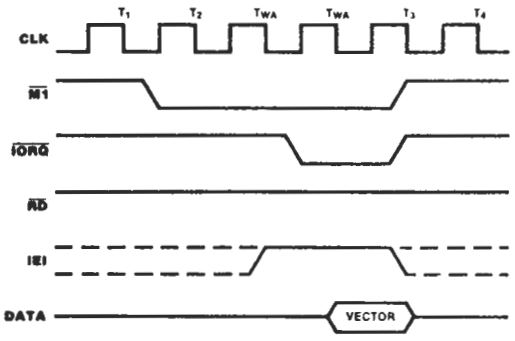



Figure 14. Interrupt Acknowledge Timing

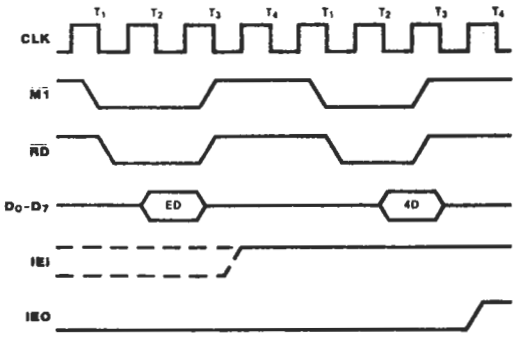



Figure 15. Return From Interrupt Timing





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Absolute Maximum Ratings**

Voltages on all inputs and outputs with respect to GND . . . . . -0.3 V to +7.0 V

Operating Ambient Temperature . . . . . As Specified in Ordering Information

Storage Temperature . . . . . -65°C to +150°C

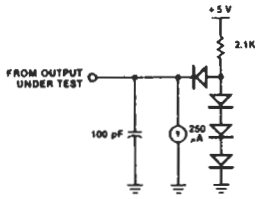
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Test Conditions**

The characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND (0 V). Positive current flows into the referenced pin. Available operating temperature ranges are:

- S* = 0°C to +70°C, +4.75 V ≤ $V_{CC}$ ≤ +5.25 V
- E* = −40°C to +85°C, +4.75 V ≤ $V_{CC}$ ≤ +5.25 V
- M* = −55°C to +125°C, +4.5 V ≤ $V_{CC}$ ≤ +5.5 V

*See Ordering Information section for package temperature range and product number.




**DC Characteristics**

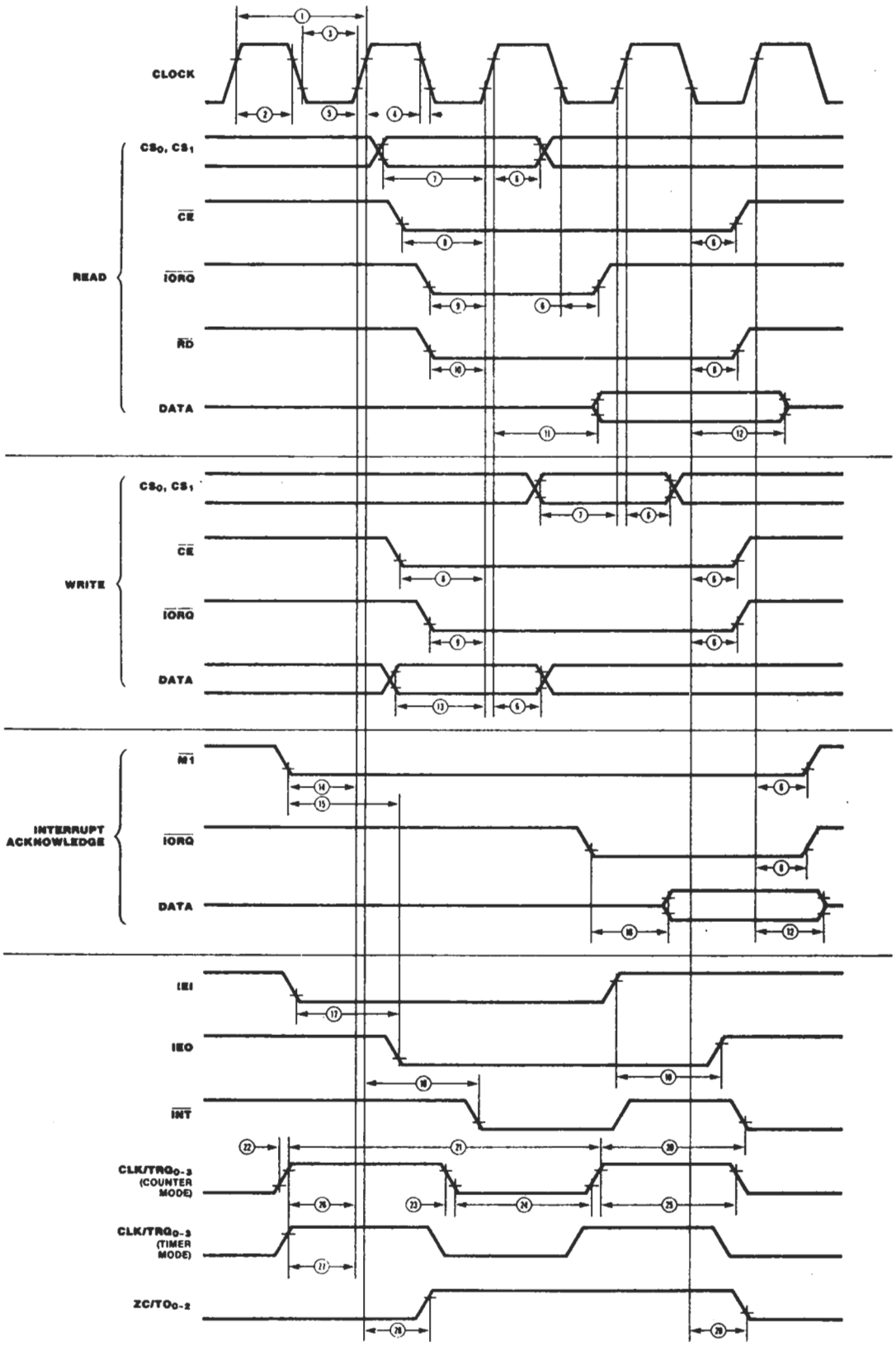
| Symbol | Parameter | Min | Max | Unit | Test Condition |
|---|---|---|---|---|---|
| $V_{ILC}$ | Clock Input Low Voltage | -0.3 | +0.45 | V | |
| $V_{IHC}$ | Clock Input High Voltage | $V_{CC}$-.6 | $V_{CC}$+.3 | V | |
| $V_{IL}$ | Input Low Voltage | -0.3 | +0.8 | V | |
| $V_{IH}$ | Input High Voltage | +2.0 | $V_{CC}$ | V | |
| $V_{OL}$ | Output Low Voltage | | +0.4 | V | $I_{OL}$ = 2 mA |
| $V_{OH}$ | Output High Voltage | +2.4 | | V | $I_{OH}$ = −250 µA |
| $I_{CC}$ | Power Supply Current | | +20 | mA | |
| $I_{LI}$ | Input Leakage Current | | ±10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| $I_{LO}$ | 3-State Output Leakage Current in Float | | ±10 | µA | $V_{OUT}$ = 0.4 to $V_{CC}$ |
| $I_{OHD}$ | Darlington Drive Current | -1.5 | | mA | $V_{OH}$ = 1.5 V, $R_{EXT}$ = 390Ω |

| Symbol | Parameter | Max | Unit | Condition |
|---|---|---|---|---|
| CLK | Clock Capacitance | 20 | pF | Unmeasured pins returned to ground |
| $C_{IN}$ | Input Capacitance | 5 | pF | |
| $C_{OUT}$ | Output Capacitance | 10 | pF | |

$T_A$ = 25°C, f = 1 MHz



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."








"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

| Number | Symbol | Parameter | Z-80 CTC Min (ns) | Z-80 CTC Max (ns) | Z-80A CTC Min (ns) | Z-80A CTC Max (ns) | Z-80B CTC Min (ns) | Z-80B CTC Max (ns) | Notes* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TcC | Clock Cycle Time | 400 | [1] | 250 | [1] | 165 | [1] | |
| 2 | TwCH | Clock Width (High) | 170 | 2000 | 105 | 2000 | 65 | 2000 | |
| 3 | TwCl | Clock Width (Low) | 170 | 2000 | 105 | 2000 | 65 | 2000 | |
| 4 | TfC | Clock Fall Time | | 30 | | 30 | | 20 | |
| 5 | TrC | Clock Rise Time | | 30 | | 30 | | 20 | |
| 6 | Th | All Hold Times | 0 | | 0 | | 0 | | |
| 7 | TsCS(C) | CS to Clock ↑ Setup Time | 250 | | 160 | | 100 | | |
| 8 | TsCE(C) | $\overline{CE}$ to Clock ↑ Setup Time | 200 | | 150 | | 100 | | |
| 9 | TsIO(C) | $\overline{IORQ}$ ↓ to Clock ↑ Setup Time | 250 | | 115 | | 70 | | |
| 10 | TsRD(C) | $\overline{RD}$ ↓ to Clock ↑ Setup Time | 240 | | 115 | | 70 | | |
| 11 | TdC(DO) | Clock ↑ to Data Out Delay | | 240 | | 200 | | 130 | [2] |
| 12 | TdC(DOz) | Clock ↓ to Data Out Float Delay | | 230 | | 110 | | 90 | |
| 13 | TsDI(C) | Data In to Clock ↑ Setup Time | 60 | | 50 | | 40 | | |
| 14 | TsM1(C) | $\overline{M1}$ to Clock ↑ Setup Time | 210 | | 90 | | 70 | | |
| 15 | TdM1(IEO) | $\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt immediately preceding $\overline{M1}$) | | 300 | | 190 | | 130 | [3] |
| 16 | TdIO(DOI) | $\overline{IORQ}$ ↓ to Data Out Delay (INTA Cycle) | | 340 | | 160 | | 110 | [2] |
| 17 | TdIEI(IEOf) | IEI ↓ to IEO ↓ Delay | | 190 | | 130 | | 100 | [3] |
| 18 | TdIEI(IEOr) | IEI ↑ to IEO ↑ Delay (After ED Decode) | | 220 | | 160 | | 110 | [3] |
| 19 | TdC(INT) | Clock ↑ to $\overline{INT}$ ↓ Delay | (TcC + 200) | | (TcC + 140) | | TcC + 120 | | [4] |
| 20 | TdCLK(INT) | CLK/TRG ↑ to $\overline{INT}$ ↓ | | | | | | | |
| | | tsCTR(C) satisfied | (19) + (26) | | (19) + (26) | | (19) + (26) | | [5] |
| | | tsCTR(C) not satisfied | (1) + (19) + (26) | | (1) + (19) + (26) | | (1) + (19) + (26) | | [5] |
| 21 | TcCTR | CLK/TRG Cycle Time | (2TcC) | | (2TcC) | | 2TcC | | [5] |
| 22 | TrCTR | CLK/TRG Rise Time | | 50 | | 50 | | 40 | |
| 23 | TfCTR | CLK/TRG Fall Time | | 50 | | 50 | | 40 | |
| 24 | TwCTRl | CLK/TRG Width (Low) | 200 | | 200 | | 120 | | |
| 25 | TwCTRh | CLK/TRG Width (High) | 200 | | 200 | | 120 | | |
| 26 | TsCTR(Cs) | CLK/TRG ↑ to Clock ↑ Setup Time for Immediate Count | 300 | | 210 | | 150 | | [5] |
| 27 | TsCTR(Ct) | CLK/TRG ↑ to Clock ↑ Setup Time for enabling of Prescaler on following clock↑ | 210 | | 210 | | 150 | | [4] |
| 28 | TdC(ZC/TOr) | Clock ↑ to ZC/TO ↑ Delay | | 260 | | 190 | | 140 | |
| 29 | TdC(ZC/TOf) | Clock ↓ to ZC/TO ↓ Delay | | 190 | | 190 | | 140 | |

[A] 2.5 TcC > (n-2) TdIEI(IEOf) + TdM1(IEO) + TsIEI(IO) + TTL buffer delay, if any.
[B] $\overline{RESET}$ must be active for a minimum of 3 clock cycles.

NOTES:
[1] TcC = TwCh + TwCl + TrC + TfC.
[2] Increase delay by 10 ns for each 50 pF increase in loading, 200 pF maximum for data lines, and 100 pF for control lines.
[3] Increase delay by 2 ns for each 10 pF increase in loading, 100 pF maximum.
[4] Timer mode.
[5] Counter mode.
[6] RESET must be active for a minimum of 3 clock cycles.
* All timings are preliminary and subject to change.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

| Product Number | Package/ Temp | Speed | Description | Product Number | Package/ Temp | Speed | Description |
|---|---|---|---|---|---|---|---|
| Z8430 | CE | 2.5 MHz | Z80 CTC (28-pin) | Z8430A | CMB | 4.0 MHz | Z80A CTC (28-pin) |
| Z8430 | CM | 2.5 MHz | Same as above | Z8530A | CS | 4.0 MHz | Same as above |
| Z8430 | CMB | 2.5 MHz | Same as above | Z8430A | DE | 4.0 MHz | Same as above |
| Z8430 | CS | 2.5 MHz | Same as above | Z8430A | DS | 4.0 MHz | Same as above |
| Z8430 | DE | 2.5 MHz | Same as above | Z8430A | PE | 4.0 MHz | Same as above |
| Z8430 | DS | 2.5 MHz | Same as above | Z8430A | PS | 4.0 MHz | Same as above |
| Z8430 | PE | 2.5 MHz | Same as above | Z8430B | CS | 6.0 MHz | Same as above |
| Z8430 | PS | 2.5 MHz | Same as above | Z8430B | DS | 6.0 MHz | Same as above |
| Z8430A | CE | 4.0 MHz | Z80A CTC (28-pin) | Z8430B | PS | 6.0 MHz | Same as above |
| Z8430A | CM | 4.0 MHz | Same as above | | | | |

*NOTES: C = Ceramic, D = Cerdip, P = Plastic; E = -40°C to +85°C, M = -55°C to +125°C, MB = -55°C to +125°C with MIL STD-883 Class B processing, S = 0°C to +70°C.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

# Z8470 Z80® DART Dual Asynchronous Receiver/Transmitter

# Zilog

## Product Specification

September 1983



**Features**

- Two independent full-duplex channels with separate modem controls. Modem status can be monitored.
- In x1 clock mode, data rates are 0 to 500K bits/second with a 2.5 MHz clock, or 0 to 800K bits/second with a 4.0 MHz clock.
- Receiver data registers are quadruply buffered; the transmitter is doubly buffered.
- Programmable options include 1, 1½ or 2 stop bits; even, odd or no parity; and x1, x16, x32 and x64 clock modes.
- Break generation and detection as well as parity-, overrun- and framing-error detection are available.
- Interrupt features include a programmable interrupt vector, a "status affects vector" mode for fast interrupt processing, and the standard Z-80 peripheral daisy-chain interrupt structure that provides automatic interrupt vectoring with no external logic.
- On-chip logic for ring indication and carrier-detect status.

**Description**

The Z-80 DART (Dual-Channel Asynchronous Receiver/Transmitter) is a dual-channel multi-function peripheral component that satisfies a wide variety of asynchronous serial data communications requirements in microcomputer systems. The Z-80 DART is used as a serial-to-parallel, parallel-to-serial converter/controller in asynchronous applications. In addition, the device also provides modem controls for both channels. In applications where modem controls are not needed, these lines can be used for general-purpose I/O.

Zilog also offers the Z-80 SIO, a more versatile device that provides synchronous (Bisync, HDLC and SDLC) as well as asynchronous operation.

The Z-80 DART is fabricated with n-channel silicon-gate depletion-load technology, and is packaged in a 40-pin plastic or ceramic DIP.

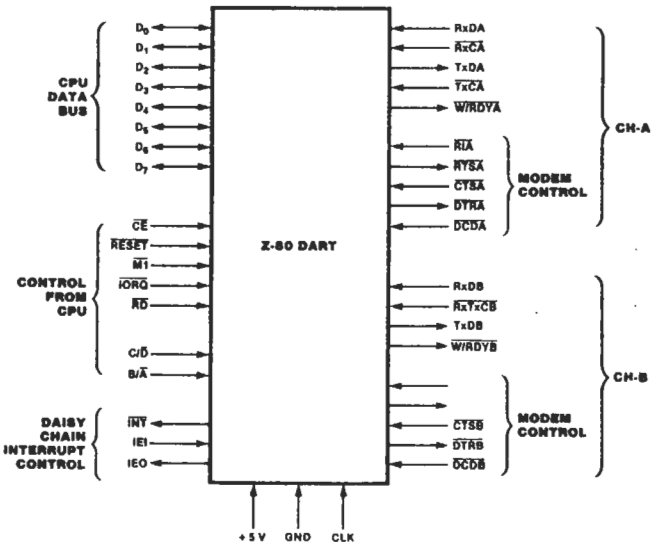



Figure 1. Z80 DART Pin Functions

Figure 2. Pin Assignments



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Pin Description**

**B/$\overline{A}$.** *Channel A Or B Select* (input, High selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the Z-80 DART.

**C/$\overline{D}$.** *Control Or Data Select* (input, High selects Control). This input specifies the type of information (control or data) transferred on the data bus between the CPU and the Z-80 DART.

**$\overline{CE}$.** *Chip Enable* (input, active Low). A Low at this input enables the Z-80 DART to accept command or data input from the CPU during a write cycle, or to transmit data to the CPU during a read cycle.

**CLK.** *System Clock* (input). The Z-80 DART uses the standard Z-80 single-phase system clock to synchronize internal signals.

**$\overline{CTSA}$, $\overline{CTSB}$.** *Clear To Send* (inputs, active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals.

**$D_0$-$D_7$.** *System Data Bus* (bidirectional, 3-state) transfers data and commands between the CPU and the Z-80 DART.

**$\overline{DCDA}$, $\overline{DCDB}$.** *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if the Z-80 DART is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered.

**$\overline{DTRA}$, $\overline{DTRB}$.** *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be programmed as general-purpose outputs.

**IEI.** *Interrupt Enable In* (input, active High) is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

**IEO.** *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this Z-80 DART. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

**$\overline{INT}$.** *Interrupt Request* (output, open drain, active Low). When the Z-80 DART is requesting an interrupt, it pulls $\overline{INT}$ Low.

**$\overline{M1}$.** *Machine Cycle One* (input from Z-80 CPU, active Low). When $\overline{M1}$ and $\overline{RD}$ are both active, the Z-80 CPU is fetching an instruction from memory; when $\overline{M1}$ is active while $\overline{IORQ}$ is active, the Z-80 DART accepts $\overline{M1}$ and $\overline{IORQ}$ as an interrupt acknowledge if the Z-80 DART is the highest priority device that has interrupted the Z-80 CPU.

**$\overline{IORQ}$.** *Input/Output Request* (input from CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$ and $\overline{RD}$ to transfer commands and data between the CPU and the Z-80 DART. When $\overline{CE}$, $\overline{RD}$ and $\overline{IORQ}$ are all active, the channel selected by B/$\overline{A}$ transfers data to the CPU (a read operation). When $\overline{CE}$ and $\overline{IORQ}$ are active, but $\overline{RD}$ is inactive, the channel selected by B/$\overline{A}$ is written to by the CPU with either data or control information as specified by C/$\overline{D}$.

**$\overline{RxCA}$, $\overline{RxCB}$.** *Receiver Clocks* (inputs). Receive data is sampled on the rising edge of $\overline{RxC}$. The Receive Clocks may be 1, 16, 32 or 64 times the data rate.

**$\overline{RD}$.** *Read Cycle Status.* (input from CPU, active Low). If $\overline{RD}$ is active, a memory or I/O read operation is in progress.

**RxDA, RxDB.** *Receive Data* (inputs, active High).

**$\overline{RESET}$.** *Reset* (input, active Low). Disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High and disables all interrupts.

**$\overline{RIA}$, $\overline{RIB}$.** *Ring Indicator* (inputs, Active Low). These inputs are similar to $\overline{CTS}$ and $\overline{DCD}$. The Z-80 DART detects both logic level transitions and interrupts the CPU. When not used in switched-line applications, these inputs can be used as general-purpose inputs.

**$\overline{RTSA}$, $\overline{RTSB}$.** *Request to Send* (outputs, active Low). When the RTS bit is set, the $\overline{RTS}$ output goes Low. When the RTS bit is reset, the output goes High after the transmitter empties.

**$\overline{TxCA}$, $\overline{TxCB}$.** *Transmitter Clocks* (inputs). TxD changes on the falling edge of $\overline{TxC}$. The Transmitter Clocks may be 1, 16, 32 or 64 times the data rate; however, the clock multiplier for the transmitter and the receiver must be the same. The Transmit Clock inputs are Schmitt-trigger buffered. Both the Receiver and Transmitter Clocks may be driven by the Z-80 CTC Counter Time Circuit for programmable baud rate generation.

**TxDA, TxDB.** *Transmit Data* (outputs, active High).

**$\overline{W/RDYA}$, $\overline{W/RDYB}$.** *Wait/Ready* (outputs, open drain when programmed for Wait function, driven High and Low when programmed for Ready function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the Z-80 DART data rate. The reset state is open drain.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Functional Description

The functional capabilities of the Z-80 DART can be described from two different points of view: as a data communications device, it transmits and receives serial data, and meets the requirements of asynchronous data communications protocols; as a Z-80 family peripheral, it interacts with the Z-80 CPU and other Z-80 peripheral circuits, and shares the data, address and control buses, as well as being a part of the Z-80 interrupt structure. As a peripheral to other microprocessors, the Z-80 DART offers valuable features such as non-vectored interrupts, polling and simple handshake capability.

The first part of the following functional description introduces Z-80 DART data communications capabilities; the second part describes the interaction between the CPU and the Z-80 DART.

The Z-80 DART offers RS-232 serial communications support by providing device signals for external modem control. In addition to dual-channel Request To Send, Clear To Send, and Data Carrier Detect ports, the Z-80 DART also features a dual channel Ring Indicator (RIA, RIB) input to facilitate local/remote or station-to-station communication capability.

**Communications Capabilities.** The Z-80 DART provides two independent full-duplex channels for use as an asynchronous receiver/transmitter. The following is a short description of receiver/transmitter capabilities. For more details, refer to the Asynchronous Mode section of the *Z-80 SIO Technical Manual*. The Z-80 DART offers transmission and reception of five to eight bits per character, plus optional even or odd parity. The transmitter can supply one, one and a half or two stop bits per character and can provide a break output at any time. The receiver break detection logic interrupts the CPU both at the start and end of a received break. Reception is protected from spikes by a transient spike rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the Receive Data input. If the Low does not persist—as in the case of a transient—the character assembly process is not started.

Framing errors and overrun errors are detected and buffered together with the character on which they occurred. Vectored interrupts allow fast servicing of interrupting conditions using dedicated routines. Furthermore, a built-in checking process avoids interpreting a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit is begun.

The Z-80 DART does not require symmetric Transmit and Receive Clock signals—a feature that allows it to be used with a Z-80 CTC or any other clock source. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32 or 1/64 of the clock rate supplied to the Receive and Transmit Clock inputs. When using Channel B, the bit rates for transmit and receive operations must be the same because $\overline{RxC}$ and $\overline{TxC}$ are bonded together ($\overline{RxTxCB}$).

**I/O Interface Capabilities.** The Z-80 DART offers the choice of Polling, Interrupt (vectored or non-vectored) and Block Transfer modes to transfer data, status and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

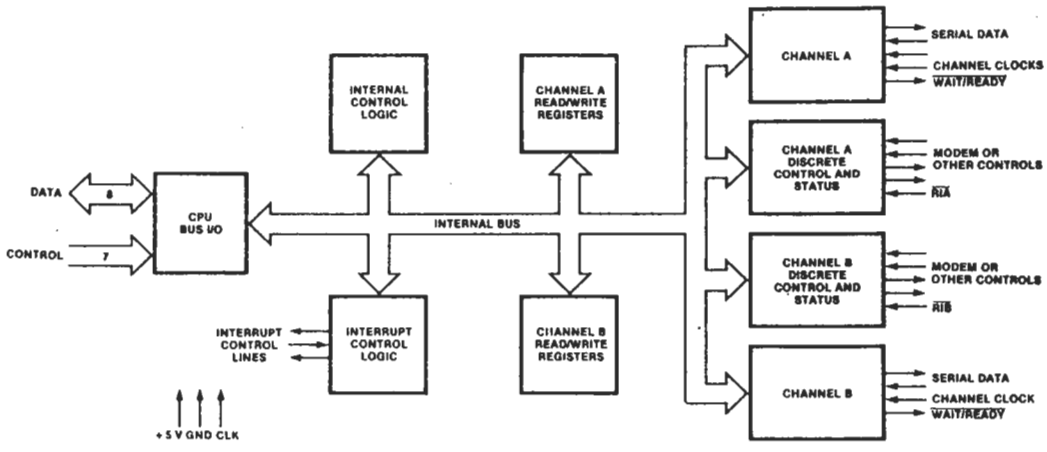



**Figure 3. Block Diagram**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Functional Description** (Continued)

*POLLING.* There are no interrupts in the Polled mode. Status registers RR0 and RR1 are updated at appropriate times for each function being performed. All the interrupt modes of the Z-80 DART must be disabled to operate the device in a polled environment.

While in its Polling sequence, the CPU examines the status contained in RR0 for each channel; the RR0 status bits serve as an acknowledge to the Poll inquiry. The two RR0 status bits $D_0$ and $D_2$ indicate that a data transfer is needed. The status also indicates Error or other special status conditions (see "Z-80 DART Programming"). The Special Receive Condition status contained in RR1 does not have to be read in a Polling sequence because the status bits in RR1 are accompanied by a Receive Character Available status in RR0.

*INTERRUPTS.* The Z-80 DART offers an elaborate interrupt scheme that provides fast interrupt response in real-time applications. As a member of the Z-80 family, the Z-80 DART can be daisy-chained along with other Z-80 peripherals for peripheral interrupt-priority resolution. In addition, the internal interrupts of the Z-80 DART are nested to prioritize the various interrupts generated by Channels A and B. Channel B registers WR2 and RR2 contain the interrupt vector that points to an interrupt service routine in the memory. To eliminate the necessity of writing a status analysis routine, the Z-80 DART can modify the interrupt vector in RR2 so it points directly to one of eight interrupt service routines. This is done under program control by setting a program bit (WR1, $D_2$) in Channel B called "Status Affects Vector." When this bit is set, the interrupt vector in RR2 is modified according to the assigned priority of the various interrupting conditions.

Transmit interrupts, Receive interrupts and External/Status interrupts are the main sources of interrupts. Each interrupt source is enabled under program control with Channel A having a higher priority than Channel B, and with Receiver, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted by the transmit buffer *becoming* empty. (This implies that the transmitter must have had a data character written into it so it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on the first received character
- Interrupt on all received characters
- Interrupt on a Special Receive condition

Interrupt On First Character is typically used with the Block Transfer mode. Interrupt On All Receive Characters can optionally modify the interrupt vector in the event of a parity error. The Special Receive Condition interrupt can occur on a character basis. The Special Receive condition can cause an interrupt only if the Interrupt On First Receive Character or Interrupt On All Receive Characters mode is selected. In Interrupt On First Receive Character, an interrupt can occur from Special Receive conditions (except Parity Error) after the first receive character interrupt (example: Receive Overrun interrupt).

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$ and $\overline{RI}$ pins; however, an External/Status interrupt is also caused by the detection of a Break sequence in the data stream. The interrupt caused by the Break sequence has a special feature that allows the Z-80 DART to interrupt when the Break sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Break condition.

*CPU/DMA BLOCK TRANSFER.* The Z-80 DART provides a Block Transfer mode to accommodate CPU block transfer functions and DMA block transfers (Z-80 DMA or other designs). The Block Transfer mode uses the $\overline{W/RDY}$ output in conjunction with the Wait/Ready bits of Write Register 1. The $\overline{W/RDY}$ output can be defined under software control as a Wait line in the CPU Block Transfer mode or as a Ready line in the DMA Block Transfer mode.

To a DMA controller, the Z-80 DART Ready output indicates that the Z-80 DART is ready to transfer data to or from memory. To the CPU, the Wait output indicates that the Z-80 DART is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Internal Architecture**

The device internal structure includes a Z-80 CPU interface, internal control and interrupt logic, and two full-duplex channels. Each channel contains read and write registers, and discrete control and status logic that provides the interface to modems or other external devices.

The read and write register group includes five 8-bit control registers and two status registers. The interrupt vector is written into an additional 8-bit register (Write Register 2) in Channel B that may be read through Read Register 2 in Channel B. The registers for both channels are designated as follows:

WR0-WR5 — Write Registers 0 through 5
RR0-RR2 — Read Registers 0 through 2

The bit assignment and functional grouping of each register is configured to simplify and organize the programming process.

The logic for both channels provides formats, bit synchronization and validation for data transferred to and from the channel interface. The modem control inputs Clear to Send ($\overline{CTS}$), Data Carrier Detect ($\overline{DCD}$) and Ring Indicator ($\overline{RI}$) are monitored by the control logic under program control. All the modem control signals are general purpose in nature and can be used for functions other than modem control.

For automatic interrupt vectoring, the interrupt control logic determines which channel and which device within the channel has the highest priority. Priority is fixed with Channel A assigned a higher priority than Channel B; Receive, Transmit and External/Status interrupts are prioritized in that order within each channel.

**Data Path.** The transmit and receive data path illustrated for Channel A in Figure 4 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service a Receive Character Available interrupt in a high-speed data transfer.

The transmitter has an 8-bit transmit data register that is loaded from the internal data bus, and a 9-bit transmit shift register that is loaded from the transmit data register.

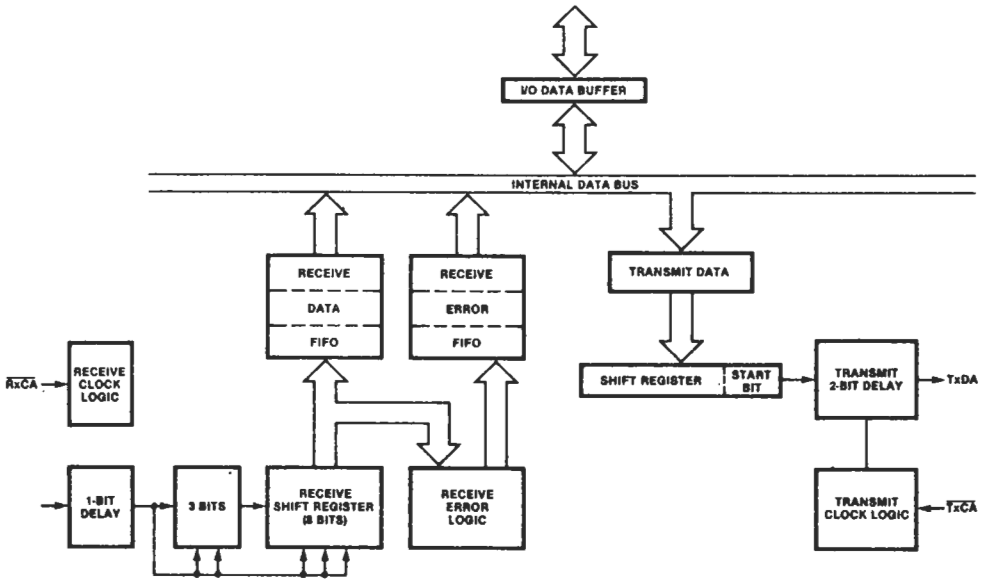



**Figure 4. Data Path**



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Read, Write and Interrupt Timing**

**Read Cycle.** The timing signals generated by a Z-80 CPU input instruction to read a Data or Status byte from the Z-80 DART are illustrated in Figure 5a.

**Write Cycle.** Figure 5b illustrates the timing and data signals generated by a Z-80 CPU output instruction to write a Data or Control byte into the Z-80 DART.

**Interrupt Acknowledge Cycle.** After receiving an Interrupt Request signal ($\overline{INT}$ pulled Low), the Z-80 CPU sends an Interrupt Acknowledge signal ($\overline{M1}$ and $\overline{IORQ}$ both Low). The daisy-chained interrupt circuits determine the highest priority interrupt requestor. The IEI of the highest priority peripheral is terminated High. For any peripheral that has no interrupt pending or under service, IEO = IEI. Any peripheral that does have an interrupt pending or under service forces its IEO Low.

To insure stable conditions in the daisy chain, all interrupt status signals are prevented from changing while $\overline{M1}$ is Low. When $\overline{IORQ}$ is Low, the highest priority interrupt requestor (the one with IEI High) places its interrupt vector on the data bus and sets its internal interrupt-under-service latch.

Refer to the *Z-80 SIO Technical Manual* for additional details on the interrupt daisy chain and interrupt nesting.

**Return From Interrupt Cycle.** Normally, the Z-80 CPU issues an RETI (Return From Interrupt) instruction at the end of an interrupt service routine. RETI is a 2-byte opcode (ED-4D) that resets the interrupt-under-service latch to terminate the interrupt that has just been processed.

When used with other CPUs, the Z-80 DART allows the user to return from the interrupt cycle with a special command called "Return From Interrupt" in Write Register 0 of Channel A. This command is interpreted by the Z-80 DART in exactly the same way it would interpret an RETI command on the data bus.

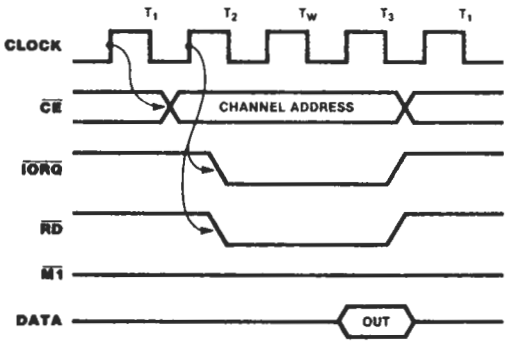



Figure 5a. Read Cycle

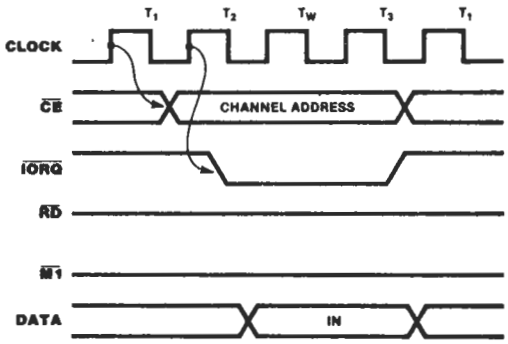



Figure 5b. Write Cycle

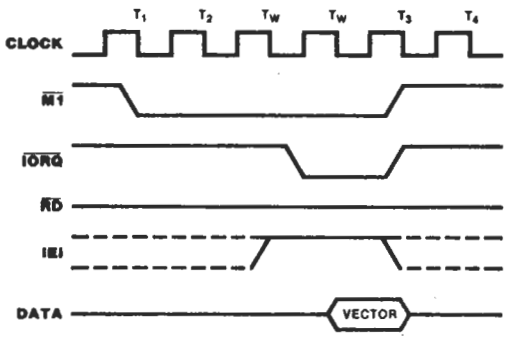



Figure 5c. Interrupt Acknowledge Cycle

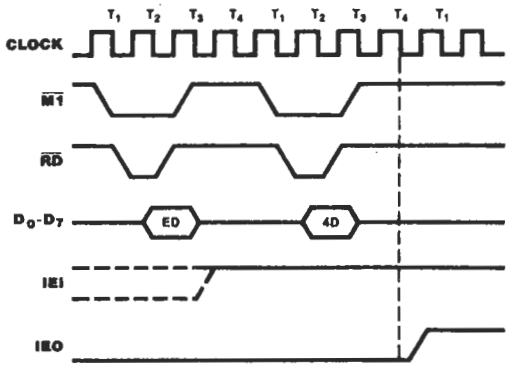



Figure 5d. Return from Interrupt Cycle





"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Z-80 DART Programming**

To program the Z-80 DART, the system program first issues a series of commands that initialize the basic mode and then other commands that qualify conditions within the selected mode. For example, the character length, clock rate, number of stop bits, even or odd parity are first set, then the Interrupt mode and, finally, receiver or transmitter enable.

Both channels contain command registers that must be programmed via the system program prior to operation. The Channel Select input (B/$\overline{A}$) and the Control/Data input (C/$\overline{D}$) are the command structure addressing controls, and are normally controlled by the CPU address bus.

**Write Registers.** The Z-80 DART contains six registers (WR0-WR5) in each channel that are programmed separately by the system program to configure the functional personality of the channels (Figure 4). With the exception of WR0, programming the write registers requires two bytes. The first byte contains three bits ($D_0$-$D_2$) that point to the selected register; the second byte is the actual control word that is written into the register to configure the Z-80 DART.

WR0 is a special case in that all the basic commands ($CMD_0$-$CMD_2$) can be accessed with a single byte. Reset (internal or external) initializes the pointer bits $D_0$-$D_2$ to point to WR0. This means that a register cannot be pointed to in the same operation as a channel reset.

| Write Register Functions | |
|---|---|
| WR0 | Register pointers, initialization commands for the various modes, etc. |
| WR1 | Transmit/Receive interrupt and data transfer mode definition. |
| WR2 | Interrupt vector (Channel B only) |
| WR3 | Receive parameters and control |
| WR4 | Transmit/Receive miscellaneous parameters and modes |
| WR5 | Transmit parameters and controls |

**Read Registers.** The Z-80 DART contains three registers (RR0-RR2) that can be read to obtain the status information for each channel (except for RR2, which applies to Channel B only). The status information includes error conditions, interrupt vector and standard communications-interface signals.

To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing an input instruction, the contents of the addressed read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

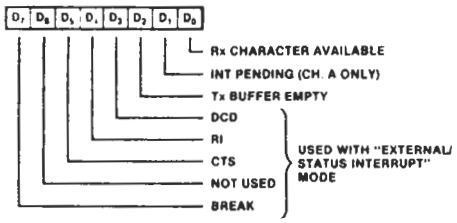
| Read Register Functions | |
|---|---|
| RR0 | Transmit/Receive buffer status, interrupt status and external status |
| RR1 | Special Receive Condition status |
| RR2 | Modified interrupt vector (Channel B only) |



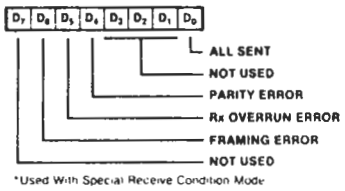
"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## Z-80 DART Read and Write Registers

### READ REGISTER 0

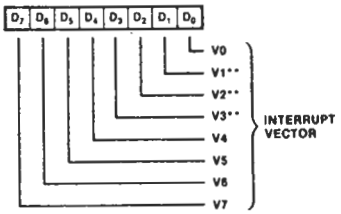



### READ REGISTER 1*



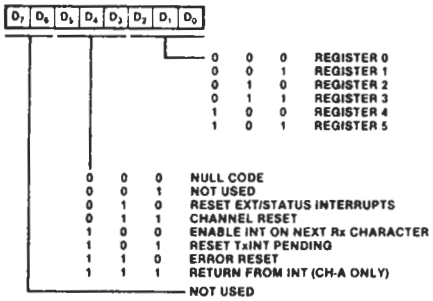

*Used With Special Receive Condition Mode

### READ REGISTER 2

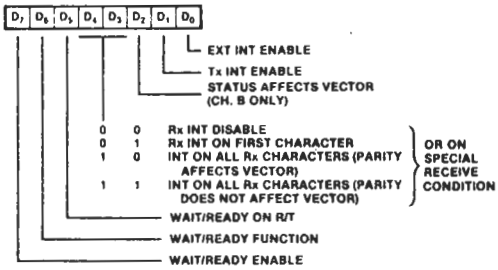



**Variable If "Status Affects Vector" Is Programmed

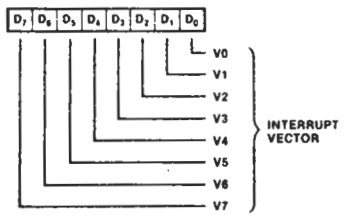
### WRITE REGISTER 0



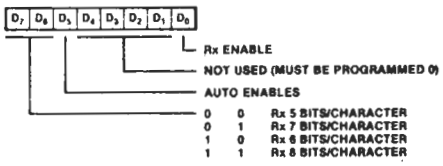

### WRITE REGISTER 1



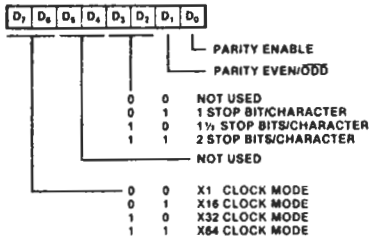

### WRITE REGISTER 2 (CHANNEL B ONLY)



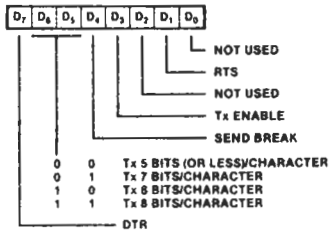

### WRITE REGISTER 3




### WRITE REGISTER 4




### WRITE REGISTER 5






"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**Absolute Maximum Ratings**

Voltages on all inputs and outputs with respect to GND . . . . . -0.3 V to +7.0 V
Operating Ambient Temperature . . . . . As Specified in Ordering Information
Storage Temperature . . . . . -65°C to +150°C

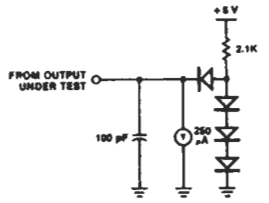
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Test Conditions**

The characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND (0 V). Positive current flows into the referenced pin. Available operating temperature ranges are:

- S* = 0°C to +70°C, $+4.75\ V \le V_{CC} \le +5.25\ V$
- E* = −40°C to +85°C, $+4.75\ V \le V_{CC} \le +5.25\ V$
- M* = −55°C to +125°C, $+4.5\ V \le V_{CC} \le +5.5\ V$

*See Ordering Information section for package temperature range and product number.




**DC Characteristics**

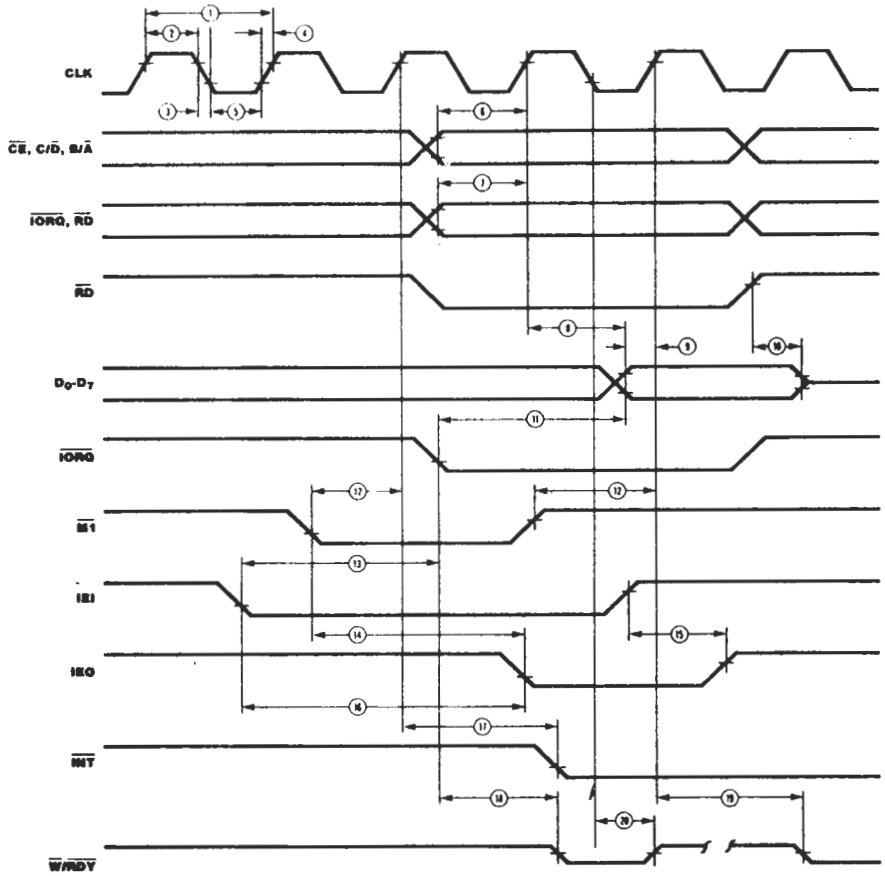
| Symbol | Parameter | Min | Max | Unit | Test Condition |
|---|---|---|---|---|---|
| $V_{ILC}$ | Clock Input Low Voltage | -0.3 | +0.45 | V | |
| $V_{IHC}$ | Clock Input High Voltage | $V_{CC}$-0.6 | +5.5 | V | |
| $V_{IL}$ | Input Low Voltage | -0.3 | +0.8 | V | |
| $V_{IH}$ | Input High Voltage | +2.0 | +5.5 | V | |
| $V_{OL}$ | Output Low Voltage | | +0.4 | V | $I_{OL}$ = 2.0 mA |
| $V_{OH}$ | Output High Voltage | +2.4 | | V | $I_{OH}$ = -250 µA |
| $I_L$ | Input/3-State Output Leakage Current | -10 | +10 | µA | 0.4<V<2.4V |
| $I_{L(RI)}$ | $\overline{RI}$ Pin Leakage Current | -40 | +10 | µA | 0.4<V<2.4V |
| $I_{CC}$ | Power Supply Current | | 100 | mA | |

$T_A$ = 0°C to 70°C, $V_{CC}$ = +5V, ±5%



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

## AC Electrical Characteristics



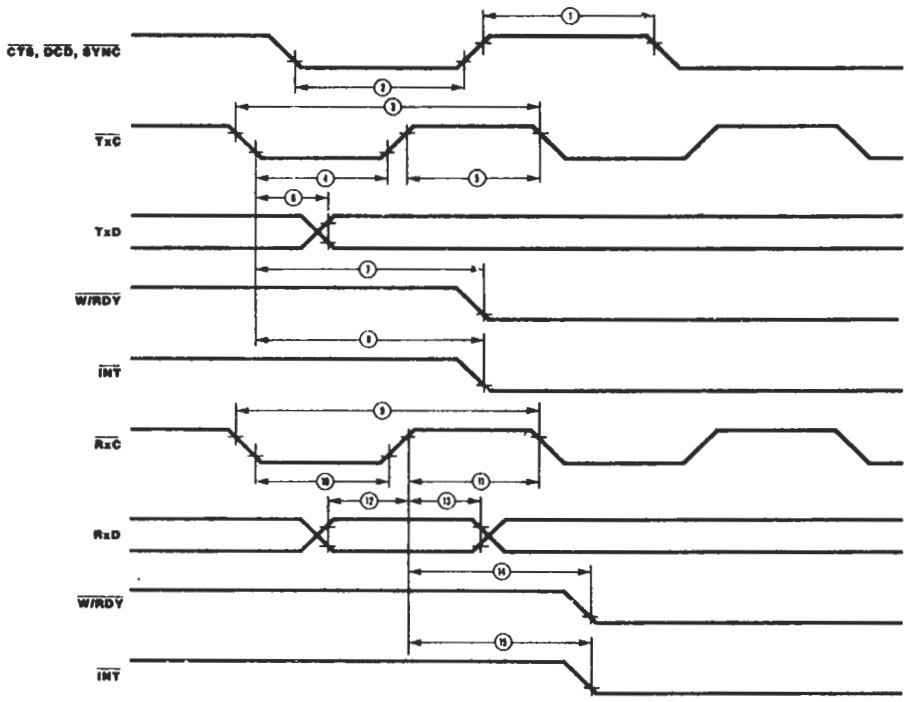

| Number | Symbol | Parameter | Z-80 DART Min | Z-80 DART Max | Z-80A DART Min | Z-80A DART Max | Z-80B DART*† Min | Z-80B DART*† Max |
|---|---|---|---|---|---|---|---|---|
| 1 | TcC | Clock Cycle Time | 400 | 4000 | 250 | 4000 | 165 | 4000 |
| 2 | TwCh | Clock Width (High) | 170 | 2000 | 105 | 2000 | 70 | 2000 |
| 3 | TfC | Clock Fall Time | | 30 | | 30 | | 15 |
| 4 | TrC | Clock Rise Time | | 30 | | 30 | | 15 |
| 5 | TwCl | Clock Width (Low) | 170 | 2000 | 105 | 2000 | 70 | 2000 |
| 6 | TsAD(C) | $\overline{CE}$, C/$\overline{D}$, B/$\overline{A}$ to Clock ↑ Setup Time | 160 | | 145 | | 60 | |
| 7 | TsCS(C) | $\overline{IORQ}$, $\overline{RD}$ to Clock ↑ Setup Time | 240 | | 115 | | 60 | |
| 8 | TdC(DO) | Clock ↑ to Data Out Delay | | 240 | | 220 | | 150 |
| 9 | TsDI(C) | Data In to Clock ↑ Setup Time (Write or M1 Cycle) | 50 | | 50 | | 30 | |
| 10 | TdRD(DOz) | $\overline{RD}$ ↑ to Data Out Float Delay | | 230 | | 110 | | 90 |
| 11 | TdIO(DOI) | $\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle) | | 340 | | 160 | | 100 |
| 12 | TsM1(C) | $\overline{M1}$ to Clock ↑ Setup Time | 210 | | 90 | | 75 | |
| 13 | TsIEI(IO) | IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle) | 200 | | 140 | | 120 | |
| 14 | TdM1(IEO) | $\overline{M1}$ ↓ to IEO ↓ Delay (interrupt before M1) | | 300 | | 190 | | 160 |
| 15 | TdIEI(IEOr) | IEI ↑ to IEO ↑ Delay (after ED decode) | | 150 | | 100 | | 70 |
| 16 | TdIEI(IEOf) | IEI ↓ to IEO ↓ Delay | | 150 | | 100 | | 70 |
| 17 | TdC(INT) | Clock ↑ to INT ↓ Delay | | 200 | | 200 | | 150 |
| 18 | TdIO(W/RWf) | $\overline{IORQ}$ ↓ or $\overline{CE}$ ↓ to W/RDY ↓ Delay (Wait Mode) | | 300 | | 210 | | 175 |
| 19 | TdC(W/RR) | Clock ↑ to W/RDY ↓ Delay (Ready Mode) | | 120 | | 120 | | 100 |
| 20 | TdC(W/RWz) | Clock ↓ to W/RDY Float Delay (Wait Mode) | | 150 | | 130 | | 110 |

*All timings are preliminary and subject to change.
†Units in ns.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

**AC Electrical Characteristics** (Continued)




| Number | Symbol | Parameter | Z-80 DART Min | Z-80 DART Max | Z-80A DART Min | Z-80A DART Max | Z-80B DART[1] Min | Z-80B DART[1] Max | Notes† |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TwPh | Pulse Width (High) | 200 | | 200 | | 200 | | 2 |
| 2 | TwPl | Pulse Width (Low) | 200 | | 200 | | 200 | | 2 |
| 3 | TcTxC | $\overline{TxC}$ Cycle Time | 400 | ∞ | 400 | ∞ | 330 | ∞ | 2 |
| 4 | TwTxCl | $\overline{TxC}$ Width (Low) | 180 | ∞ | 180 | ∞ | 100 | ∞ | 2 |
| 5 | TwTxCh | $\overline{TxC}$ Width (High) | 180 | ∞ | 180 | ∞ | 100 | ∞ | 2 |
| 6 | TdTxC(TxD) | $\overline{TxC}$ ↓ to TxD Delay | | 400 | | 300 | | 220 | 2 |
| 7 | TdTxC(W/RRf) | $\overline{TxC}$ ↓ to W/RDY ↑ Delay (Ready Mode) | 5 | 9 | 5 | 9 | 5 | 9 | 3 |
| 8 | TdTxC(INT) | $\overline{TxC}$ ↓ to $\overline{INT}$ ↑ Delay | 5 | 9 | 5 | 9 | 5 | 9 | 3 |
| 9 | TcRxC | $\overline{RxC}$ Cycle Time | 400 | ∞ | 100 | ∞ | 330 | ∞ | 2 |
| 10 | TwRxCl | $\overline{RxC}$ Width (Low) | 180 | ∞ | 180 | ∞ | 100 | ∞ | 2 |
| 11 | TwRxCh | $\overline{RxC}$ Width (High) | 180 | ∞ | 180 | ∞ | 100 | ∞ | 2 |
| 12 | TsRxD(RxC) | RxD to $\overline{RxC}$ ↑ Setup Time (x1 Mode) | 0 | | 0 | | 0 | | 2 |
| 13 | ThRxD(RxC) | RxD Hold Time (x1 Mode) | 140 | | 140 | | 100 | | 2 |
| 14 | TdRxC(W/RRf) | $\overline{RxC}$ ↑ to W/RDY ↑ Delay (Ready Mode) | 10 | 13 | 10 | 13 | 10 | 13 | 3 |
| 15 | TdRxC(INT) | $\overline{RxC}$ ↑ to $\overline{INT}$ ↑ Delay | 10 | 13 | 10 | 13 | 10 | 13 | 3 |

NOTES:
† In all modes, the System Clock rate must be at least five times the maximum data rate. $\overline{RESET}$ must be active a minimum of one complete clock cycle.

1. Timings are preliminary and subject to change.
2. Units in nanoseconds (ns).
3. Units equal to System Clock Periods.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

| Product Number | Package/ Temp | Speed | Description |
|---|---|---|---|
| Z8470 | CE | 2.5 MHz | Z80 DART (40-pin) |
| Z8470 | CM | 2.5 MHz | Same as above |
| Z8470 | CMB | 2.5 MHz | Same as above |
| Z8470 | CS | 2.5 MHz | Same as above |
| Z8470 | DE. | 2.5 MHz | Same as above |
| Z8470 | DS | 2.5 MHz | Same as above |
| Z8470 | PE | 2.5 MHz | Same as above |
| Z8470 | PS | 2.5 MHz | Same as above |
| Z8470A | CE | 4.0 MHz | Z80A DART (40-pin) |
| Z8470A | CM | 4.0 MHz | Same as above |
| Z8470A | CMB | 4.0 MHz | Same as above |
| Z8470A | CS | 4.0 MHz | Z80A DART (40-pin) |
| Z8470A | DE | 4.0 MHz | Same as above |
| Z8470A | DS | 4.0 MHz | Same as above |
| Z8470A | PE | 4.0 MHz | Same as above |
| Z8470A | PS | 4.0 MHz | Same as above |
| Z8470B | CE | 6.0 MHz | Z80B DART (40-pin) |
| Z8470B | CS | 6.0 MHz | Same as above |
| Z8470B | DS | 6.0 MHz | Same as above |
| Z8470B | PS | 6.0 MHz | Same as above |

*NOTES: C = Ceramic, D = Cerdip, P = Plastic, E = -40°C to +85°C, M = -55°C to +125°C, MB = -55°C to +125°C with MIL-STD-883 Class B processing, S = 0°C to +70°C.



"Reproduced by permission ©1983 Zilog, Inc.
This material shall not be reproduced without
the written consent of Zilog, Inc."

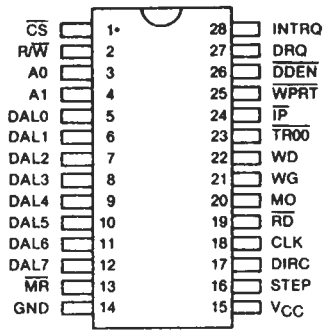
PRELIMINARY

**WESTERN DIGITAL**
C O R P O R A T I O N

# WD1770/1772 5¼" Floppy Disk Controller/Formatter

## FEATURES

- 28 PIN DIP
- SINGLE 5V SUPPLY
- BUILT-IN DATA SEPARATOR
- BUILT-IN WRITE PRECOMPENSATION
- 5¼" SINGLE AND DOUBLE DENSITY
- MOTOR CONTROL
- 128, 256, 512 OR 1024 SECTOR LENGTHS
- TTL COMPATIBLE
- 8 BIT BIDIRECTIONAL DATA BUS
- TWO VERSIONS AVAILABLE
  WD1770 = STANDARD 179X STEP RATES
  WD1772 = FASTER STEP RATES




**PIN DESIGNATION**

## DESCRIPTION

The WD1770 is a MOS/LSI device which performs the functions of a 5¼" Floppy Disk Controller/Formatter. It is similar to its predecessor, the WD179X, but also contains a digital data separator and write precompensation circuitry. The drive side of the interface needs no additional logic except for buffers/receivers. Designed for 5¼" single or double density operation, the device contains a programmable Motor On signal.

The WD1770 is implemented in NMOS silicon gate technology and is available in a 28 pin dual-in-line.

The WD1770 is a low cost version of the FD179X Floppy Disk Controller/Formatter. It is compatible with the 179X, but has a built-in digital data separator and write precompensation circuits. A single read line ($\overline{RD}$, Pin 19) is the only input required to recover serial FM or MFM data from the disk drive. The device has been specifically designed for control of 5¼" floppy disk drives with data rates of 125 KBits/Sec (single density) and 250 KBits/Sec (double density). In addition, write precompensation of 125 Nsec from nominal can be enabled at any point through simple software commands. Another programmable feature, Motor On, has been incorporated to enable the spindle motor automatically prior to operating a selected drive.

Two versions of the WD1770 are available. The standard version is compatible with the 179X stepping rates, while the WD1772 offers stepping rates of 2, 3, 5 and 6 msec.

The processor interface consists of an 8-bit bidirectional bus for transfer of status, data, and commands. All host communication with the drive occurs through these data lines. They are capable of driving one standard TTL load or three "LS" loads.

June, 1983

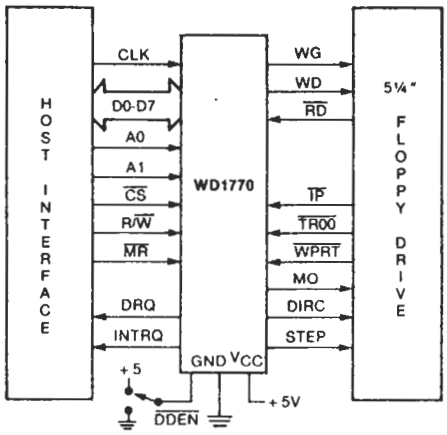
Reproduced with permission from
Western Digital Corporation

| PIN NUMBER | PIN NAME | SYMBOL | FUNCTION |
|---|---|---|---|
| 1 | CHIP SELECT | $\overline{CS}$ | A logic low on this input selects the chip and enable Host communication with the device. |
| 2 | READ/WRITE | R/$\overline{W}$ | A logic high on this input controls the placement of data on the $\overline{D0}$-$\overline{D7}$ lines from a selected register, while a logic low causes a write operation to a selected register. |
| 3,4 | ADDRESS 0,1 | A0, A1 | These two inputs select a register to Read/Write data:<br>$\overline{CS}$ A1 A0 R/$\overline{W}$ = 1 R/$\overline{W}$ = 0<br>0 0 0 Status Reg Command Reg<br>0 0 1 Track Reg Track Reg<br>0 1 0 Sector Reg Sector Reg<br>0 1 1 Data Reg Data Reg |
| 5-12 | DATA ACCESS LINES 0 THROUGH 7 | DAL0-DAL7 | Eight bit bidirectional bus used for transfer of data, control, or status. This bus is enabled by $\overline{CS}$ and R/$\overline{W}$. Each line will drive one TTL load. |
| 13 | MASTER RESET | $\overline{MR}$ | A logic low pulse on this line resets the device and initializes the status register (internal pull-up). |
| 14 | GROUND | GND | Ground. |
| 15 | POWER SUPPLY | $V_{CC}$ | +5V ±5% power supply input. |
| 16 | STEP | STEP | The Step output contains a pulse for each step of the drive's R/$\overline{W}$ head. The WD1770 and WD1772 offer different step rates. |
| 17 | DIRECTION | DIRC | The Direction output is high when stepping in towards the center of the diskette, and low when stepping out. |
| 18 | CLOCK | CLK | This input requires a free-running 50% duty cycle clock (for internal timing) at 8 MHZ ± 1%. |
| 19 | READ DATA | $\overline{RD}$ | This active low input is the raw data line containing both clock and data pulses from the drive. |
| 20 | MOTOR ON | MO | Active high output used to enable the spindle motor prior to read, write or stepping operations. |
| 21 | WRITE GATE | WG | This output is made valid prior to writing on the diskette. |
| 22 | WRITE DATA | WD | FM or MFM clock and data pulses are placed on this line to be written on the diskette. |
| 23 | TRACK 00 | $\overline{TR00}$ | This active low input informs the WD1770 that the drive's R/$\overline{W}$ heads are positioned over Track zero (internal pull-up). |
| 24 | INDEX PULSE | $\overline{IP}$ | This active low input informs the WD1770 when the physical index hole has been encountered on the diskette (internal pull-up). |
| 25 | WRITE PROTECT | $\overline{WPRT}$ | This input is sampled whenever a Write Command is received. A logic low on this line will prevent any Write Command from executing (internal pull-up). |
| 26 | DOUBLE DENSITY ENABLE | $\overline{DDEN}$ | This input pin selects either single (FM) or double (MFM) density. When $\overline{DDEN}$ = 0, double density is selected (internal pull-up). |



Reproduced with permission from
Western Digital Corporation

| PIN NUMBER | PIN NAME | SYMBOL | FUNCTION |
|---|---|---|---|
| 27 | DATA REQUEST | DRQ | This active high output indicates that the Data Register is full (on a Read) or empty (on a Write operation). |
| 28 | INTERRUPT REQUEST | INTRQ | This active high output is set at the completion of any command or reset a read of the Status Register. |




**WD1770 SYSTEM BLOCK DIAGRAM**

## ARCHITECTURE

The Floppy Disk Formatter block diagram is illustrated on page 4. The primary sections include the parallel processor interface and the Floppy Disk interface.

**Data Shift Register** — This 8-bit register assembles serial data from the Read Data input ($\overline{RD}$) during Read operations and transfers serial data to the Write Data output during Write operations.

**Data Register** — This 8-bit register is used as a holding register during Disk Read and Write operations. In Disk Read operations, the assembled data byte is transferred in parallel to the Data Register from the Data Shift Register. In Disk Write operations, information is transferred in parallel from the Data Register to the Data Shift Register.

When executing the Seek command, the Data Register holds the address of the desired Track position. This register is loaded from the DAL and gated onto the DAL under processor control.

**Track Register** — This 8-bit register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in and decremented by one when the head is stepped out (towards track 00). The contents of the register are compared with the recorded track number in the ID field during disk Read, Write, and Verify operations. The Track Register can be loaded from or transferred to the DAL. This Register should not be loaded when the device is busy.

**Sector Register (SR)** — This 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk Read or Write operations. The Sector Register contents can be loaded from or transferred to the DAL. This register should not be loaded when the device is busy.

**Command Register (CR)** — This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

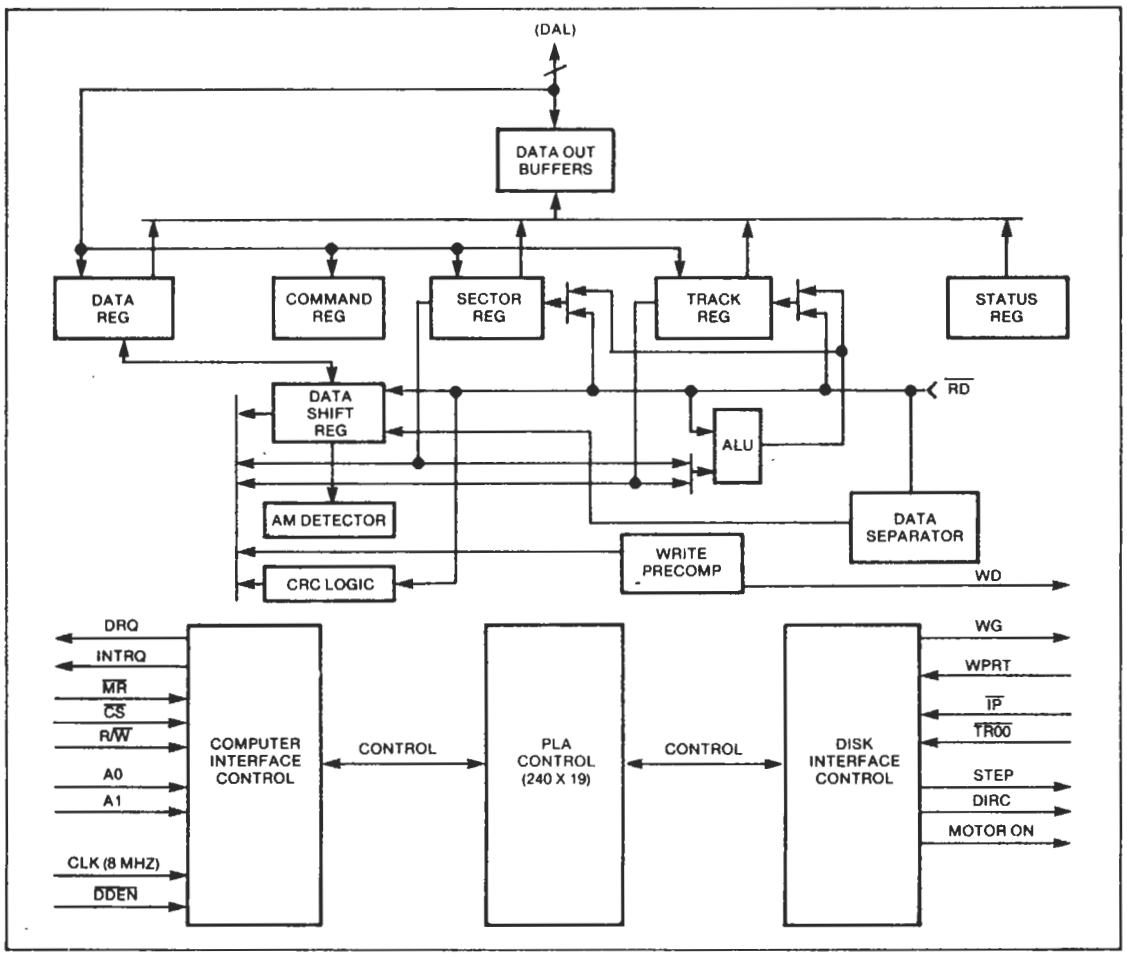
**Status Register (STR)** — This 8-bit register holds device Status information. The meaning of the Status bits is a function of the type of command previously executed. This register can be read onto the DAL, but not loaded from the DAL.

**CRC Logic** — This logic is used to check or to generate the 16-bit Cyclic Redundancy Check (CRC). The polynomial is:
$G(x) = x^{16} + x^{12} + x^5 + 1$.
The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

**Arithmetic/Logic Unit (ALU)** — The ALU is a serial comparator, incrementer, and decrementer and is used for register modification and comparisons with the disk recorded ID field.



Reproduced with permission from
Western Digital Corporation




**WD1770 BLOCK DIAGRAM**

**Timing and Control** — All computer and Floppy Disk interface controls are generated through this logic. The internal device timing is generated from an external crystal clock. The FD1770 has two different modes of operation according to the state of $\overline{DDEN}$. When $\overline{DDEN}$ = 0, double density (MFM) is enabled. When $\overline{DDEN}$ = 1, single density is enabled.

**AM Detector** — The address mark detector detects ID, data and index address marks during read and write operations.

**Data Separator** — A digital data separator consisting of a ring shift register and data window detection logic provides read data and a recovery clock to the AM detector.

## PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines (DAL) and associated control signals. The DAL are used to transfer Data, Status, and Control words out of, or into the WD1770. The DAL are three state buffers that are enabled as output drivers when Chip Select ($\overline{CS}$) and $R/\overline{W}$ = 1 are active or act as input receivers when $\overline{CS}$ and $R/\overline{W}$ = 0 are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and $\overline{CS}$ is made low. The address bits A1 and A0, combined with the signal $R/\overline{W}$ during a Read operation or Write operation are interpreted as selecting the following registers:



Reproduced with permission from
Western Digital Corporation

| A1 · A0 | | READ (R/$\overline{W}$ = 1) | WRITE (R/$\overline{W}$ = 0) |
|---|---|---|---|
| 0 | 0 | Status Register | Command Register |
| 0 | 1 | Track Register | Track Register |
| 1 | 0 | Sector Register | Sector Register |
| 1 | 1 | Data Register | Data Register |

During Direct Memory Access (DMA) types of data transfers between the Data Register of the WD1770 and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operations continues until the end of sector is reached.

On Disk Write operations the Data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

The WD1770 has two modes of operation according to the state $\overline{DDEN}$ (Pin 26). When $\overline{DDEN}$ = 1, single density is selected. In either case, the CLK input (Pin 18) is at 8 MHZ.

## GENERAL DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, $\overline{DDEN}$ should be placed to logical "1." For MFM formats, $\overline{DDEN}$ should be placed to a logical "0." Sector lengths are determined at format time by the fourth byte in the "ID" field.

| SECTOR LENGTH TABLE | |
|---|---|
| SECTOR LENGTH FIELD (HEX) | NUMBER OF BYTES IN SECTOR (DECIMAL) |
| 00 | 128 |
| 01 | 256 |
| 02 | 512 |
| 03 | 1024 |

The number of sectors per tract as far as the WD1770 is concerned can be from 1 to 255 sectors. The number of tracks as far as the WD1770 is concerned is from 0 to 255 tracks.

## GENERAL DISK WRITE OPERATION

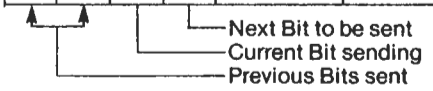
When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the device before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{Write\ Protect}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set.

For Write operations, the WD1770 provides Write Gate (Pin 21) to enable a Write condition, and Write Data (Pin 22) which consists of a series of active high pulses. These pulses contain both Clock and Data information in FM and MFM. Write Data provides the unique missing clock patterns for recording Address Marks.

The Precomp Enable bit in Write commands allow automatic Write precompensation to take place. The outgoing Write Data stream is delayed or advanced from nominal by 125 nanoseconds according to the following table:

| PATTERN | | | | MFM | FM |
|---|---|---|---|---|---|
| X | 1 | 1 | 0 | Early | N/A |
| X | 0 | 1 | 1 | Late | N/A |
| 0 | 0 | 0 | 1 | Early | N/A |
| 1 | 0 | 0 | 0 | Late | N/A |




Precompensation is typically enabled on the innermost tracks where bit shifts usually occur and bit density is at its maximun.

## COMMAND DESCRIPTION

The WD1770 will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force Interrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, an interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Commands and types are summarized in Table 1.



Reproduced with permission from
Western Digital Corporation

## COMMAND SUMMARY

| TYPE | COMMAND | BITS 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| I | Restore | 0 | 0 | 0 | 0 | h | V | $r_1$ | $r_0$ |
| I | Seek | 0 | 0 | 0 | 1 | h | V | $r_1$ | $r_0$ |
| I | Step | 0 | 0 | 1 | u | h | V | $r_1$ | $r_0$ |
| I | Step-in | 0 | 1 | 0 | u | h | V | $r_1$ | $r_0$ |
| I | Step-out | 0 | 1 | 1 | u | h | V | $r_1$ | $r_0$ |
| II | Read Sector | 1 | 0 | 0 | m | h | E | 0 | 0 |
| II | Write Sector | 1 | 0 | 1 | m | h | E | P | $a_0$ |
| III | Read Address | 1 | 1 | 0 | 0 | h | E | 0 | 0 |
| III | Read Track | 1 | 1 | 1 | 0 | h | E | 0 | 0 |
| III | Write Track | 1 | 1 | 1 | 1 | h | E | P | 0 |
| IV | Force Interrupt | 1 | 1 | 0 | 1 | $I_3$ | $I_2$ | $I_1$ | $I_0$ |

## FLAG SUMMARY

### TYPE I COMMANDS

**h = Motor On Flag (Bit 3)**

h = 0, Enable Spin-Up Sequence
h = 1, Disable Spin-Up Sequence

**V = Verify Flag (Bit 2)**

V = 0, No Verify
V = 1, Verify on Destination Track

**$r_1, r_0$ = Stepping Rate (Bits 1, 0)**

| $r_1$ | $r_0$ | WD1770 | WD1772 |
|---|---|---|---|
| 0 | 0 | 6 ms | 2 ms |
| 0 | 1 | 12 ms | 3 ms |
| 1 | 0 | 20 ms | 5 ms |
| 1 | 1 | 30 ms | 6 ms |

**u = Update Flag (Bit 4)**

u = 0, No Update
u = 1, Update Track Register

### TYPE II & III COMMANDS

**m = Multiple Sector Flag (Bit 4)**

m = 0, Single Sector
m = 1, Multiple Sector

**$a_0$ = Data Address Mark (Bit 0)**

$a_0$ = 0, Write Normal Data Mark
$a_0$ = 1, Write Deleted Data Mark

**E = 30ms Settling Delay (Bit 2)**

E = 0, No Delay
E = 1, Add 30ms Delay

**P = Write Precompensation (Bit 1)**

P = 0, Enable Write Precomp
P = 1, Disable Write Precomp

### TYPE IV COMMANDS

**$I_3$-$I_0$ Interrupt Condition (Bits 3-0)**

$I_0$ = 1, Don't Care
$I_1$ = 1, Don't Care
$I_2$ = 1, Interrupt on Index Pulse
$I_3$ = 1, Immediate Interrupt
$I_3$-$I_0$ = 0, Terminate without Interrupt

## TYPE I COMMANDS

The Type I Commands include the Restore, Seek, Step, Step-In, and Step-Out commands. Each of the Type I Commands contains a rate field ($r_0$,$r_1$), which determines the stepping motor rate.

A 4$\mu$s (MFM) or 8 $\mu$s (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output. The chip will step the drive in the same direction it last stepped unless the command changes the direction.

The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 24$\mu$s before the first stepping pulse is generated.

After the last directional step an additional 30 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. There is also a 30 ms head settling time if the E flag is set in any Type II or III command.

When a Seek, Step or Restore command is executed, an optional verification of Read/Write head position can be performed by setting bit 2 (V = 1) in the command word to a logic 1. The verification operation begins at the end of the 30 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. If there is a match but not a valid CRC, the CRC error status bit is set (Status Bit 3), and the next encountered ID field is read from the disk for the verification operation.

The WD1770 must find an ID field with correct track number and correct CRC within 5 revolutions of the media, otherwise the seek error is set and an INTRQ is generated. If V = 0, no verification is performed.

All commands, except the Force Interrupt command, may be programmed via the h Flag to delay for spindle motor start up time. If the h Flag is set and the Motor On line (Pin 20) is low when a command is received, the WD1770 will force Motor On to a logic 1 and wait 6 revolutions before executing the command. At 300 RPM, this guarantees a one second spindle start up time. If after finishing the command, the device remains idle for 10 revolutions, the Motor

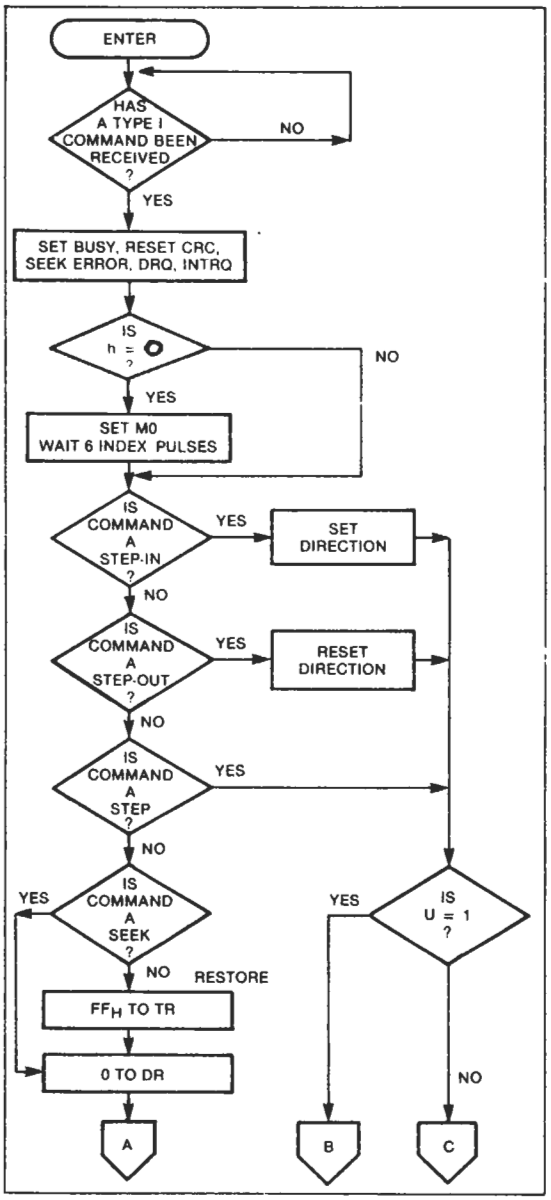


Reproduced with permission from
Western Digital Corporation

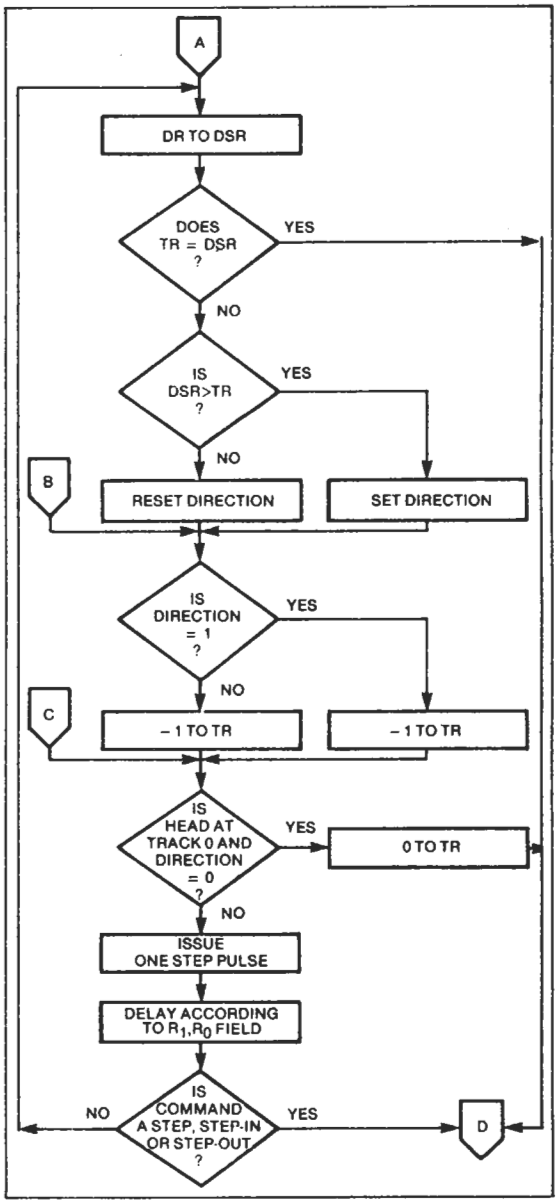
On line will go back to a logic 0. If a command is issued while Motor On is high, the command will execute immediately, defeating the 6 revolution start up. This feature allows consecutive Read or Write commands without waiting for motor start up each time; the WD1770 assumes the spindle motor is up to speed.

**RESTORE (SEEK TRACK 0)**

Upon receipt of this command, the Track 00 ($\overline{TR00}$) input is sampled. If $\overline{TR00}$ is active low indicating the Read/Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TR00}$ is not active low, stepping pulses (Pin 16) at a rate specified by the $r_1,r_0$ field are issued until the $\overline{TR00}$ input is activated.




TYPE I COMMAND FLOW




TYPE I COMMAND FLOW

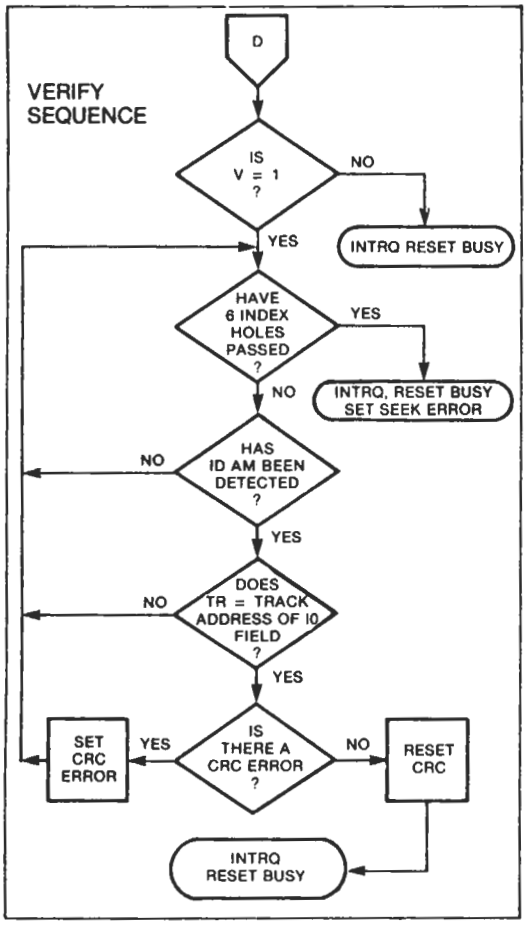


Reproduced with permission from
Western Digital Corporation

At this time, the Track Register is loaded with zeroes and an interrupt is generated. If the $\overline{TR00}$ input does not go active low after 255 stepping pulses, the WD1770 terminates operation, interrupts, and sets the Seek error status bit, providing the V flag is set. A verification operation also takes place if the V flag is set. The h bit allows the Motor On option at the start of command.

### SEEK

This command assumes that the Track Register contains the track number of the current position of the Read/Write head and the Data Register contains the desired track number. The WD1770 will update the Track Register and issue stepping pulses in the appropriate direction until the contents of the Track Register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command. Note: When using multiple drives, the track register must be updated for the drive selected before seeks are issued.




**TYPE I COMMAND FLOW**

### STEP

Upon receipt of this command, the WD1770 issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the $r_1,r_0$ field, a verification takes place if the V flag is on. If the U flag is on, the Track Register is updated. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command.

### STEP-IN

Upon receipt of this command, the WD1770 issues one stepping pulse in the direction towards track 76. If the U flag is on, the Track Register is incremented by one. After a delay determined by the $r_1,r_0$ field, a verification takes place if the V flag is on. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command.

### STEP-OUT

Upon receipt of this command, the WD1770 issues one stepping pulse in the direction towards track 0. If the U flag is on, the Track Register is decremented by one. After delay determined by the $r_1,r_0$ field, a verification takes place if the V flag is on. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command.

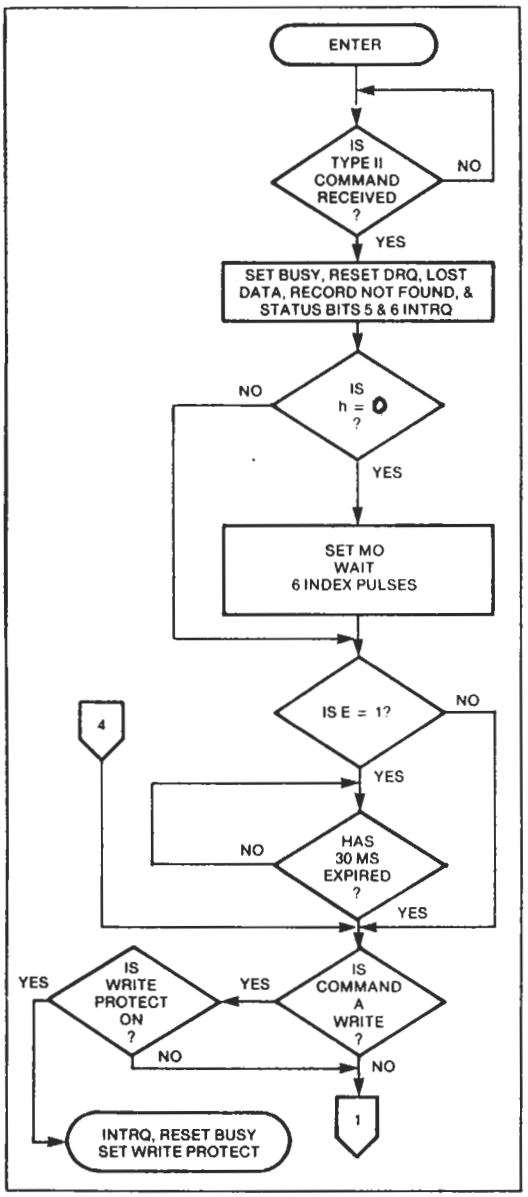
### TYPE II COMMANDS

The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the Sector Register with the desired sector number. Upon receipt of the Type II command, the busy status bit is set. If the E flag = 1 the command will execute after a 30 msec delay.

When an ID field is located on the disk, the WD1770 compares the Track Number on the ID field with the Track Register. If there is not a match, the next encountered ID field is read and a comparison is again made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from depending upon the command. The WD1770 must find an ID field with a Track number, Sector number, and CRC within four revolutions of the disk, other-



Reproduced with permission from
Western Digital Corporation




**TYPE II COMMAND**

wise, the Record not found status bit is set (Status Bit 4) and the command is terminated with an interrupt (INTRQ).

Each of the Type II Commands contains an (m) flag which determines if multiple records (sectors) are to be read or written, depending upon the command. If m = 0, a single sector is read or written and an interrupt is generated at the completion of the command. If m = 1, multiple records are read or written with the sector register internally updated so that an address verification can occur on the next record. The WD1770 will continue to read or write multiple records and update the sector register in numerical ascending sequence until the sector register exceeds the number of sectors on the track or until the Force Interrupt command is loaded into the Command Register, which terminates the command and generates an interrupt.

For example: If the WD1770 is instructed to read sector 27 and there are only 26 on the track, the sector register exceeds the number available. The WD1770 will search for 5 disk revolutions, interrupt out, reset busy, and set the record not found status bit.

### READ SECTOR

Upon receipt of the Read Sector command, the Busy status bit is set, and when a ID field is encountered that has the correct track number, correct sector number, and correct CRC, the data field is presented to the computer. The Data Address Mark of the data field must be found within 30 bytes in single density and 43 bytes in double density of the last ID field CRC byte; if not, the ID field is searched for and verified again followed by the Data Address Mark search. If after 5 revolutions the DAM cannot be found, the Record Not Found status bit is set and the operation is terminated. When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the computer has not read the previous contents of the DR before a new character is transferred that character is lost and the Lost Data Status bit is set. This sequence continues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple record command).

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown:

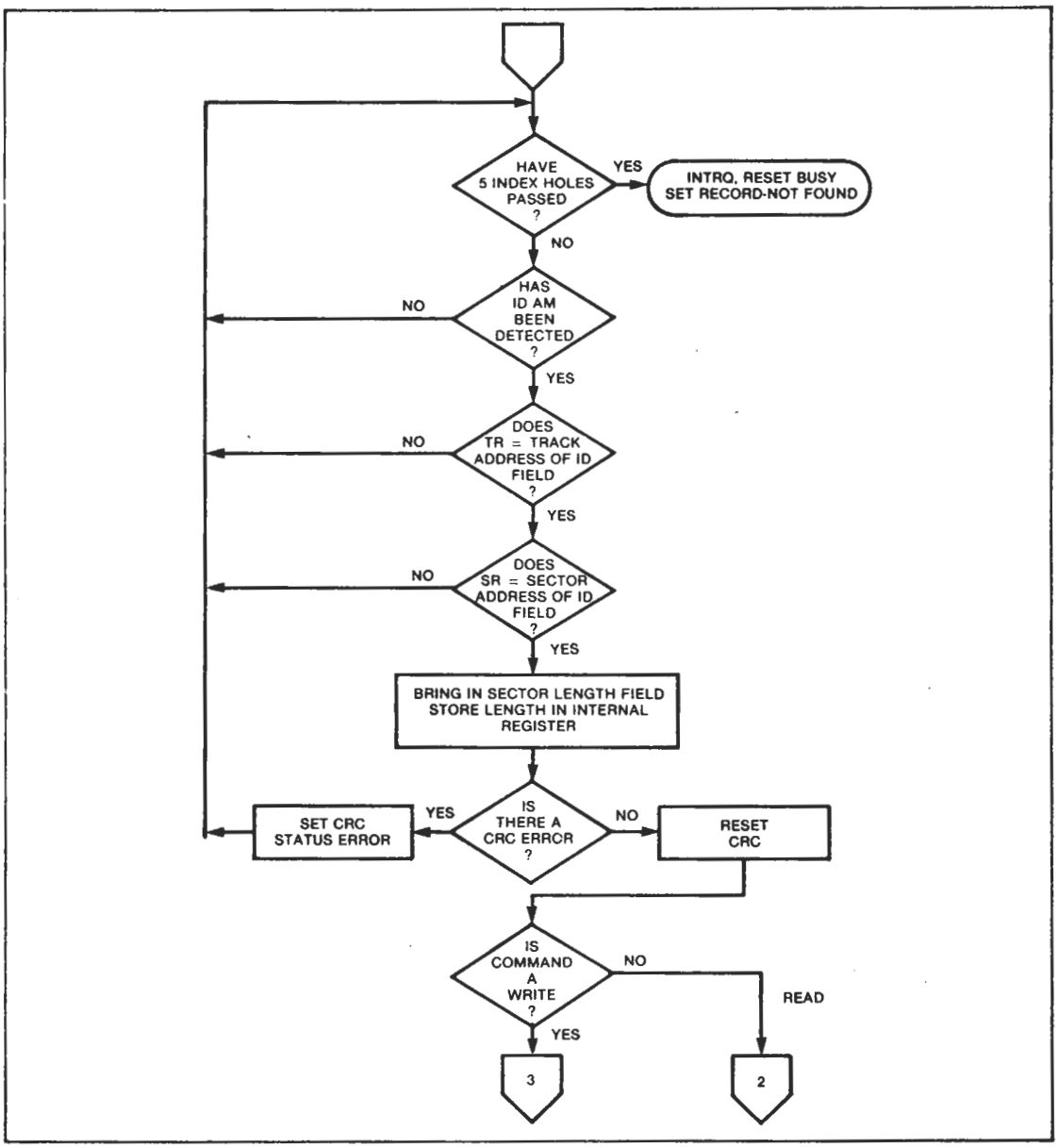
| STATUS BIT 5 | |
|---|---|
| 1 | Deleted Data Mark |
| 0 | Data Mark |

### WRITE SECTOR

Upon receipt of the Write Sector command, the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, and correct CRC, a DRQ is generated. The WD1770 counts off 11 bytes in single density and 22 bytes in double density from the CRC field and the Write Gate (WG) output is made active if the DRQ is serviced (i.e., the DR has been loaded by the computer). If DRQ has not been serviced, the command is terminated



Reproduced with permission from
Western Digital Corporation




**TYPE II COMMAND**

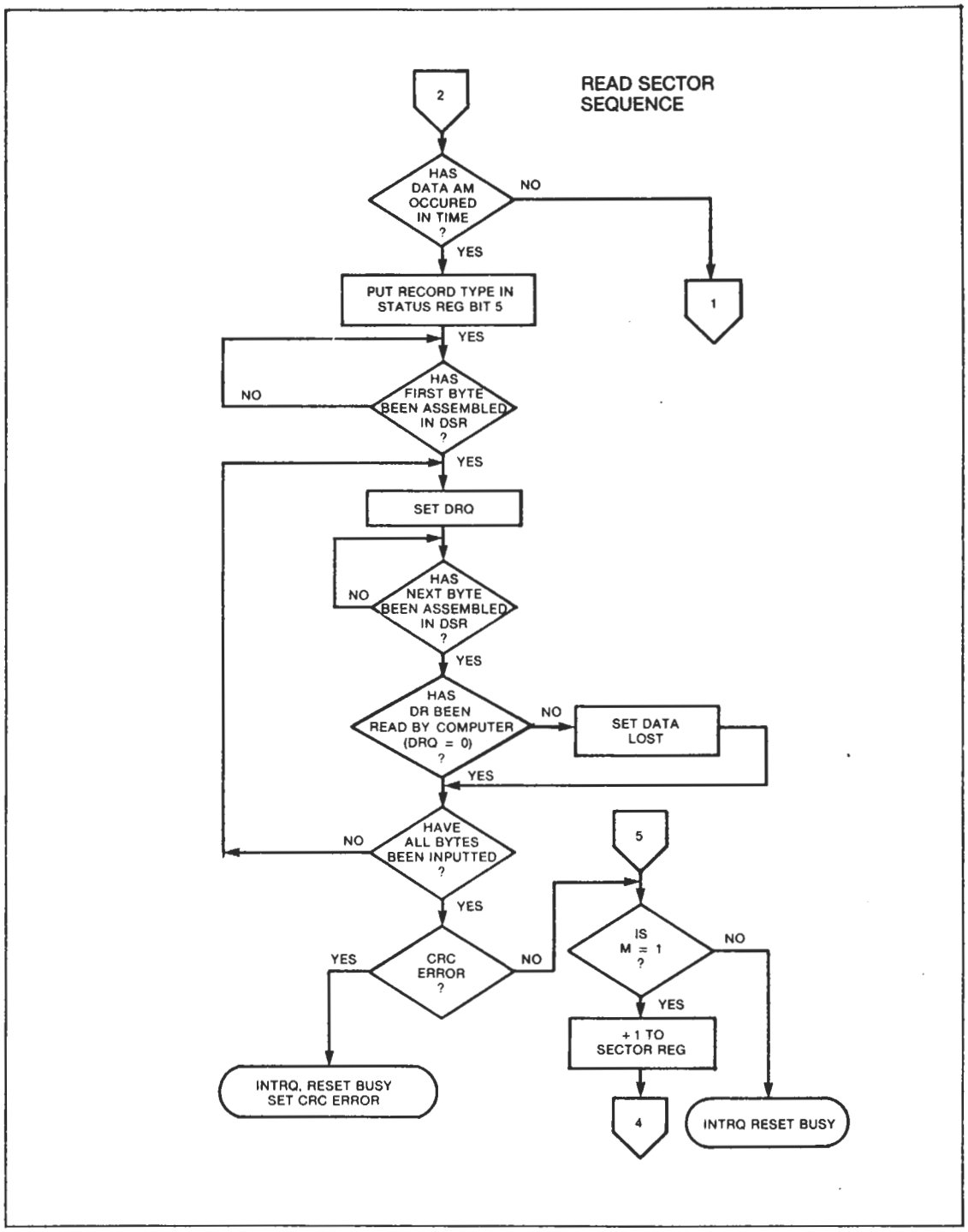
and the Lost Data status bit is set. If the DRQ has been serviced, the WG is made active and six bytes of zeroes in single density and 12 bytes in double density are then written on the disk. At this time, the Data Address Mark is then written on the disk as determined by the $a_0$ field of the command as shown below:

| $a_0$ | DATA ADDRESS MARK (BIT 0) |
|---|---|
| 1 | Deleted Data Mark |
| 0 | Data Mark |

The WD1770 then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit



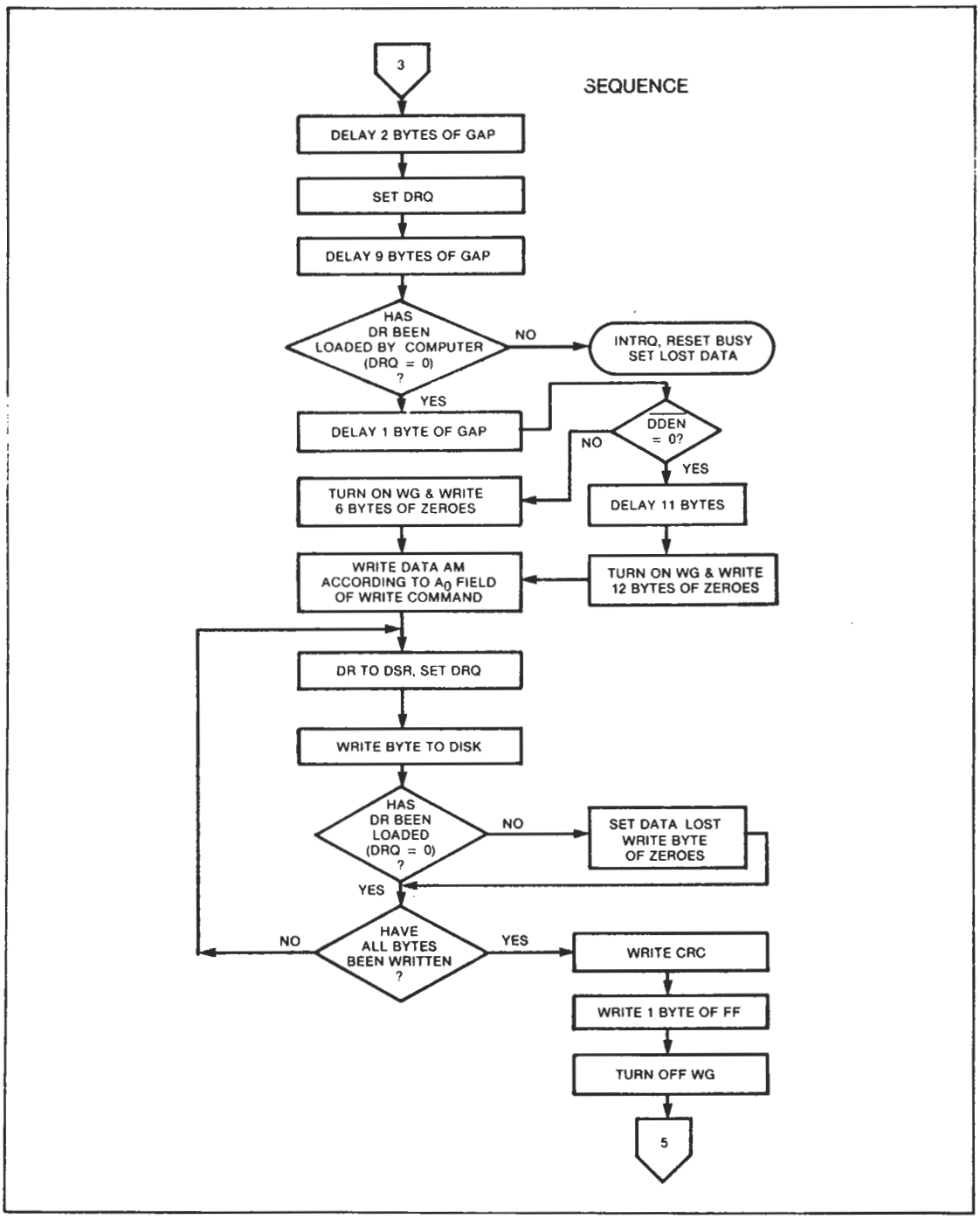
Reproduced with permission from
Western Digital Corporation




**TYPE II COMMAND**



Reproduced with permission from
Western Digital Corporation




**TYPE II COMMAND**



Reproduced with permission from
Western Digital Corporation

is set and a byte of zeroes is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one byte of logic ones in FM or in MFM. The WG output is then deactivated. INTRQ will set 24μsec (MFM) after the last CRC byte is written. For partial sector writing, the proper method is to write data and fill the balance with zeroes.

## TYPE III COMMANDS

### Read Address

Upon receipt of the Read Address command, the Busy Status Bit is set. The next encountered ID field is then read in from the disk, and six data bytes of the ID field are assembled and transferred to the DR, and a DRQ is generated for each byte. The six bytes of the ID field are shown below:

| TRACK ADDR | SIDE NUMBER | SECTOR ADDRESS | SECTOR LENGTH | CRC 1 | CRC 2 |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |

Although the CRC characters are transferred to the computer, the WD1770 checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register so that a comparison can be made by the user. At the end of the operation an interrupt is generated and the Busy Status is reset.

### Read Track

Upon receipt of the READ track command, the head is loaded and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. All Gap, Header, and data bytes are assembled and transferred to the data register and DRQ's are generated for each byte. The accumulation of bytes is synchronized to each address mark encountered. An interrupt is generated at the completion of the command.

This command has several characteristics which make it suitable for diagnostic purposes. They are: no CRC checking is performed; gap information is included in the data stream; and the address mark detector is on for the duration of the command. Because the AM detector is always on, write splices or noise may cause the chip to look for an AM.

The ID AM, ID field, ID CRC bytes, DAM, Data, and Data CRC Bytes for each sector will be correct. The Gap Bytes may be read incorrectly during write-splice time because of synchronization.

### WRITE TRACK FORMATTING THE DISK

(Refer to section on Type III commands for flow diagrams.)

Formatting the disk is a relatively simple task when operating programmed I/O or when operating under DMA with a large amount of memory. Data and gap information must be provided at the computer interface. Formatting the disk is accomplished by positioning the R/W head over the desired track number and issuing the Write Track command.

Upon receipt of the Write Track command, the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving the command, but writing will not start until after the first byte has been loaded into the Data Register. If the DR has not been loaded within 3 byte times, the operation is terminated making the device Not Busy, the Lost Data Status Bit is set, and the interrupt is activated. If a byte is not present in the DR when needed, a byte of zeroes is substituted.

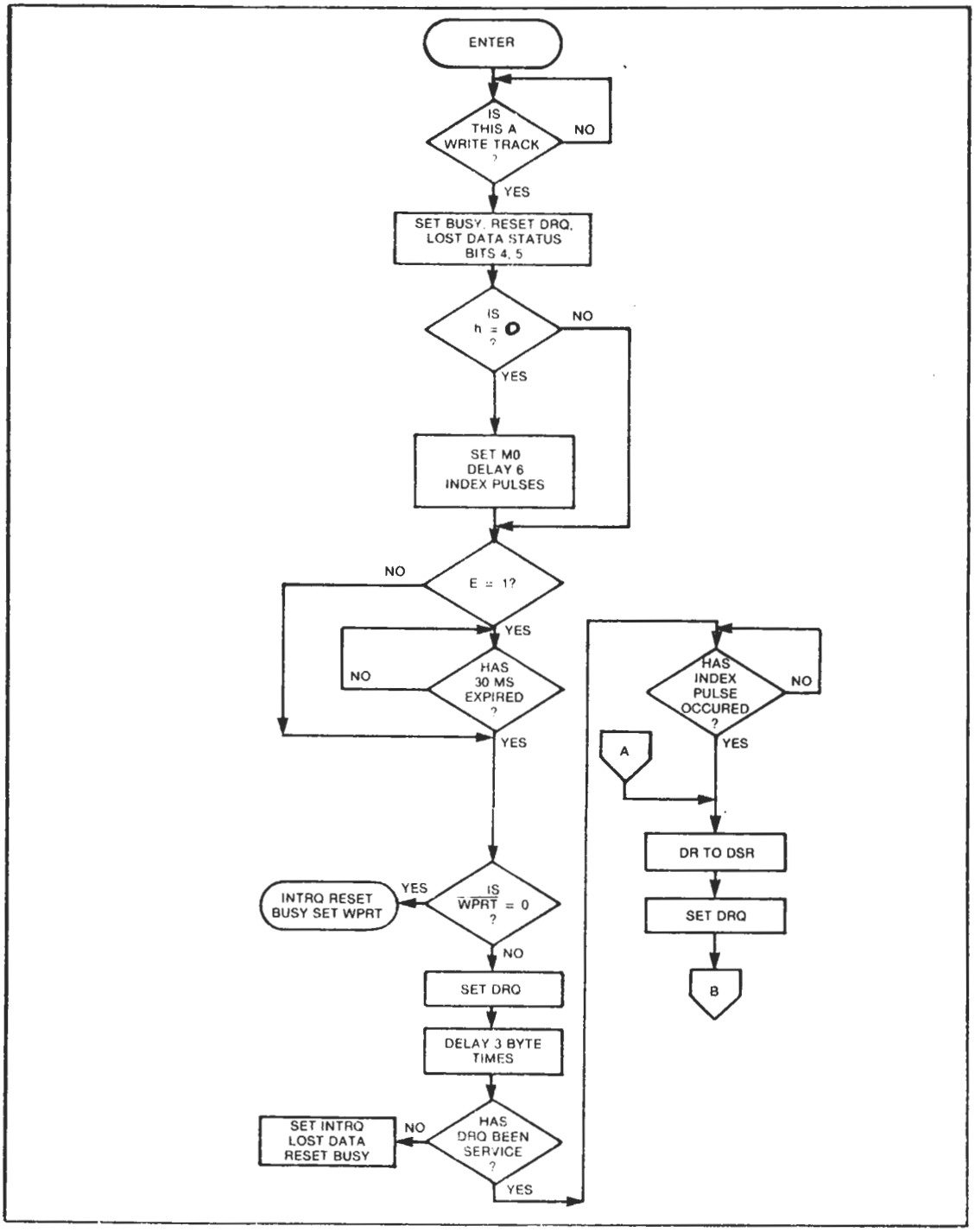
This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the WD1770 detects a data pattern of F5 through FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation.

| DATA PATTERN IN DR (HEX) | IN FM ($\overline{DDEN}$ = 1) | IN MFM ($\overline{DDEN}$ = 0) |
|---|---|---|
| 00 thru F4 | Write 00 thru F4 with CLK = FF | Write 00 thru F4, in MFM |
| F5 | Not Allowed | Write A1* in MFM, Present CRC |
| F6 | Not Allowed | Write C2** in MFM |
| F7 | Generate 2 CRC bytes | Generate 2 CRC bytes |
| F8 thru FB | Write F8 thru FB, CLK = C7, Preset CRC | Write F8 thru FB, in MFM |
| FC | Write FC with CLK = D7 | Write FC in MFM |
| FD | Write FD with CLK = FF | Write FD in MFM |
| FE | Write FE, CLK = C7, Preset CRC | Write FE in MFM |
| FF | Write FF with CLK = FF | Write FF in MFM |

*Missing clock transition between bits 4 and 5.
**Missing clock transition between bits 3 and 4.



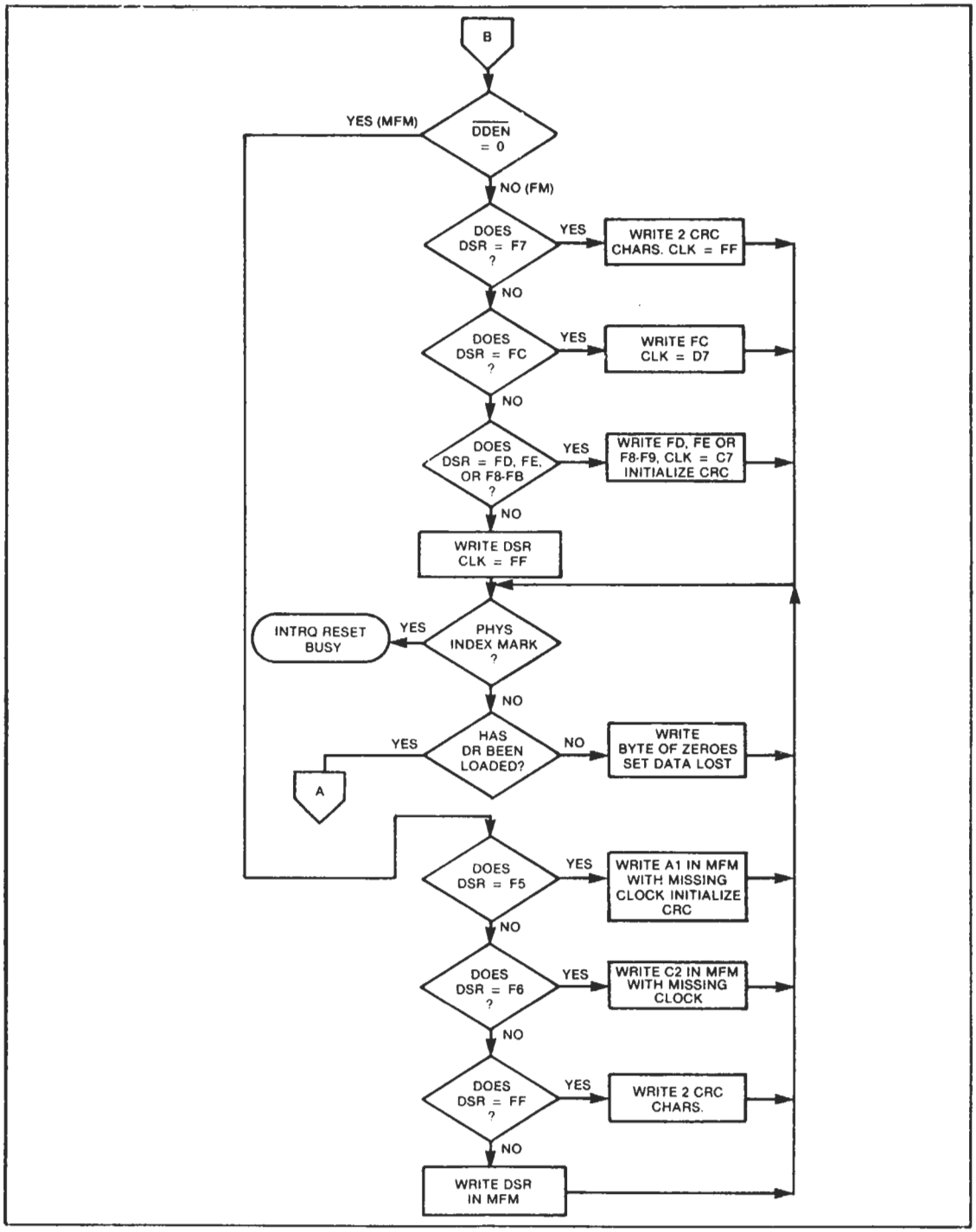
Reproduced with permission from
Western Digital Corporation




**TYPE III COMMAND WRITE TRACK**



Reproduced with permission from
Western Digital Corporation




**TYPE III COMMAND WRITE TRACK**



Reproduced with permission from
Western Digital Corporation

The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR in FM or by receipt of F5 in MFM. An F7 pattern will generate two CRC characters in FM or MFM. As a consequence, the patterns F5 through FE must not appear in the gaps, data fields, or ID fields. Also, CRC's must be generated by an F7 pattern.

Disks may be formatted in IBM 3740 or System 34 formats with sector lengths of 128, 256, 512, or 1024 bytes.

## TYPE IV COMMANDS

The Forced Interrupt command is generally used to terminate a multiple sector read or write command or to insure Type I status in the status register. This command can be loaded into the command register at any time. If there is a current command under execution (busy status bit set) the command will be terminated and the busy status bit reset.

The lower four bits of the command determine the conditional interrupt as follows:

$I_0$ = Don't Care
$I_1$ = Don't Care
$I_2$ = Every Index Pulse
$I_3$ = Immediate Interrupt

The conditional interrupt is enabled when the corresponding bit positions of the command ($I_3$-$I_0$) are set to a 1. Then, when the condition for interrupt is met, the INTRQ line will go high signifying that the condition specified has occurred. If $I_3$-$I_0$ are all set to zero (HEX D0), no interrupt will occur but any command presently under execution will be immediately terminated. When using the immediate interrupt condition ($I_3$ = 1) an interrupt will be immediately generated and the current command terminated. Reading the status or writing to the command register will not automatically clear the interrupt. The HEX D0 is the only command that will enable the immediate interrupt (HEX D8) to clear on a subsequent load command register or read status register operation. Follow a HEX D8 with D0 command.

Wait 16 micro sec (double density) or 32 micro sec (single density) before issuing a new command after issuing a forced interrupt. Loading a new command sooner than this will nullify the forced interrupt.

Forced interrupt stops any command at the end of an internal micro-instruction and generates INTRQ when the specified condition is met. Forced interrupt will wait until ALU operations in progress are complete (CRC calculations, compares, etc.).

### Status Register

Upon receipt of any command, except the Force Interrupt command, the Busy Status bit is set and the rest of the status bits are updated or cleared for the new command. If the Force Interrupt Command is received when there is a current command under execution, the Busy status bit is reset. and the rest of the status bits are unchanged. If the Force Interrupt command is received when there is not a current command under execution, the Busy Status bit is reset and the rest of the status bits are updated or cleared. In this case, Status reflects the Type I commands.

The user has the option of reading the status register through program control or using the DRQ line with DMA or interrupt methods. When the Data register is read the DRQ bit in the status register and the DRQ line are automatically reset. A write to the Data register also causes both DRQ's to reset.

The busy bit in the status may be monitored with a user program to determine when a command is complete, in lieu of using the INTRQ line. When using the INTRQ, a busy status check is not recommended because a read of the status register to determine the condition of busy will reset the INTRQ line.

The format of the Status Register is shown below:

| (BITS) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |

### RECOMMENDED — 128 BYTES/SECTOR

Shown below is the recommended single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one Data Request.

| NUMBER OF BYTES | HEX VALUE OF BYTE WRITTEN |
|---|---|
| 40 | FF (or 00) |
| 6 | 00 |
| 1 | FE (ID Address Mark) |
| 1 | Track Number |
| 1 | Side Number (00 or 01) |
| 1 | Sector Number (1 thru 1A) |
| 1 | 00 (Sector Length) |
| 1 | F7 (2 CRC's written) |
| 11 | FF (or 00) |
| 6 | 00 |
| 1 | FB (Data Address Mark) |
| 128 | Data (IBM uses E5) |
| 1 | F7 (2 CRC's written) |
| 10 | FF (or 00) |
| 369** | FF (or 00) |

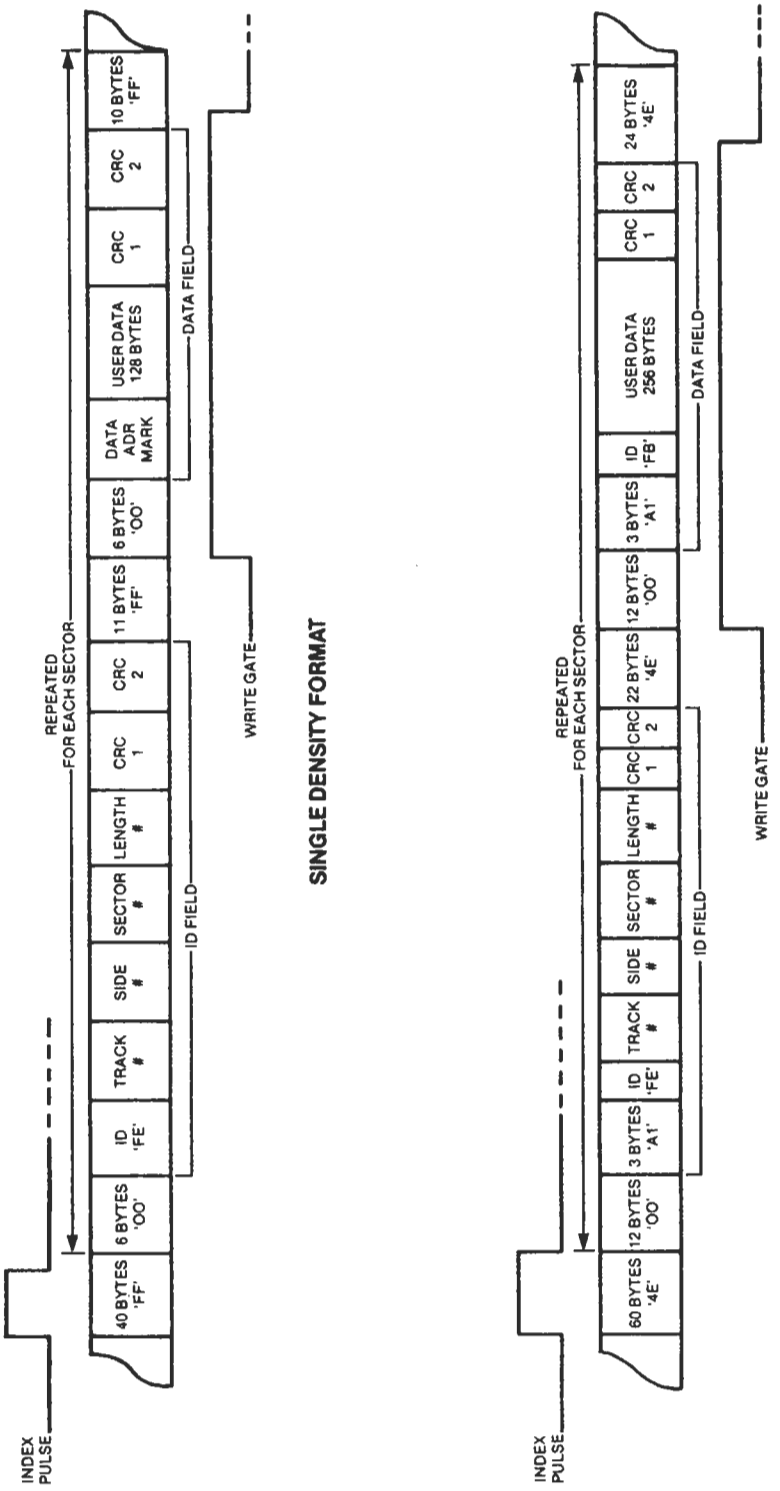
*Write bracketed field 16 times.
**Continue writing until WD1770 interrupts out. Approx. 369 bytes.

### 256 BYTES/SECTOR

Shown below is the recommended dual-density format with 256 bytes/sector. In order to format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.



Reproduced with permission from
Western Digital Corporation






Reproduced with permission from
Western Digital Corporation

| NUMBER OF BYTES | HEX VALUE OF BYTE WRITTEN |
|---|---|
| 60 | 4E |
| 12 | 00 |
| 3 | F5 (Writes A1) |
| 1 | FE (ID Address Mark) |
| 1 | Track Number (0 thru 4C) |
| 1 | Side Number (0 or 1) |
| 1 | Sector Number (1 thru 1A) |
| 1 | 01 (Sector Length) |
| 1 | F7 (2 CRC's written) |
| 22 | 4E |
| 12 | 00 |
| 3 | F5 (Writes A1) |
| 1 | FB (Data Address Mark) |
| 256 | DATA |
| 1 | F7 (2 CRC's written) |
| 24 | 4E |
| 668** | 4E |

*Write bracketed field 16 times.
**Continue writing until WD1770 interrupts out. Approx. 668 bytes.

**1. Non-Standard Formats**

Variations in the recommended formats are possible to a limited extent if the following requirements are met:

1) Sector size must be 128, 256, 512 of 1024 bytes.
2) Gap 2 cannot be varied from the recommended format.
3) 3 bytes of A1 must be used in MFM.

In addition, the Index Address Mark is not required for operation by the WD1770 Gap 1, 3, and 4 lengths can be as short as 2 bytes for WD1770 operation, however PLL lock up time, motor speed variation, write-splice area, etc. will add more bytes to each gap to achieve proper operation. It is recommended that the recommended format be used for highest system reliability.

| | FM | MFM |
|---|---|---|
| Gap I | 16 bytes FF | 32 bytes 4E |
| Gap II | 11 bytes FF | 22 bytes 4E |
| * | 6 bytes 00 | 12 bytes 00 |
| * | | 3 bytes A1 |
| Gap III** | 10 bytes FF * | 24 bytes 4E |
| | 4 bytes 00 | 8 bytes 00 |
| | | 3 bytes A1 |
| Gap IV | 16 bytes FF | 16 bytes 4E |

*Byte counts must be exact.
**Byte counts are minimum, except exactly 3 bytes of A1 must be written.

**STATUS REGISTER DESCRIPTION**

| BIT NAME | MEANING |
|---|---|
| S7 MOTOR ON | This bit reflects the status of the Motor On output. |
| S6 WRITE PROTECT | On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated. |
| S5 RECORD TYPE/SPIN-UP | When set, this bit indicates that the Motor Spin-Up sequence has completed (6 revolutions) on Type I commands. Type 2 & 3 commands, this bit indicates record Type. 0 = Data Mark. 1 = Deleted Data Mark. |
| S4 RECORD NOT FOUND (RNF) | When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated. |
| S3 CRC ERROR | If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated. |
| S2 LOST DATA/ TRACK 00 | When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when update. On Type I commands, this bit reflects the status of the TRACK 00 Pin. |
| S1 DATA REQUEST/ INDEX | This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated. On Type I commands, this bit indicates the status of the Index Pin. |
| S0 BUSY | When set, command is under execution. When reset, no command is under execution. |

**DC ELECTRICAL CHARACTERISTICS**

MAXIMUM RATINGS

Storage Temperature ..... −55°C to +125°C
Operating Temperature ..... 0°C to 70°C Ambient

Maximum Voltage to Any Input with Respect to $V_{SS}$ ..... (−15 to −0.3V)



Reproduced with permission from
Western Digital Corporation

## DC OPERATING CHARACTERISTICS

TA = 0°C to 70°C, $V_{SS}$ = 0V, $V_{CC}$ = +5V ±.25V

| SYMBOL | CHARACTERISTIC | MIN. | MAX. | UNITS | CONDITIONS |
|---|---|---|---|---|---|
| $I_{IL}$ | Input Leakage | | 10 | μA | $V_{IN} = V_{CC}$ |
| $I_{OL}$ | Output Leakage | | 10 | μA | $V_{OUT} = V_{CC}$ |
| $V_{IH}$ | Input High Voltage | 2.0 | | V | |
| $V_{IL}$ | Input Low Voltage | | 0.8 | V | |
| $V_{OH}$ | Output High Voltage | 2.4 | | V | $I_0 = -100\,\mu A$ |
| $V_{OL}$ | Output Low Voltage | | 0.40 | V | $I_0 = 1.6$ mA |
| $P_D$ | Power Dissipation | | .75 | W | |
| $R_{PU}$ | Internal Pull-Up | 100 | 1700 | μA | $V_{IN} = 0V$ |
| $I_{CC}$ | Supply Current | 75 (Typ) | 150 | mA | |

## AC TIMING CHARACTERISTICS

TA = 0°C to 70°C, $V_{SS}$ = 0V, $V_{CC}$ = +5V ±.25V

**READ ENABLE TIMING** — $\overline{RE}$ such that: R/$\overline{W}$ = 1, $\overline{CS}$ = 0.

| SYMBOL | CHARACTERISTIC | MIN. | TYP. | MAX. | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| TRE | RE Pulse Width of $\overline{CS}$ | 150 | | | nsec | $C_L$ = 50 pf |
| TDRR | DRQ Reset from $\overline{RE}$ | | 25 | 100 | nsec | |
| TIRR | INTRQ Reset from $\overline{RE}$ | | | 8000 | nsec | |
| TDV | Data Valid from $\overline{RE}$ | | 100 | 200 | nsec | $C_L$ = 50 pf |
| TDOH | Data Hold from $\overline{RE}$ | 50 | | 150 | nsec | $C_L$ = 50 pf |

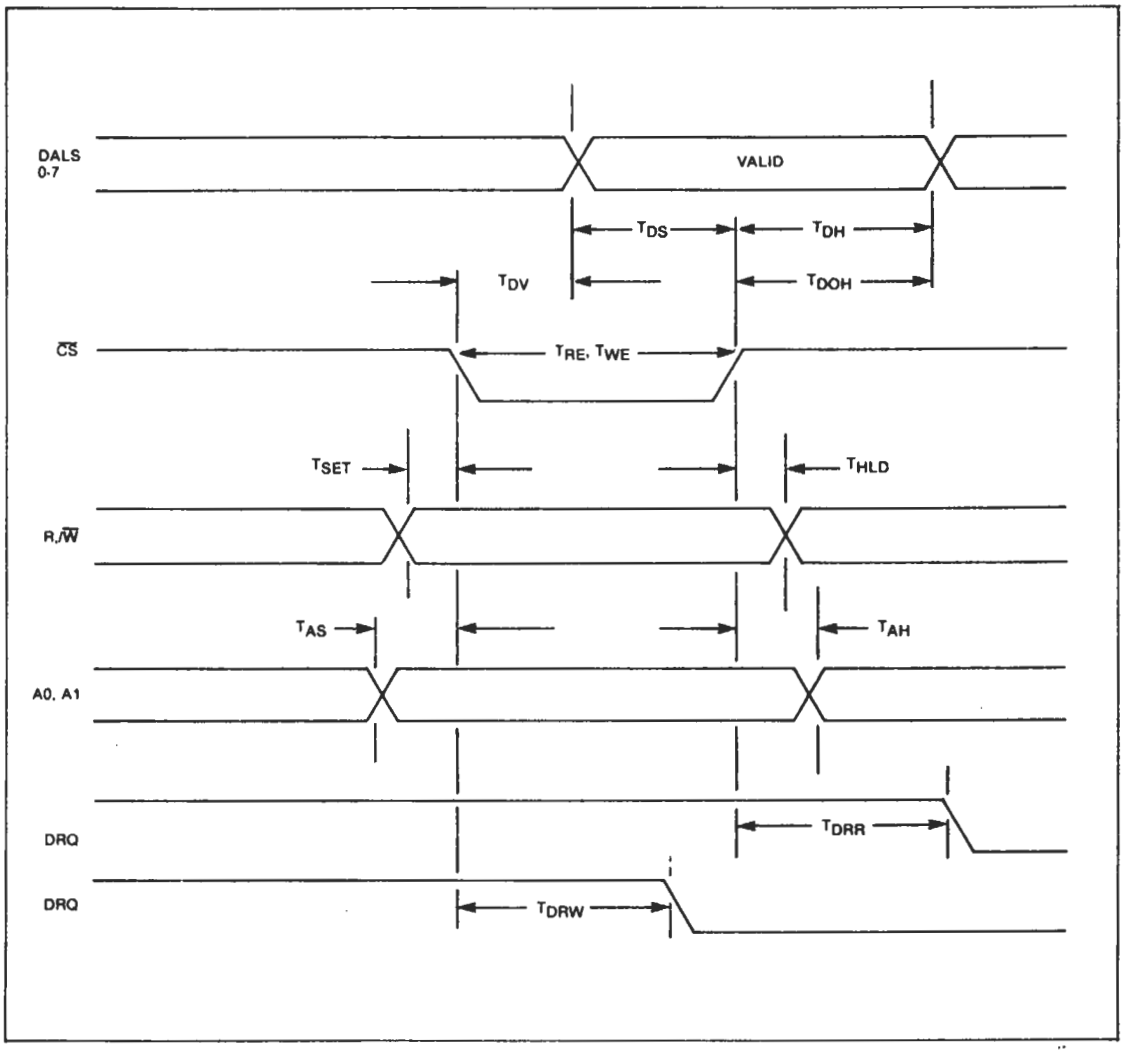
Note: DRQ and INTRQ reset are from rising edge (lagging) of $\overline{RE}$, whereas resets are from falling edge (leading) of $\overline{WE}$.

**WRITE ENABLE TIMING** — $\overline{WE}$ such that: R/$\overline{W}$ = 0, $\overline{CS}$ = 0.

| SYMBOL | CHARACTERISTIC | MIN. | TYP. | MAX. | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| TAS | Setup ADDR to $\overline{CS}$ | 50 | | | nsec | |
| TSET | Setup R/$\overline{W}$ to $\overline{CS}$ | 0 | | | nsec | |
| TAH | Hold ADDR from $\overline{CS}$ | 20 | | | nsec | |
| THLD | Hold R/$\overline{W}$ from $\overline{CS}$ | 0 | | | nsec | |
| TWE | $\overline{WE}$ Pulse Width | 150 | | | nsec | |
| TDRW | DRQ Reset from $\overline{WE}$ | | 100 | 200 | nsec | |
| TIRW | INTRQ Reset from $\overline{WE}$ | | | 8000 | nsec | |
| TDS | Data Setup to $\overline{WE}$ | 150 | | | nsec | |
| TDH | Data Hold from $\overline{WE}$ | 0 | | | nsec | |



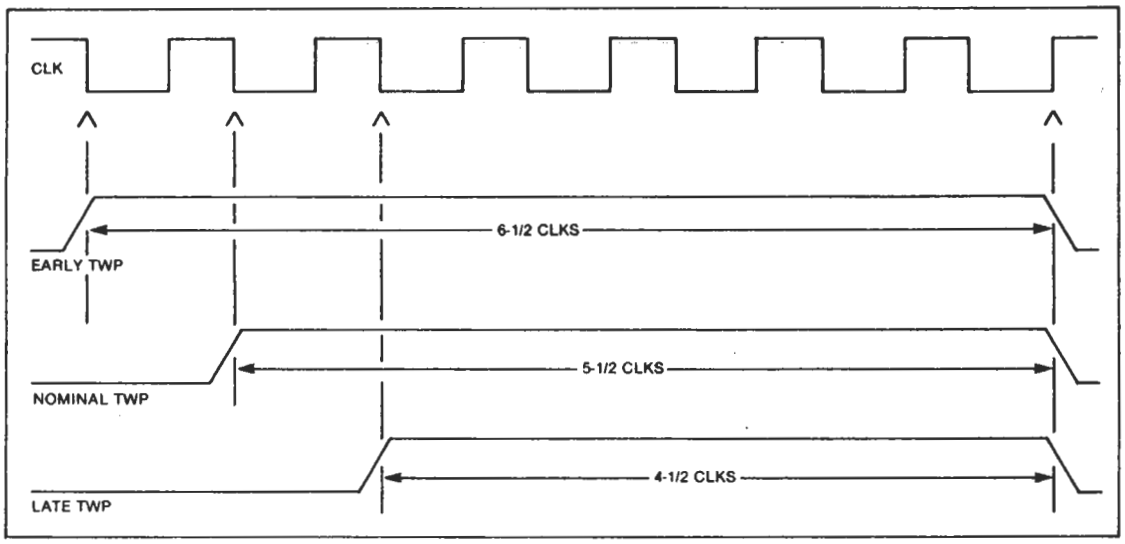
Reproduced with permission from
Western Digital Corporation




**REGISTER TIMINGS**



Reproduced with permission from
Western Digital Corporation




**WRITE DATA TIMING**

**WRITE DATA TIMING:**

| SYMBOL | CHARACTERISTIC | MIN. | TYP. | MAX. | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| Twg | Write Gate to Write Data | | 4<br>2 | | µsec<br>µsec | FM<br>MFM |
| Tbc | Write Data Cycle Time | | 4,6,8 | | µsec | |
| Twf | Write Gate off from WD | | 4<br>2 | | µsec<br>µsec | FM<br>MFM |
| Twp | Write Data Pulse Width | | 820<br>690<br>570<br>1380 | | nsec<br>nsec<br>nsec<br>nsec | Early MFM<br>Nominal MFM<br>Late MFM<br>FM |

**INPUT DATA TIMING:**

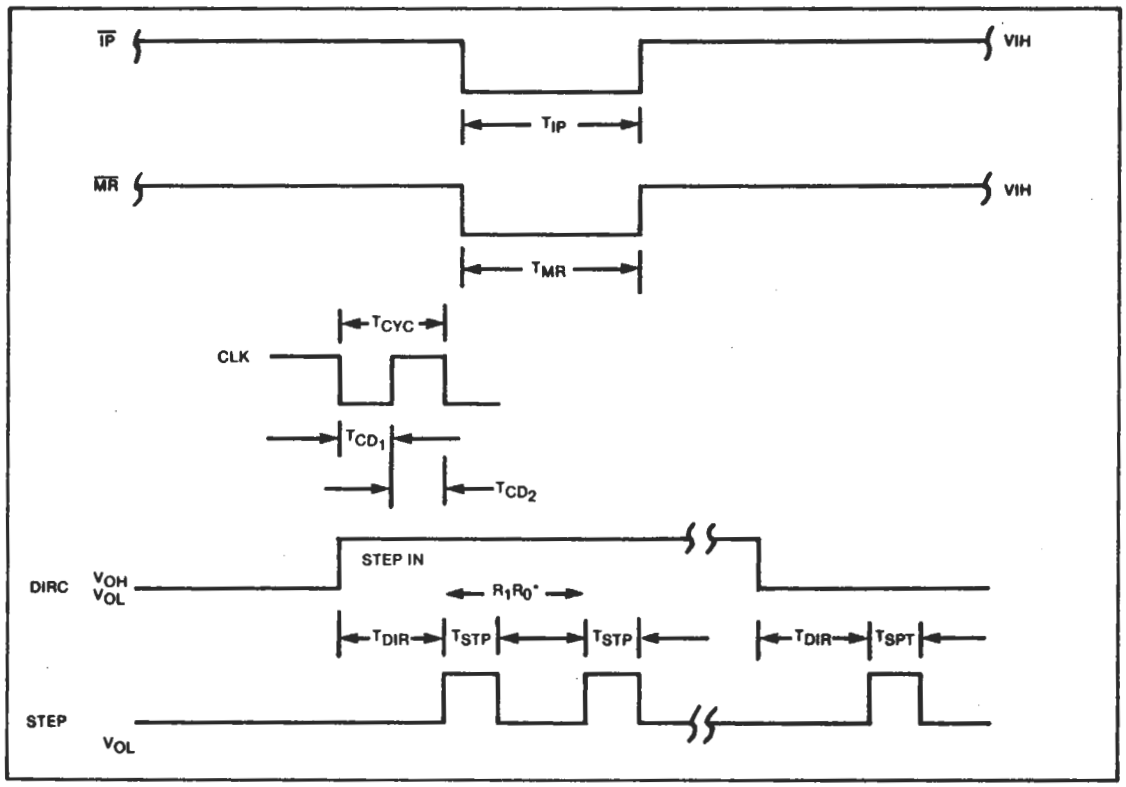
| SYMBOL | CHARACTERISTIC | MIN. | TYP. | MAX. | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| TPW | $\overline{\text{Raw Read}}$ Pulse Width | 200 | | | nsec | |
| TBC | $\overline{\text{Raw Read}}$ Cycle Time | 3000 | | | nsec | |

**MISCELLANEOUS TIMING:**

| SYMBOL | CHARACTERISTIC | MIN. | TYP. | MAX. | UNITS | CONDITIONS |
|---|---|---|---|---|---|---|
| $TCD_1$ | Clock Duty (low) | 50 | 67 | | nsec | |
| $TCD_2$ | Clock Duty (high) | 50 | 67 | | nsec | |
| TSTP | Step Pulse Output | | 4<br>8 | | µsec | MFM<br>FM |
| TDIR | Dir Setup to Step | | 24<br>48 | | µsec | MFM<br>FM |
| TMR | Master Reset Pulse Width | 50 | | | µsec | |
| TIP | Index Pulse Width | 20 | | | µsec | |



Reproduced with permission from
Western Digital Corporation




**MISCELLANEOUS TIMING**



Reproduced with permission from
Western Digital Corporation

# NCR 5380 SCSI INTERFACE
## PRODUCT BRIEF

### SCSI INTERFACE

- Asynchronous data transfer to 1.5 MBPS
- Supports both initiator and target roles
- Parity generation w/optional checking
- Supports arbitration
- Direct control of all bus signals
- High current outputs drive SCSI bus directly

### MPU INTERFACE

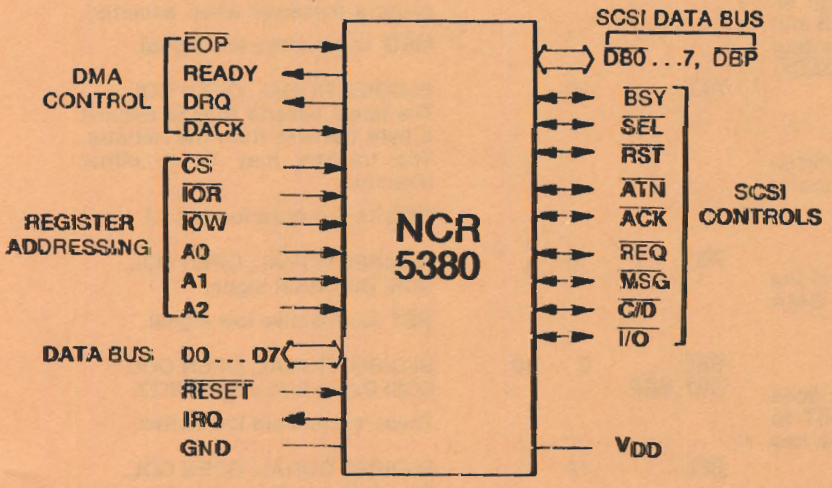
- Memory or I/O mapped interface
- DMA or programmed I/O
- Normal or block mode DMA
- Optional MPU interrupts

The NCR 5380 is designed to accommodate the Small Computer Systems Interface (SCSI) as defined by the ANSI X3T9.2 committee. The 5380 operates in both the Initiator and Target roles and can therefore be used in host adapter and control unit designs. This device supports arbitration, including reselection, and is intended to be used in systems that require either open collector or differential pair transceivers.* It has special high current outputs for driving the SCSI bus directly in the open collector mode.

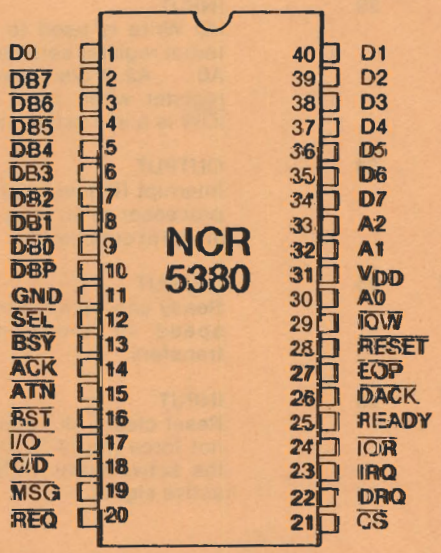
The NCR 5380 communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory mapped I/O. Minimal processor intervention is required for DMA transfers because the 5380 controls the necessary handshake signals. The NCR 5380 interrupts the MPU when it detects a bus condition that requires attention. Normal and block mode DMA is provided to match many popular DMA controllers.

* Differential pair operation is supported in the NCR 5381 (48 PIN).

### FUNCTIONAL PIN GROUPING




### PINOUT



NCR

# PIN DESCRIPTIONS

## MICROPROCESSOR INTERFACE SIGNALS

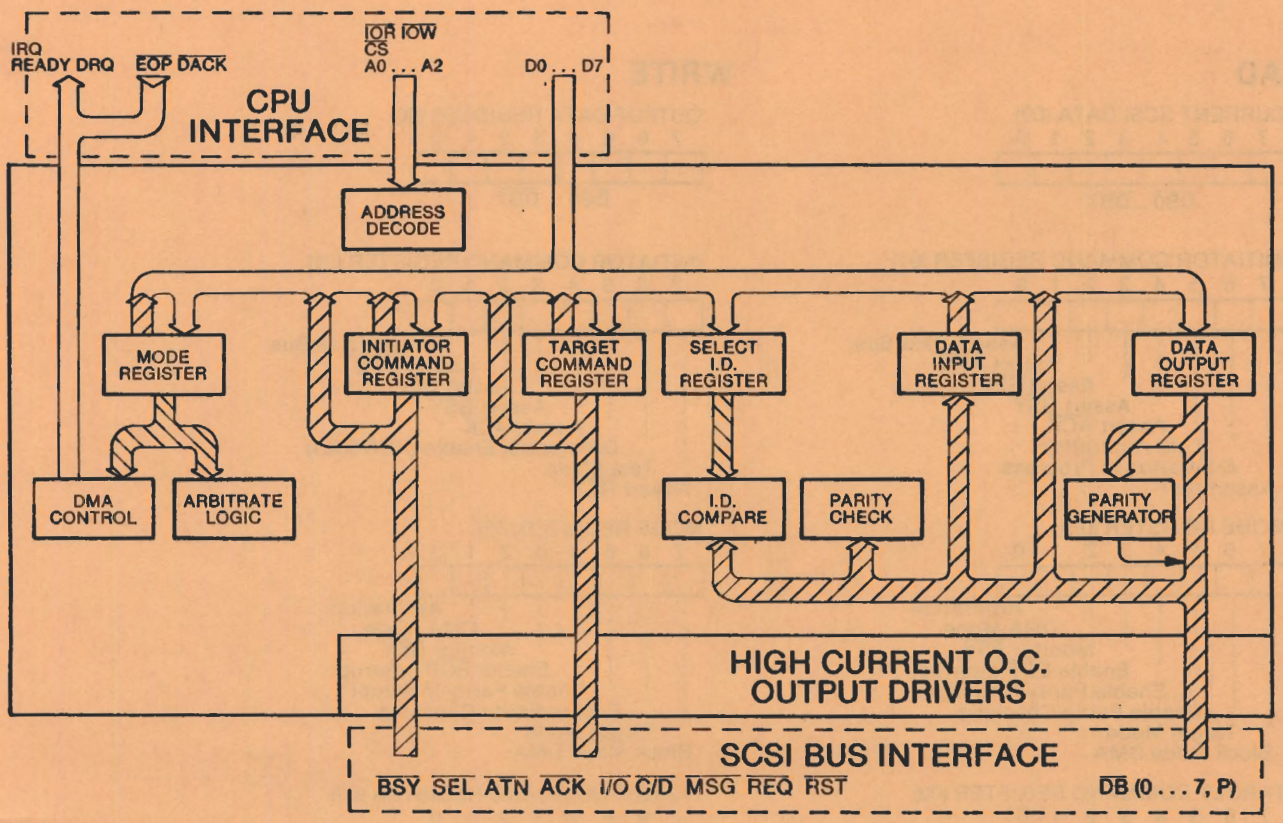
| Pin Name | Pin Number | Description |
|---|---|---|
| A0 . . . A2 | 30, 32, 33 | INPUTS<br>This address is used with $\overline{CS}$, $\overline{IOR}$ or $\overline{IOW}$ to address all internal registers. |
| $\overline{CS}$ | 21 | INPUT<br>Chip Select enables a read or write of the internal register selected by A0 . . . A2. $\overline{CS}$ is a low active signal. |
| $\overline{DACK}$ | 26 | INPUT<br>DMA Acknowledge resets DRQ and selects the data register for input or output. $\overline{DACK}$ is a low active signal. |
| DRQ | 22 | OUTPUT<br>DMA Request indicates that the data register is ready to be read or written. DRQ occurs only if DMA MODE is true in the command register. It is cleared by $\overline{DACK}$. |
| D0 . . . D7 | 34 . . . 40, 1 | BI-DIRECTIONAL, TRI-STATE<br>Microprocessor data bus Active high |
| $\overline{EOP}$ | 27 | INPUT<br>The End of Process signal is true during the last byte of a DMA transfer. This stops additional transfers but allows the current transfer to finish. $\overline{EOP}$ is a low active signal. |
| $\overline{IOR}$ | 24 | INPUT<br>I/O Read is used to read an internal register selected by $\overline{CS}$ and A0 . . . A2. It also selects the data register when used with $\overline{DACK}$. $\overline{IOR}$ is a low active signal. |
| $\overline{IOW}$ | 29 | INPUT<br>I/O Write is used to write an internal register selected by $\overline{CS}$ and A0 . . . A2. It also selects the data register when used with $\overline{DACK}$. $\overline{IOW}$ is a low active signal. |
| IRQ | 23 | OUTPUT<br>Interrupt Request alerts the microprocessor of an error condition or an event completion. |
| READY | 25 | OUTPUT<br>Ready can be used to control the speed of block mode DMA transfers. |
| $\overline{RESET}$ | 28 | INPUT<br>Reset clears all registers. It does not force the SCSI signal $\overline{RST}$ to the active state. $\overline{RESET}$ is a low active signal. |

## POWER SIGNALS

| Pin Name | Pin Number | Description |
|---|---|---|
| $V_{DD}$ | 31 | +5 VOLTS |
| GND | 11 | GROUND |

## SCSI INTERFACE SIGNALS

| Pin Name | Pin Number | Description |
|---|---|---|
| $\overline{ACK}$ | 14 | BI-DIRECTIONAL, OPEN COL.<br>INITIATOR ROLE: The chip asserts this signal in response to $\overline{REQ}$ for a byte transfer on the SCSI bus.<br>TARGET ROLE: $\overline{ACK}$ is received as a response to the $\overline{REQ}$ signal.<br>$\overline{ACK}$ is an active low signal. |
| $\overline{ATN}$ | 15 | BI-DIRECTIONAL, OPEN COL.<br>INITIATOR ROLE: The chip asserts this signal when the microprocessor requests the attention condition.<br>TARGET ROLE: $\overline{ATN}$ is a received signal.<br>$\overline{ATN}$ is an active low signal. |
| $\overline{BSY}$ | 13 | BI-DIRECTIONAL, OPEN COL.<br>The SCSI $\overline{BSY}$ signal can be driven and received concurrently.<br>$\overline{BSY}$ is an active low signal. |
| $\overline{C/D}$ | 18 | BI-DIRECTIONAL, OPEN COL.<br>Command/Data is an input for an initiator, an output for a target. It indicates a command when asserted.<br>$\overline{C/D}$ is an active low signal. |
| $\overline{I/O}$ | 17 | BI-DIRECTIONAL, OPEN COL.<br>Input/Output is an input for an initiator, an output for a target. It indicates an input to the initiator when asserted.<br>$\overline{I/O}$ is an active low signal. |
| $\overline{MSG}$ | 19 | BI-DIRECTIONAL, OPEN COL.<br>Message is an input for an initiator, an output for a target. It indicates a message when asserted.<br>$\overline{MSG}$ is an active low signal. |
| $\overline{REQ}$ | 20 | BI-DIRECTIONAL, OPEN COL.<br>The target asserts $\overline{REQ}$ to request a byte transfer from the initiator. The transfer may be in either direction.<br>$\overline{REQ}$ is an active low signal. |
| $\overline{RST}$ | 16 | BI-DIRECTIONAL, OPEN COL.<br>SCSI BUS reset signal<br>$\overline{RST}$ is an active low signal. |
| $\overline{SB0}$...<br>$\overline{SB7}$, $\overline{SBP}$ | 2 . . . 10 | BI-DIRECTIONAL, OPEN COL.<br>SCSI DATA BUS and PARITY<br>These signals are low active |
| $\overline{SEL}$ | 12 | BI-DIRECTIONAL, OPEN COL.<br>Select is used for selection and reselect operations.<br>$\overline{SEL}$ is an active low signal. |

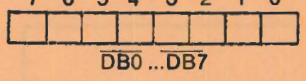



## REGISTER SUMMARY

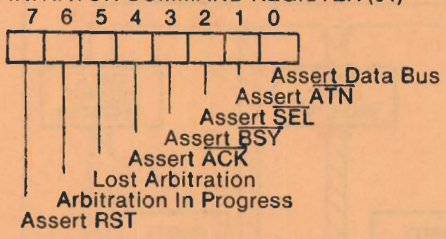
| A2 | A1 | A0 | R/W | REGISTER NAME |
|---|---|---|---|---|
| 0 | 0 | 0 | R | Current SCSI Data |
| 0 | 0 | 0 | W | Output Data Register |
| 0 | 0 | 1 | R/W | Initiator Command Reg. |
| 0 | 1 | 0 | R/W | Mode Register |
| 0 | 1 | 1 | R/W | Target Command Reg. |
| 1 | 0 | 0 | R | SCSI Bus Status |
| 1 | 0 | 0 | W | Select Enable Register |
| 1 | 0 | 1 | R | Bus & Status Register |
| 1 | 0 | 1 | W | Start DMA Send |
| 1 | 1 | 0 | R | Input Data Reg. |
| 1 | 1 | 0 | W | Start Target Rec. DMA |
| 1 | 1 | 1 | R | Reset Parity/Interrupts |
| 1 | 1 | 1 | W | Start Init. Rec. DMA |

## READ

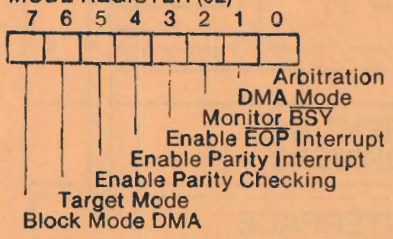
CURRENT SCSI DATA (00)



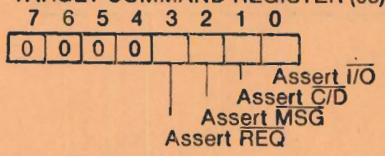

INITIATOR COMMAND REGISTER (01)



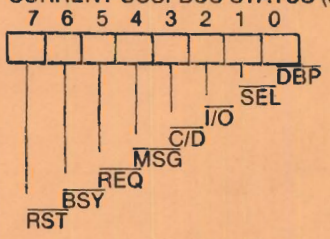

MODE REGISTER (02)



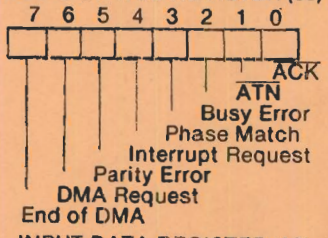

TARGET COMMAND REGISTER (03)



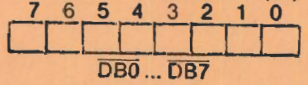

CURRENT SCSI BUS STATUS (04)



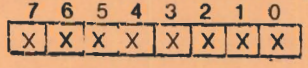

BUS & STATUS REGISTER (05)




INPUT DATA REGISTER (06)

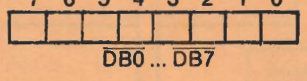



RESET PARITY/INTERRUPT (07)

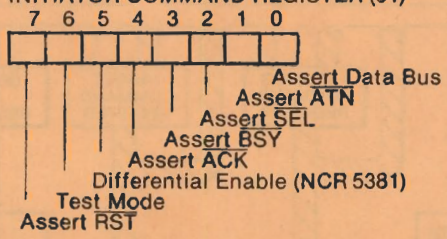



## WRITE

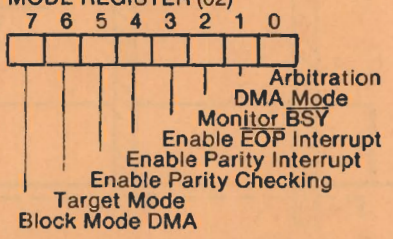
OUTPUT DATA REGISTER (00)



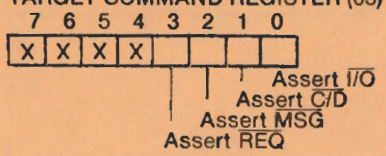

INITIATOR COMMAND REGISTER (01)



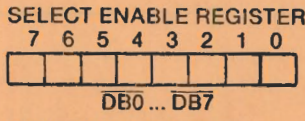

MODE REGISTER (02)



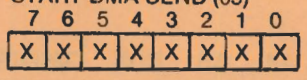

TARGET COMMAND REGISTER (03)



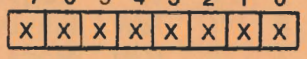

SELECT ENABLE REGISTER (04)



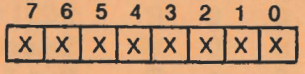

START DMA SEND (05)




START DMA TARGET RECEIVE (06)




START DMA INITIATOR RECEIVE (07)




NOTE: X = DON'T CARE

## ELECTRICAL CHARACTERISTICS
### OPERATING CONDITIONS

| PARAMETER | SYMBOL | MIN | MAX | UNITS |
|---|---|---|---|---|
| Supply Voltage | $V_{DD}$ | 4.75 | 5.25 | Volts |
| Supply Current | $I_{DD}$ | | 145 | mA. |
| Ambient Temperature | TA | 0 | 70 | °C |

## INPUT SIGNAL REQUIREMENTS

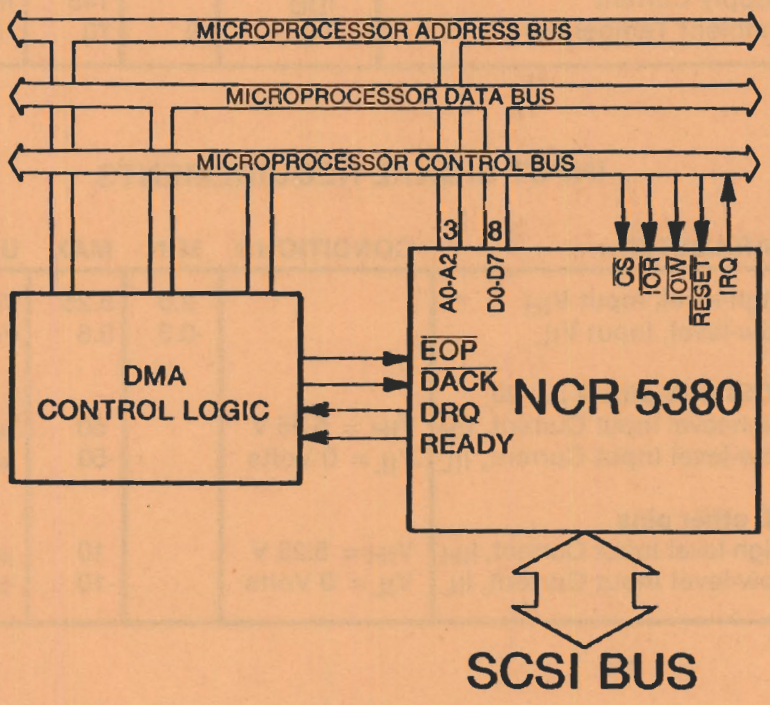
| PARAMETER | CONDITIONS | MIN | MAX | UNITS |
|---|---|---|---|---|
| High-level, Input $V_{IH}$ | | 2.0 | 5.25 | Volts |
| Low-level, Input $V_{IL}$ | | -0.3 | 0.8 | Volts |
| **SCSI BUS pins 2 . . . 20** | | | | |
| High-level Input Current, $I_{IH}$ | $V_{IH}$ = 5.25 V | | 50 | µa. |
| Low-level Input Current, $I_{IL}$ | $V_{IL}$ = 0 Volts | | -50 | µa. |
| **All other pins** | | | | |
| High-level Input Current, $I_{IH}$ | $V_{IH}$ = 5.25 V | | 10 | µa. |
| Low-level Input Current, $I_{IL}$ | $V_{IL}$ = 0 Volts | | -10 | µa. |

## OUTPUT SIGNAL REQUIREMENTS

| PARAMETER | CONDITIONS | MIN | MAX | UNITS |
|---|---|---|---|---|
| **SCSI BUS pins 2 . . . 20** | | | | |
| Low-level Output $V_{OL}$ | $V_{DD}$ = 4.75 V<br>$I_{OL}$ = 48.0mA. | | 0.5 | Volts |
| **All other pins** | | | | |
| High-level Output $V_{OH}$ | $V_{DD}$ = 4.75 V<br>$I_{OH}$ = -3.0mA. | 2.4 | | Volts |
| Low-level Output $V_{OL}$ | $V_{DD}$ = 4.75 V<br>$I_{OL}$ = 7.0 mA. | | 0.5 | Volts |

**PRELIMINARY**

Notice: This is not a final specification.
Some parametric limits are subject to change.




NCR

**NCR MICROELECTRONICS DIVISION**
**1635 Aeroplaza Drive**
**Colorado Springs, Colorado 80916**
**Phone: 800/525-2252**
**Telex: 45 2457 NCR MICRO CSP**

**While the information herein presented has been checked for both accuracy and reliability, NCR assumes no responsibility for either its use or for the infringement of any patents or other rights of third parties, which would result from its use. The publication and dissernination of the enclosed information confers no license, by implication or otherwise, under any patent or patent rights owned by NCR.**