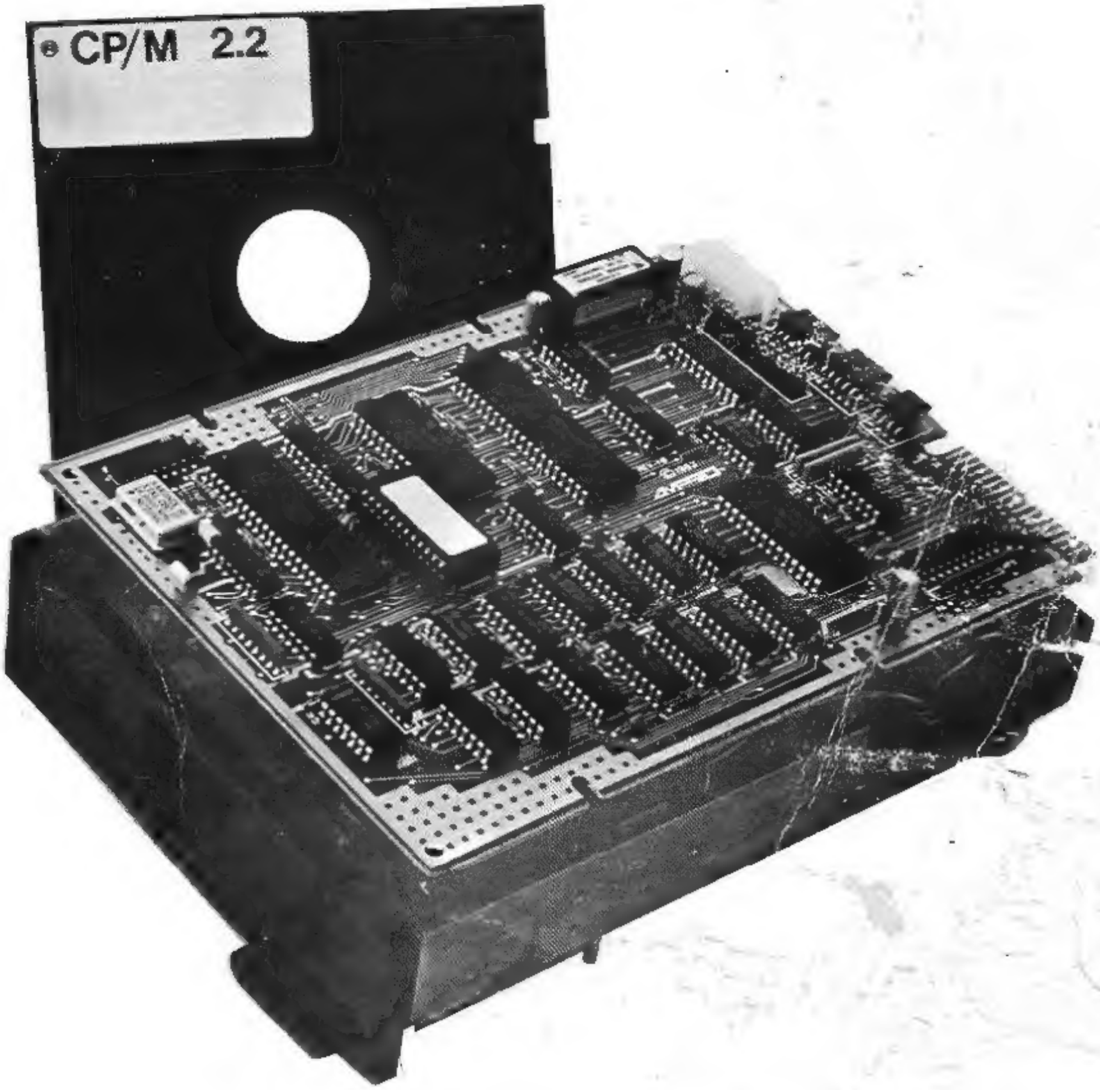
AMPRO

COMPUTERS, INCORPORATED

Little Board™

User's Manual



67 East Evelyn Ave. • Mountain View, CA 94041 • (415) 962-0230

LITTLE BOARD ™

USER´S MANUAL

AMPRO COMPUTERS, INC.
P.O. Box 390427
Mountain View, California 94039

(415) 962-0230

P/N: A74001-B

PREFACE

This manual is for users of the Little Board who wish to know how to incorporate it into a computer system. There are five sections, organized as follows:

- Section 1 - General Information - General information pertaining to the Little Board, its major features, and a functional description of each portion of the computer.
- Section 2 - How To Build a Computer - Descriptions of the external components necessary to construct a CP/M-based computer with two floppy disk drives. Included are tables listing the pinouts of each of the six board connectors, as well as special considerations, features, and specifications.
- Section 3 - How to Use the Computer - Brief descriptions of standard CP/M commands and utility programs. Included are descriptions of the utilities supplied with the Little Board.
- Section 4 - Programming Information - Descriptions of I/O addresses and other requirements for custom programming of the Little Board.
- Section 5 - Theory of Operation - Specific technical details of Little Board operation.

Specifications are subject to change without notice. The contents of this document are believed to be accurate. If errors are found, 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:

Z80A and all references to ´Z80´ are trademarks of Zilog, Inc.
CP/M is a trademark of Digital Research, Inc.
IBM is a trademark of International Business Machines
ZCPR3 is a program copyright by Richard L. Conn (used with permission)
DISK7 is a trademark of Echelon, Inc (used with permission)

For users of this manual who want more detailed information on the CP/M Operating System, the following are recommended for reference:

CP/M Primer, Stephen M. Murtha and Mitchell Waite, Howard W. Sams
CP/M Handbook, Rodnay Zaks
CP/M Revealed, Jack D. Dennon, Hayen Books
CP/M Operating System Manual, Digital Research, Pacific Grove, California; approx. $35.00 plus serial number
ZCPR3 information and source code: Echelon, Inc., 101 1st Street, Suite 427, Los Altos, CA 94022.

TABLE OF CONTENTS

SECTION 1 - General Description

SECTION 2 - How To Build A Computer

SECTION 3 - How To Use the Computer

TABLE OF CONTENTS
(Continued)

SECTION 4 - Programming Information

SECTION 5 - Theory of Operation

SECTION 1

GENERAL DESCRIPTION

1.1 INTRODUCTION

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

1.2 OVERVIEW

The AMPRO Little Board computer is a complete 8-bit, Z80-based 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
- 64 kilobytes of high-speed dynamic RAM
- two serial RS232C I/O ports
- one Centronics-compatible parallel printer port
- floppy disk interface capable of controlling from one to four single- or double-sided, single- or double-density 5 1/4-inch floppy disk drives.
- drives can be 48 tpi or 98 tpi, and can be mixed in the system.
- board size is compatible with most 5 1/4-inch floppy disk drives
- minimum external components
- power supply voltages compatible with 5 1/4-inch floppy disk drives

1.3 FUNCTIONAL DESCRIPTION

The following paragraphs briefly describe the Little Board computer. More detailed information can be found in Section 5, Theory of Operation.

1.3.1 CPU, Memory, and Timing

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

Two types of memory are used with the Little Board: EPROM and RAM. A single 2732 4k x 8 bit EPROM is used to initialize the system and load the CP/M operating system from floppy disk. After power-up the EPROM is enabled rather than RAM, and can be turned off or on by software.

System RAM consists of eight 64k x 1 bit dynamic RAM devices. Control circuitry for the RAM memory is entirely digital (noone-shotsorR-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. A third channel is optionally used by the floppy disk controller. The fourth CTC channel is available for use as a programmable timer in applications programs.

1.3.2 Serial I/O Ports

A Z80 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 serial port, 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.

Baud rate clocks are provided by the CTC for baud rates up to 9600 baud. Additionally, other circuitry provides for baud rates of 9600, 19,200, and 38,400 baud on Port A only. Since the two serial ports are otherwise identical, either can be programmed as a terminal, modem, or other RS232C device.

1.3.3 Parallel I/O Port

The parallel output supports 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 signals are defined by software.

1.3.4 Floppy Disk Controller

A Western Digital WD1770 floppy disk controller device provides all of the functions required to interface with standard 5 1/4-inch floppy disk drives, and many of the 3 to 4-inch "micro" floppy disk drives. The WD1770 includes the following capabilities within a single LSI device:

- digital phase locked loop
- digital write precompensation
- motor on start/stop delay
- selectable step rates: 6, 12, 20, and 30 mS

Timing for the floppy disk interface is derived directly from the 8MHz system clock, with no delay lines, R-C time constants, or one-shots. This again results in a very high degree of reliability.

1.4 LITTLE BOARD SPECIFICATIONS

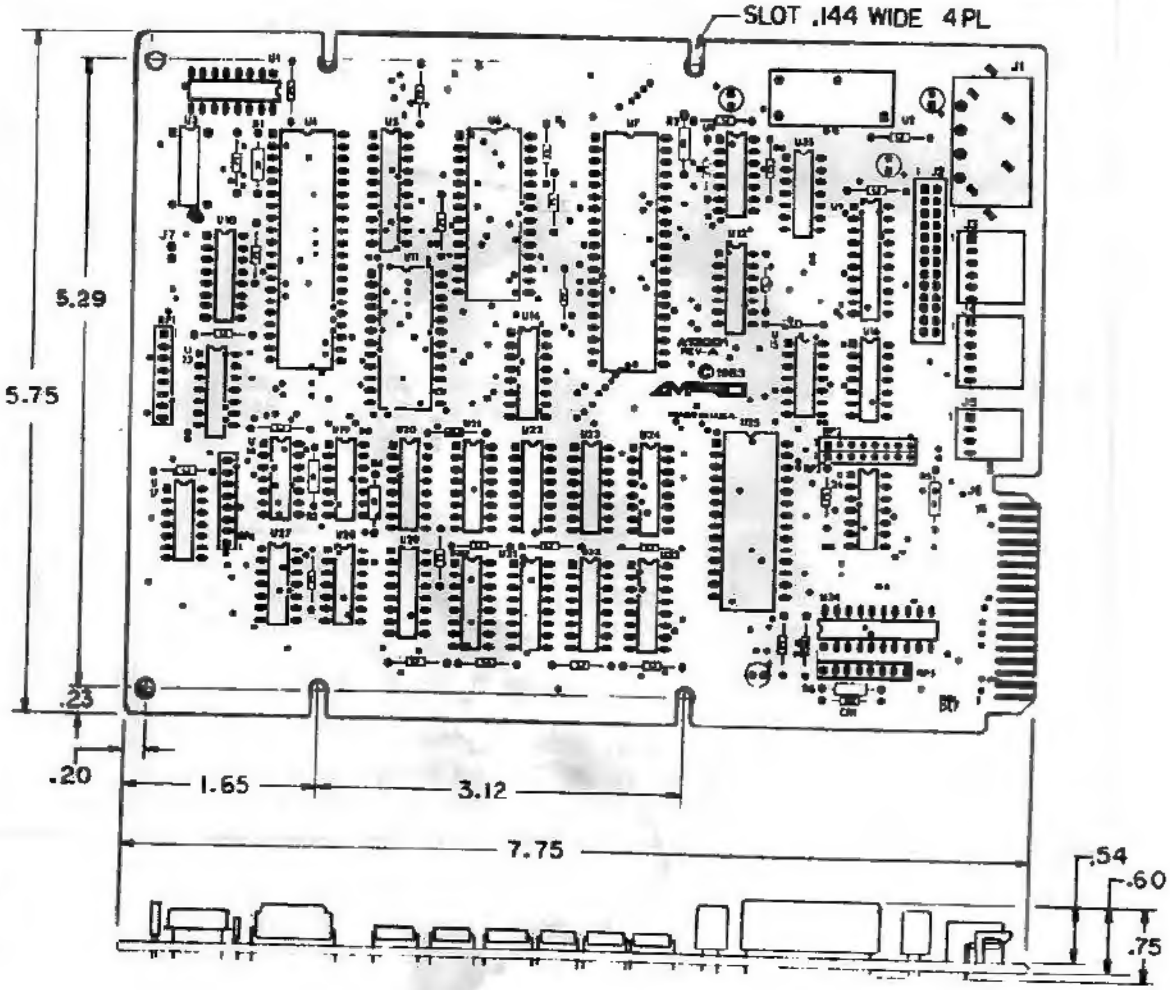
Table 1-1 lists the specifications for the Little Board computer.

Table 1-1. Little Board Specifications

<table>
<tr><td>CPU:</td><td>4MHz Z80A, 8-bit microprocessor</td></tr>
<tr><td>MEMORY:</td><td>64 kilobytes of dynamic RAM
4 kilobytes of EPROM (2732-type)</td></tr>
<tr><td>TIMER:</td><td>Z80A CTC</td></tr>
<tr><td>SERIAL I/O:</td><td>Z80A Dual Asynchronous Receiver/Transmitter (DART)
2 - RS232C compatible ports
Software-controlled baud rates:
Channel A - 75 to 9600 baud (low)
9,600, 19,200, 38,400 baud (high)
Channel B - 75 to 9600 baud
Four standard RS232C signals per port:
Transmit Data
Receive Data
Handshake Out
Handshake In
Two Ground pins</td></tr>
<tr><td>PARALLEL I/O:</td><td>Centronics-compatible printer port
10 signals supported:
Data Bits 1-8
Data Strobe
Printer Busy
12 Ground pins</td></tr>
<tr><td>DISK I/O:</td><td>Supports 1 to 4 single- or double-density,
single- or double-sided,
5 1/4-inch floppy disk drives
<table>
<tr><th>48 tpi drive</th><th>96 tpi drive</th></tr>
<tr><td>512 bytes/sector</td><td>1024 bytes/sector</td></tr>
<tr><td>10 sectors/track</td><td>5 sectors/track</td></tr>
<tr><td>40 tracks/side</td><td>80 tracks/side</td></tr>
</table>
Two tracks reserved for CP/M
Data rate: 250k bps
Digital phase locked loop
Software enabled digital write precompensation</td></tr>
</table>

Table 1-1. Little Board Specifications
(Continued).

POWER:	+5VDC +/-5% @ 0.75A +12VDC +/-5% @ 0.05A On-board -12VDC supply for RS232C ports Compatible with standard 5 1/4-inch floppy disk drives
ENVIRONMENT:	0 to 55 degrees C, operating, 5 to 95% humidity, non-condensing, 10,000 feet maximum altitude
SIZE:	7.75 x 5.75 x 0.75 inches Can be mounted on bottom of most 5 1/4-inch floppy disk drives
SOFTWARE:	CP/M version 2.2 Disk Operating System on 5 1/4-inch disk AMPRO Little Board System Utilities Boot program in a single 2732-type EPROM
DOCUMENTATION	AMPRO Little Board User's Manual AMPRO Little Board Technical Support Package (optional)



SECTION 2

HOW TO BUILD A COMPUTER

2.1 INTRODUCTION

This section describes what is required to build a two-drive, 64k byte RAM, CP/M 2.2 based computer, with the Little Board as the heart of the machine. This project requires some knowledge of electronics wiring and circuit techniques.

2.2 WHAT IS NEEDED

A very minimum number of external parts will turn a tiny circuit board into a very powerful personal computer. The components listed in Table 2-1 are easily available parts.

Table 2-1. External Components

Item	Description
2ea. Disk Drives	5 1/4-inch floppy disk drives, 48 tpi or 96 tpi, single- or double-sided.
Power Supply	+12VDC @ 2.0A, +5VDC @ 2.5A
Reset switch	s.p.s.t., normally open, w/LED indicator
Cables	2-RS232C, 1-Parallel, disk interface, and power cables for Little Board and drives.
Cabinet	Housing for computer system.
NOTE: External serial I/O cables are available from AMPRO.	

Figure 2-1 shows the Little Board external 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 size of approximately 7 x 8 x 11 inches is adequate. Tables 2-2 through 2-6 list the cable connector pinouts for the various external Little Board connectors. Table 2-7 lists the mating connectors and part numbers of suggested manufacturers.

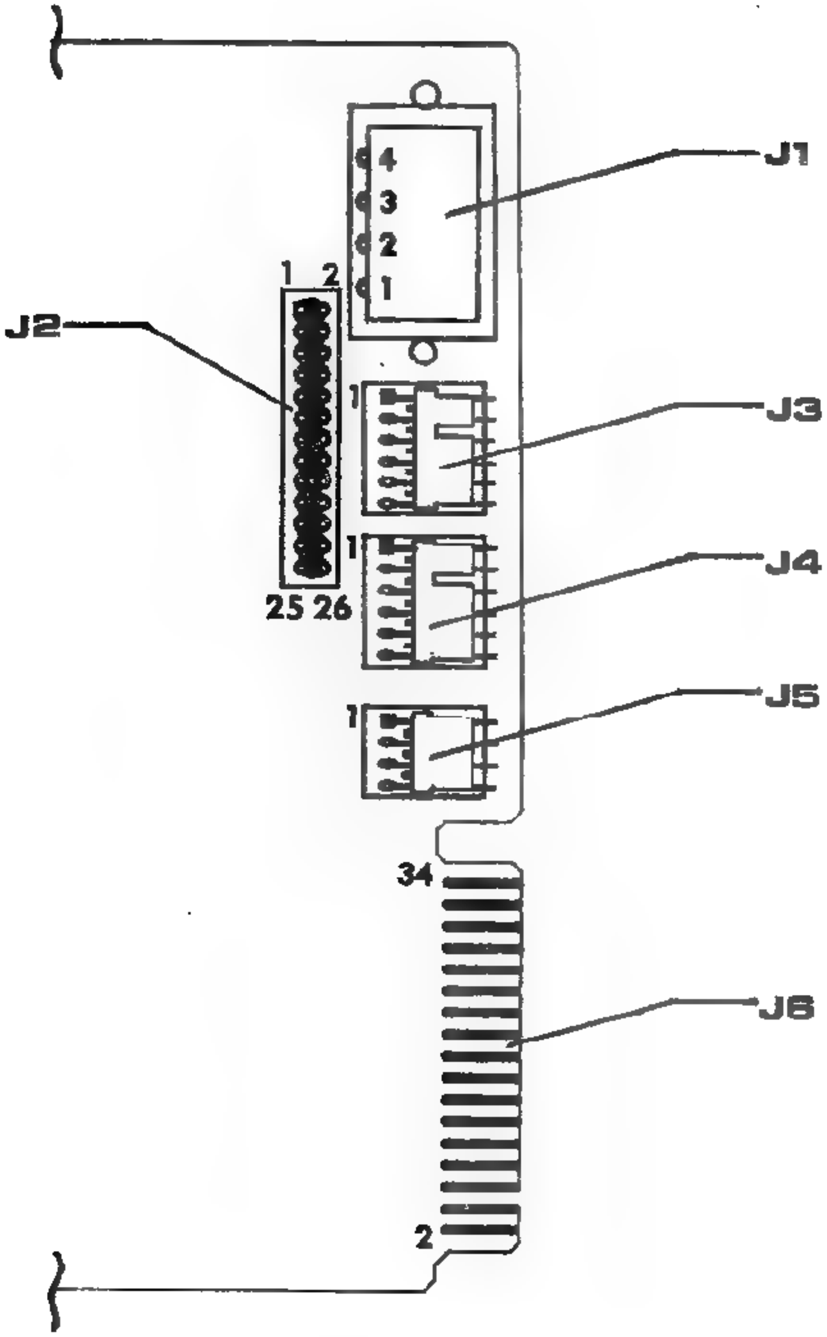


Figure 2-1. Little Board Connector Locations.

2.2.1 DC Power Input

The power connector pinout is identical with that of power connectors on nearly all 5 1/4-inch floppy disk drives. Board connector J1 contains the Little Board power connections. Note that pin 1 on J1 is reversed from the other connectors. Refer to Table 2-2.

CAUTION

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

Table 2-2. Power Connections, J1.

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

2.2.2 Parallel Printer Port

The Little Board's 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. Note that the pin numbering for the printer connector differs from that of the header. The header is numbered as shown in Figure 2-1. Note that some printers may include unique signals not shown. The board connector is J2. Refer to Table 2-3.

2.2.3 Serial RS232C I/O Ports A and B

Table 2-4 lists the cable connections for each of the two RS232C I/O ports. Serial port A is board connector J3, and Serial port B is board connector J4. The DB-25 connector pins are shown for Data Communication Equipment (DCE) wiring. For Data Terminal Equipment (DTE) wiring, reverse pins 2 & 3 and pins 5 & 20 at the DB-25 connector. Refer to Table 2-4.

2.2.4 Reset/Power On Connector

This connector is for connection to an external s.p.s.t switch to provide the master RESET signal. In addition, a 15 mA current source provides power to an LED power-on indicator. The board connector is J5. Refer to Table 2-5.

Table 2-3. Parallel Printer Cable Connections, J2.

Pin	Signal Name	Function	Printer Signal Pins
1	DS*	Data Strobe to printer (low)	1
3	Data 1	LSB of printer data	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	9
19	Not Used		
21	BUSY	Printer BUSY input to Little Board	11
23	Not Used		
25	Not Used		
2-26	All even	Signal grounds	19-27 & 29

Table 2-4. External Serial I/O Cable Connections, J3/J4.

Pin	Signal Name	Function	DB-25 Pin (DCE)
1	Ground	Protective Ground	1
2	Ground	Signal Ground	7
3	TxD	Data Output	3
4	HSO	Handshake Signal Out	5
5	RxD	Data Input	2
6	HSI	Handshake Signal In	20
Note: AMPRO Cable P/N is A60005-001			

Table 2-5. Reset/Power-On Connections, 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

2.2.5 Floppy Disk Interface

Table 2-6 lists the floppy disk drive interface cable connections. This cable is a flat ribbon cable with 34 conductors. A single PC edge-type connector is at the Little Board end, while there can be from 1 to 4 PC edge-type connectors at the disk drive end. The board connector is J6.

Table 2-6. Floppy Disk Interface Connections, J6.

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

2.2.6 External Connector Part Numbers

The part numbers listed in Table 2-7 are those of the manufacturer, not AMPRO. Cable assemblies for the serial ports are available from AMPRO at nominal cost.

2.3 DISK DRIVE CONSIDERATIONS

Nearly any type of soft-sectored, single- or double-sided, 48 tpi or 96 tpi, 5 1/4-inch floppy disk drive is usable with the Little Board. The higher quality drives you use, the better your system's reliability. Here are some things that are important to know:

- Any drive used must be compatible with the AMPRO Floppy Disk Interface.
- More than one type of floppy disk drive, up to four, can be present in the system, and in any mix.
- High quality, direct-drive 5 1/4-inch floppy disk drives are recommended.

- The first time the system is powered up, drive A must be a 48tpi (40-track) 5 1/4-inch drive, either single- or double-sided. Drives B, C and D can be any other system-compatible drive. System software is provided on a single-sided, 40-track format disk, which cannot be read by 96 tpi drives. Once the system is booted from the AMPRO-supplied CP/M disk, it is then possible to make system disks for any other AMPRO format.

- In order to use the AMPRODSK disk copy option, there must be at least two drives of the same type in the system. The PIP utility can be used between drives of different types.

- System software reads each disk to determine the format (512 or 1024 bytes per sector), and whether single- or double-sided. Single-sided disks have sectors numbered from 1 to 10. Double-sided disk sector numbers are offset by 16 (16 to 25).

- Each disk format has two system tracks reserved. Single-sided disks have the first two tracks reserved. Double-sided disks have the first track on each side reserved as system tracks.

- Each disk drive must be jumpered for a specific Drive Select value, 1 through 4. Consult your drive documentation.

- 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, select the Head Load with Motor On jumpering.

Table 2-7. External Connector Part Numbers

External Connector	Function	Part Number
P1	Power Connector	Housing: AMP 1-480424-0 Contacts: AMP 60619-1 (4 req.)
P2	Parallel I/O Board end	3M: 3399-6000 T&B: 609-2601M Molex: 15-29-8262
	Parallel I/O Printer end	AMP: 57F-30360 3M: 3366-1001 T&B: 609-36M
P3	Serial I/O, Port A	Housing: Molex 22-01-2067 Contacts: Molex 08-50-0114 (6 req.)
P4	Serial I/O, Port B	Same as P3

Table 2-7. External Connector Part Numbers
(Continued)

External Connector	Function	Part Number
P5	RESET/Power LED	Housing: Molex 22-01-2047 Contacts: Molex 08-50-0114 (4 req.)
P6	Floppy Disk Interface (Card edge connectors)	3M: 3463-0001 T&B: 609-3415M Molex: 15-29-0341

2.4 SYSTEM CONNECTIONS

With the system completed, only connection to a terminal and a source of power for the computer are required. With the serial ports wired as shown, both RS232C connectors will be DCE. For connection to a terminal (DTE), you must use a straight-through RS232C cable. To connect to a modem (DCE), you must use a cable in which connections to pins 2 and 3 and 20 and 5 are reversed at one end relative to the opposite end. Set the terminal as follows:

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

Set your terminal so that the Most Significant Bit (data bit #8) is transmitted as a 0 ("low" or "space"). Some terminals do not have a switch to do this, automatically sending a zero for data bit #8 when parity is off. The AMPRO BIOS does not mask the MSB when 8 bit transmission is selected.

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 Section 3, system boot parameters can be changed, but initially the system comes up with Serial Port A set for 9600 baud.

2.4.1 Booting The System

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. Use the following steps.

NOTE

> The first time booting of the system requires that drive A be either a single- or double-sided, 48 tpi, 5 1/4-inch floppy disk drive.

1. Set power switches to ON. Drive select indicator on drive A should light. If it does not, DO NOT insert a disk. Switch the system power OFF, and refer to section 2.5.

2. Insert the disk labeled "CP/M 2.2" into the A drive. The system should begin reading the operating system from disk, followed by a sign-on message:

```
AMPRO 61k CP/M VERS 2.2
    dd month yy

A0>
```

If the drive indicator lights, but nothing else, try inserting the flip-side of the disk and pressing RESET.

Your computer is now ready to use. Section 3 describes the software included with the Little Board. Have a good time!

2.5 TROUBLESHOOTING

It is possible that the computer did not work the first time, or fails sometime. You may have to troubleshoot it. The following are some suggestions. It is not recommended that you send the computer to AMPRO for service. Only Little Board service can be offered.

- Recheck all wiring, soldered connections.

- Check that power is available from the power supply.

- Be certain that the drives are working.

- If both drive indicators light during power-up, with drive handles closed (across slot), the drive or drives may be incorrectly connected to the drive cables. Switch the computer OFF and reverse the drive cable connector at the Little Board.

- Check that the drive select and configuration jumpers are properly set on each drive (see drive documentation).

- 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. Refer to the EPROM Monitor User's Manual.

SECTION 3

HOW TO USE THE COMPUTER

3.1 INTRODUCTION

This section discusses the power-up procedure, system prompts, the standard CP/M intrinsic commands and utility programs. In addition, the AMPRO utility programs supplied are explained in detail. For further information on the CP/M utilities, refer to the CP/M references listed in the Preface of this manual.

3.2 POWER-UP

Section 2 discussed connecting the completed system to console device and a modem as well as initial power-up. Assuming the computer works, there are two things it is recommended that you do the first time:

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

2. Run the system configuration (CONFIG) program.

3.2.1 Making Backup Disks

It is always a good idea to have at least one backup copy of your disks. Accidents do happen; Murphy is alive and well. Refer to the section on AMPRO utility programs for information on copying disks and files. Single-drive users should read the SWAPCOPY utility program information.

3.2.2 First Time Configuration

When using the system for the first time, some of the system initial default values are probably not ideal for your system. One important parameter to set is the floppy disk drive step rate. Initially, the system is set up for and boots with a step rate of 30 mS. However, this is much too slow for some drives. Check your drive´s step rate specification, and set the CONFIG step rate to one that is closest to the drive specification. The slowest drive in the system will determine the step rate to select from the CONFIG menu. This program is self-prompting. More specific information is available in the AMPRO utilities section.

NOTE

> Any modifications to the systems should be performed only on your backup disks. Do not use the disks shipped with your Little Board.

3.3 SINGLE DRIVE OPERATION

If you have a system with only one floppy disk drive, you can do nearly anything that can be done with two or more drives. Refer to the SWAPCOPY utility program in particular, later in this section.

3.4 LITTLE BOARD CP/M FEATURES

The operating system included with the Little Board is a modified version of standard CP/M version 2.2. One major difference is that the CCP is replaced by a program called ZCPR3. Some features of ZCPR3 are:

- DIR and USER commands are not available. They are replaced by the DIR.COM utility and enhanced CP/M features using ZCPR3. See Table 3-1.

- Semi-colons (;) can be used to separate multiple commands on a single command line. The sequence

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

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

- An automatic disk search path is implemented, permitting execution of an application program from a different disk drive, without the need to specify drive name. You can be logged onto drive B, and execute a program on drive A, without typing the drive prefix for the program drive. It is even possible to execute a program from a different user area. The 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

If you wish, you can "hide" .COM (program) files in user 15. Such files will not be visible from user 0, but will execute from user 0. Sneaky, but it works. DISK7 can be used to copy files to other user areas.

3.4.1 Intrinsic CP/M 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 3-1 lists the ZCPR3 commands versus the standard CCP commands. The following abbreviations are used in Table 3-1:

DU: - Drive number,User number	(e.g., A0:, B15:)
afn - Ambiguous file name	(e.g., *.COM, MYFILE.*)
ufn - Unambiguous file name	(e.g., MYFILE.TXT, DIR.COM)

Table 3-1. ZCPR3/CCP Command Comparison

Function	ZCPR3 Command	CCP Command
Display all files	DIR	DIR
Display all files in all user areas of current disk	DIR *.* $A	No equivalent
Display all files in all drives in current user area	DIR *.* $D	No equivalent
Display all files in all user areas and all drives	DIR *.* $AD	No equivalent
Display all .ASM files	DIR *.ASM	DIR *.ASM
Send directory output to LST: device (printer)	DIR *.* $P	No equivalent
Create a file containing directory contents (SD.DIR)	DIR *.* $F	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
Print file on printer	LIST DU:ufn	No equivalent
Save memory into file with overwrite warning	SAVE n DU:ufn	No equivalent
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
Load file anywhere into memory	GET adr DU:ufn	No equivalent
Call subroutine anywhere in memory	JUMP adr	No equivalent
Change disk	D:	D:
Change user	U:	USER n
Change disk and user at same time	DU:	No equivalent

3.4.2 CP/M Utility Programs

Included with the Little Board are nine standard CP/M utility programs.

- STAT.COM - Status of disk and other I/O devices
- PIP.COM - Permits single or multiple disk-to-disk file transfers
- DDT.COM - Dynamic Debugging Tool. Standard CP/M debugger
- ASM.COM - Standard assembler for 8080 instructions
- LOAD.COM - Converts a .HEX file output by the ASM program to an executable .COM file
- ED.COM - Standard CP/M line-type editor
- SUBMIT.COM - Permits execution of multiple commands and parameters stored in a disk file
- XSUB.COM - Same as SUBMIT.COM but extended to include line input to programs
- DUMP.COM - Permits display of a file in hexadecimal values

3.4.3 IOBYTE Implementation

The IOBYTE is used by the STAT utility to assign logical I/O devices to physical I/O devices. Table 3-2 lists the standard CP/M logical/physical device assignments and choices.

Table 3-2. 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:

The three I/O ports (physical devices) are assigned in the following way:

- Serial Port A is the CRT: device
- Serial Port B is the TTY: device
- Parallel Printer Port is the LPT: device

This IOBYTE implementation permits the use of either of the two serial ports as the console port. For example, using Serial Port B as the console port (CON:=TTY:), Serial Port A´s high baud rates can be used with the LST: device port (LST:=CRT:).

The CONFIG utility program can be used to set the cold-boot defaults. For example, Serial Port B can be assigned, through CONFIG, as the main console at cold-boot. In the same fashion, either Serial Port A or B can be assigned as the list device rather than the parallel output port.

Optionally, you can use the STAT utility program. For your convenience, here are some examples (each followed by RETURN):

```
STAT CON:=TTY:     Assigns the console to Serial Port B
STAT LST:=TTY:     Assigns the list device to Serial Port B
STAT CON:=CRT:     Assigns the console to Serial Port A (default)
```

3.5 AMPRO SUPPLIED UTILITIES

There are eight additional utility programs supplied with the Little Board. The following paragraphs discuss each one.

- AMPRODSK.COM - used to copy, format, and verify disks.
- SYSGEN.COM - used to write the AMPRO CP/M operating system onto a disk.
- MULTIDSK.COM - provides compatibility with other computers´ disk formats.
- CONFIG.COM - provides for modification or setting of your system´s BIOS according to your particular requirements.
- DIR.COM - permits displaying contents of disk directories. Public domain program, used with permission.
- UNERA.COM - permits recovery of files deleted (ERAsed) from disk. For use with ZCPR3. Public domain program, used with permission.
- STARTUP.COM - permits execution of multiple commands each time STARTUP is executed. For use with ZCPR3. Public domain program, used with permission.
- MULTIFMT.COM - permits formating disks using non-AMPRO formats.
- SWAPCOPY.COM - permits copying disks using only one disk drive.
- MOVCPM.COM - Configures the operating system for a user-definable memory size, using the standard CCP.
- ZMOVCPM.COM - Configures the operating system for a user-definable memory size, using the ZCPR3 enhancement.

Each utility is self-prompting. Each parameter or choice of commands is displayed on the screen as needed. The following descriptions include the screen prompts and options. User input is underlined.

3.5.1 AMPRODSK Utility

This utility permits disk-to-disk copying, disk formatting, and disk verification. To run this program simply type

```
AMPRODSK<RETURN>
```

The program will display a sign-on message followed by

```
COPY, FORMAT, VERIFY: (C, F OR V)
Press <ESC> or ^C to exit.
```

At this point, enter one of the three letters requested. If disk copy is desired, enter 'C' with no carriage return. The following dialog is an example of the disk copy program:

```
COPY creates a duplicate of a disk.

Source drive? (A, B, C or D) A

Place source disk on drive A

Press <RETURN> to continue, <ESC> to quit: <RETURN>

Destination drive? (A, B, C or D) B

Place destination disk on drive B

Press <RETURN> to continue, <ESC> to quit: <RETURN>
```

At this point, the system will alternately display "Reading Track xx Side y" with "Writing Track xx Side y".

If the copy is successful, the above line will be replaced with "COPY complete" and continue with "Verify Track xx Side y".

If the verify operation is successful, this will be replaced with "VERIFY complete", and return to the initial AMPRODSK command line.

To Format a disk, respond to the command line with the letter F. The following dialog illustrates the disk formatting process:

```
FORMAT prepares a fresh disk for data or program storage.

Destination drive? (A, B, C or D) B
```

```
Formats Available:
     1.  Single side 48tpi
     2.  Double side 48tpi
     3.  Single side 96tpi
     4.  Double side 96tpi

Choose one (1, 2, 3, 4) 2

Place destination disk on drive B

Press <RETURN> to write, any other key to abort. <RETURN>

Format  Track xx  Side y
```

If the format operation is successful, this line will be replaced by "FORMAT complete".

Next, disk verification will occur, displaying

```
Verify  Track xx  Side y
```

If verification is successful, this line will be replaced with "VERIFY complete", and then the AMPRODSK command line.

To simply verify a disk, enter the letter V or v. The following dialog illustrates the verification process:

```
VERIFY checks the reliability of data on a disk.

Destination drive?  (A, B, C of D) A

Verify  Track xx  Side y
```

If the verification is successful, this line will be replaced with "VERIFY complete", and then the AMPRODSK command line. The Verify operation only reads the disk information; no writing occurs. In reading a disk, the computer can check the overall integrity of data contained in each disk sector. This is done by comparing the error-detection code (CRC), calculated during the read, with the code read from the disk sector. If the two do not match, that sector is bad.

3.5.2 SYSGEN Utility

This program permits writing the CP/M operating system to the disk system tracks. The source for this operation is either a memory image of the system, or the system tracks on a disk. Execute the SYSGEN.COM program by typing

```
SYSGEN<RETURN>
```

the console should display a short sign-on message, then

```
Source Drive? (A, B, C or D)
```

When a drive letter is entered, the system will respond with

```
Place Source on n, then type <RETURN>
```

Place the Source disk in the selected drive and press RETURN. Next, the Destination is requested

```
Destination Drive? (A, B, C or D)
```

Enter a drive letter, but NOT the same as the Source, then RETURN. The Destination prompt

```
Place destination disk on y, then type <RETURN>
```

will be displayed. If RETURN only is entered, SYSGEN will abort, and exit to CP/M. Otherwise, the operating system will be written on the system tracks of drive y.

After each operation, SYSGEN will request a new destination disk. As many disks as desired can be SYSGENed. When finished, simply type RETURN instead of a drive letter. Also, see Section 3.6.

3.5.3 MULTIDSK Utility

The MULTIDSK utility program provides compatibility with 5 1/4-inch disk formats written by machines other than AMPRO. MULTIDSK permits the assignment of any system drive as drive "E" (Emulation). This utility is self-prompting. The following example illustrates assigning drive B access as drive "E". To run MULTIDSK.COM, simply type

```
MULTIDSK<RETURN>
```

The program will display some sign-on messages then request

```
Which drive do you wish to use as the "E" drive? (A,B,C or D) B
```

When the drive designator is entered, the main menu will be displayed next:

```
                  MULTIDSK  MAIN  MENU
                  --------  ----  ----

               1 - Single Sided 48 TPI Menu
               2 - Double Sided 48 TPI Menu
               3 - 96 TPI Menu

                  <ESC> - Exit to CP/M

Drive B Selected as the "E" Drive.
Select 1, 2, 3 or <ESC> to Exit: 1
```

To select a single-sided, 48 tpi menu, enter 1. The following final menu will be displayed. To select the IBM (CP/M 86) format, simply enter H.

These formats require a single or double sided 48 tpi drive.

```
                SINGLE SIDED 48 TPI MENU
                ------ ----- -- --- ----

   A - ACTRIX (ACCESS)                 K - MORROW MD2
   B - DEC VT180                       L - NEC PC8001A
   C - HEATH/ZENITH 100                M - OSBORNE 1
   D - HEATH/ZENITH 89 SD              N - OSBORNE 2
   E - HEATH/ZENITH 89 DD              O - TI Pro (CP/M 86)
   F - HEATH/ZENITH 89 XD              P - TRS80-1 w/OMIKRON
   G - HEATH w/MAGNOLIA                Q - TRS80-3 w/MEM MERCHANT CP/M
   H - IBM (CP/M 86)                   R - TRS80-4 w/MONTEZUMA CP/M
   I - KAYPRO II                       S - XEROX 820-I
   J - LOBO MAX80                      T - XEROX 820-II


                    <ESC> - Return to Main Menu


   Select a format or <ESC> to Main Menu: H

   Drive B is now an IBM SSDD (CP/M 86)
   drive when you call it "E".
```

With drive "E" assigned as an IBM (CP/M 86) format drive, each time "E" is accessed, only disks with the selected format can be read or written. Each time the drive is accessed as "B", it is AMPRO-compatible only.

NOTE

96 tpi drives cannot be used with 48 tpi formats, and vice versa.

3.5.4 MULTIFMT Utility

MULTIFMT provides the option to format or verify disks having formats other than the AMPRO format. To read or write disks formatted in this way, use the MULTIDSK utility program described above. To run MULTIFMT, type

```
MULTIFMT<RETURN>
```

The system will display a sign-on message then request:

```
FORMAT or VERIFY?  (F or V)
Press <ESC> or <RETURN> to exit.
```

To format a disk, enter "F". The system will respond with:

```
FORMAT prepares a fresh diskette for data or program storage.

Destination drive?  (A, B, C or D)
```

Select the desired drive by entering the appropriate drive designator. The system will respond with a list of available formats similar to the following:

```
Formats Available:

                48 TPI FORMATS
            (48 tpi Drive Required)

A - H/Z 89 SSDD                G - KAYPRO II SSDD
B - H/Z 89 DSDD                H - MORROW MD2 SSDD
C - H/Z 89 SSXD                I - MORROW MD3 DSDD
D - H/Z 89 DSXD                J - OSBORNE 2 SSDD
E - H/Z 100 SSDD               K - TRS80-3 w/MEM MERCH CP/M SSDD
F - H/Z 100 DSDD               L - TRS80-4 w/MONTEZUMA CP/M SSDD

                96 TPI FORMATS
            (96 tpi Drive Required)

M - DEC RAINBOW SSDD           P - H/Z 89 DSDD
N - EAGLE IIE-2 SSDD           Q - H/Z 89 SSXD
O - H/Z 89 SSDD                R - H/Z 89 DSXD

Choose one or <ESC> to Restart:

Place destination disk in Drive X  (where "X" is the selected drive)

Press <RETURN> to FORMAT, any other key to abort.
```

When RETURN is pressed, the disk will be formatted then verified. If for some reason verification fails, an appropriate error message will be displayed, then return to the initial command choice message.

NOTE

96 tpi drives cannot be used with 48 tpi formats, and vice versa.

3.5.5 CONFIG Utility

CONFIG.COM permits you to set most of your system´s parameters, either temporarily or permanently. The program includes a menu of system functions. Each function is selected by choosing a number from the menu. CONFIG will then display parameters whose values can be changed. The following is a sample display of the CONFIG displays. First, sign-on messages, then:

```
     View parameters from Memory or from Disk? (M or D)
```

Enter either M or D. If D is entered, the following prompt will appear:

```
     Which disk shall you read from? (A, B, C or D)
```

After entering either M or D, the system will display a Configuration Table as follows:

```
Configuration Table:

Parameter:               Currently:

1. Terminal              Serial Port A
2. Printer               Parallel Port
3. Max. Drives           2
4. Step Rate             30 milliseconds
5. Autocommand           startup

6. Serial Port A         data bits       8
   configuration         stop bits       1
                         parity          even
                         baud rate       9600
                         hand shake      no

7. Serial Port B         data bits       8
   configuration         stop bits       1
                         parity          even
                         baud rate       300
                         hand shake      no

Any (more) changes? (1 through 7 or No)
```

To select a parameter to change, enter a number from 1 through 7. CONFIG is self-prompting. Simply enter the requested response. If a particular value is not to be changed, enter <RETURN> only. When each item is finished, a new Configuration Table will be displayed, showing the current values and functions. If no further changes are to be made, simply enter N, n, or <RETURN>.

You will now have the opportunity to store the changes in memory, to disk, or both:

```
Install changes in Memory or on Disk?  (M or D)
Any other key exits the program.
```

For example, you might want to enter M to test new system parameters, without affecting the disk system tracks. This prompt will repeat until any key other than M or D is pressed. If D is selected, enter the drive designator when requested. The next time the system is booted, the new system parameters will be used.

3.5.6 DISK7 Utility

DISK7 provides disk file manipulation, disk-to-disk file copy, viewing text file contents, and several other handy disk operations. The following is an example of the DISK7 sign-on display.

```
                   DISK 7.7 -- File Manipulation Program -- 02/11/84
  C - Copy file        D - Delete file      F - Forward 22      G - Group copy
  J - Jump to fn.ft    L - Length of file   N - New DIRectory   P - Print text
  R - Rename file      S - Stat of disk     T - Tag file        U - Untag file
  V - View text        X - Exit to CP/M     <SP> advances cursor -- B backs up

8k bytes free on DIRectory A:

  A: AMPRODSK.COM :
```

To use this program, simply enter the letter representing the desired function. Other items are requested by the program as needed. While DISK7 is running, you can select ■ new directory and user area, including drive E (foreign formats). In a single drive system, this allows viewing of files on a disk not containing DIR.COM. While DISK7 is running, disks can be swapped freely. This can be done using the ■ (new directory) command each time a disk is changed.

3.5.7 DIR Utility

The DIR program replaces the usual CP/M intrinsic DIR command, and provides additional capabilities. Both ambiguous (wild-card specifiers) and unambiguous filenames and types can be specified as in standard CP/M. However, there are trailing parameters preceded by a dollar sign ($), which extend the DIR.COM capabilities. Here are some examples of DIR use.

DIR	Display all files
DIR *.* $A	Display all files in all user areas of current disk
DIR *.* $D	Display all files on all drives in current user area
DIR *.* $AD	Display all files in all user areas and on all drives
DIR *.ASM	Display all .ASM files
DIR *.* $P	Send directory to LST: device (printer)
DIR *.* $F	Create a file containing directory contents (SD.DIR)

The $-suffixes can be combined to provide further usefulness.

3.5.8 STARTUP Utility

STARTUP provides the capability of executing multiple commands per command input line each time STARTUP itself is executed. The CONFIG utility provides for a single input command to be executed upon power-up. By using STARTUP as the power-up command, multiple commands or commands with trailing parameters can be executed at RESET or power-up. This utility is self-prompting. To enter a new command line, type

STARTUP S<RETURN>

Typing a question mark (?) displays a list of commands to which STARTUP responds:

```
STARTUP, Version 1.4
STARTUP Setup Command (?=Help)? ?
  Setup Mode Commands are --
        C -- Define STARTUP Multiple Command Line
        D -- Display STARTUP Values
        X -- Exit and Optionally Rewrite STARTUP
STARTUP Setup Command (?=Help)?
```

When you enter "X", you can rewrite STARTUP.COM with a new command line, and even a .COM file with a different filename. Several different command files, each generated through STARTUP can be used to execute various command lines. This provides a simple method for execution of lengthy or often used command sequences.

3.5.9 SWAPCOPY Utility

SWAPCOPY provides the means to transfer files from one disk to another using only one floppy disk drive. You can copy from A to A (same AMPRO format), A to E, or ▮ to A (differing formats). The destination disk must be already formatted. SWAPCOPY expects a filename argument which can either be ambiguous (using the ? and * characters) or not. To run SWAPCOPY, type

```
     SWAPCOPY filename<RETURN>

     SWAPCOPY *.COM<RETURN>  ---  copies all COM files to destination disk

     SWAPCOPY *.*<RETURN>    ---  copies all files to destination disk

     SWAPCOPY MYFILE.*<RETURN> -  copies all versions (types) of files named
                                  MYFILE to destination disk
```

SWAPCOPY prompts you to insert the SOURCE disk, then stores as much of the source file(s) into memory as possible.

SWAPCOPY then prompts you to insert the DESTINATION disk. You now remove the source disk and insert your destination disk.

With your DESTINATION disk in the drive (double check), press RETURN. The program will now write the file(s) stored in memory onto the destination disk.

If the program is finished copying, it will prompt you for the SYSTEM disk. If further copy operations are necessary, the program will prompt you for the SOURCE disk again.

3.5.10 UNERAse Utility

UNERAse permits recovery of accidentally deleted files. This program will work only if no write operations have been performed since the file or files were accidentally deleted. ANY write operation can destroy the information used by UNERAse to restore deleted files. To execute UNERA.COM, type

```
     UNERA filename.typ
```

Specified filenames can be either ambiguous or unambiguous:

```
     UNERA *.BAK      Recovers all .BAK type files
     UNERA MYFILE.*   Recovers all MYFILE.typ files, including
                      MYFILE without .typ specifier
```

For a help screen, type

```
     UNERA<RETURN>
```

3.6 GENERATING DIFFERENT SYSTEMS

The Little Board computer is supplied with both the standard CCP and ZCPR3. Your system disk contains these in two files: MOVCPM.COM and ZMOVCPM.COM respectively. ZCPR3 is the default module. Should you wish to have a standard CP/M version 2.2 system (using the CCP), use the following procedure:

Type the following command

```
     MOVCPM 61 *<RETURN>
```

The system will respond with

```
     CONSTRUCTING 61k CP/M vers 2.2
     READY FOR "SYSGEN" OR
     "SAVE 41 CPM61.COM"
     A0>
```

At this point, the system image is stored in memory. Using SYSGEN, you can write the new system on the disk system tracks

```
     SYSGEN<RETURN>
```

When SYSGEN requests the Source drive, respond with RETURN only. When SYSGEN requests the Destination drive, enter the appropriate drive designator as in the above example.

Another method that can be used is to save the system image on disk as a .COM file. Using MOVCPM or ZMOVCPM (for ZCPR3 systems) as shown above, type

```
SAVE 41 CPM61.COM
```

or

```
SAVE 41 ZCPM61.COM
```

Generating a system of your choice or for a particular need is simply a matter of typing

```
SYSGEN CPM61.COM<RETURN>
```

or

```
SYSGEN ZCPM61.COM<RETURN>
```

Using this method, SYSGEN responds only with a request for the Destination drive. This also permits changing from CCP to ZCPR3, or vice versa, with relative ease.

SECTION 4

PROGRAMMING INFORMATION

4.1 INTRODUCTION

This section discusses some of the programming techniques and peripheral device register addresses and requirements. Programming most devices is straight forward. However, the Floppy Disk Interface is relatively complex to program; it is not recommended that you attempt to write custom floppy disk routines. For more complete information on device functions not covered here, refer to the data sheets in Appendix B.

4.2 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 for each EPROM access.

In this configuration, the CPU non-maskable interrupt (NMI) input is not connected, only maskable interrupts can occur. All maskable interrupt modes are supported. The Z80A´s 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 the lowest.

- DART Channel A

- DART Channel B (lowest)

The FDC is indirectly included in the priority chain through the CTC Channel 3 (described later). In addition, the parallel printer port is also indirectly included in the interrupt priority chain. The parallel port BUSY input is connected to the DART RIB input.

4.3 MEMORY

When the EPROM enable bit in the Board Control Register (BCR) is low, the 4k EPROM is enabled in the lower 32k bytes of RAM. Note that when the EPROM is enabled, the EPROM contents are repeated eight times in lower memory. When the EPROM enable bit in the BCR is high, the EPROM disappears, leaving 64k bytes of RAM.

4.4 BOARD CONTROL REGISTER

An eight bit register located at I/O address 00H, is used to control a few main system functions (see Figure 4-1):

Bit	7	6	5	4	3	2	1	0
BCR	X	EEN	DDEN	SD1	DS4	DS3	DS2	DS1

BCR Bit	Signal	Function
7	X	Not used, don´t care.
6	EEN	EPROM Enable, 0=enabled in lower 32k 1=disabled
5	DDEN	Double Density Enable, 0 = enabled
4	SD1	Side 1, 1 = floppy side 1
3	DS4	Drive Select 4, 1 = select drive
2	DS3	Drive Select 3, 1 = select drive
1	DS2	Drive Select 2, 1 = select drive
0	DS1	Drive Select 1, 1 = select drive

Figure 4-1. Board Control Register Programming.

On power up or RESET, all bits in the BCR are automatically cleared.

4.5 COUNTER/TIMER CIRCUIT (CTC)

The CTC device contains four independent counter/timers addressed at the I/O locations shown in Table 4-1:

Table 4-1. 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 the counters is a 2.00 MHz clock. 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.

4.5.1 CTC Channel Functions

The Little Board CTC has the following assigned channel functions and options:

CTC Channel 0: Baud rate generator for DART Channel A. The CLK/TRG input for this channel is connected to 2.00 MHz. This channel can be used in either the Counter or Timer mode to generate a full range of baud rates.

CTC Channel 1: Baud rate generator for DART Channel B. The CLK/TRG input for this channel is connected to 2.00 MHz. This channel can be used in either the Counter or Timer mode to generate a full range of baud rates.

CTC Channel 2: Not used for hardware functions. The CLK/TRG input for this channel is inactive. This channel is available in the Timer mode, for applications software.

CTC Channel 3: Can be used as an optional interrupt for the FDC logic. The CLK/TRG input is connected to the FDC interrupt output signal. If FDC interrupts are desired, program this channel in Counter mode, with a count of 1, triggerable on a rising edge. Each time the FDC outputs its interrupt signal, CTC Channel 3 will cause a system interrupt. The Little Board standard BIOS does not support this channel.

4.6 SERIAL I/O PORTS

A Z80 DART device is primarily used as the interface for the Little Board´s two serial I/O ports. In addition, three of the DART I/O signals are used for other interfaces.

The DART internal registers are accessed through four, non-consecutive I/O addresses. Each register is both read and write. In order to correctly read the DART external status signals (CTS, RI, DCD, etc), a Reset External Status command must first be sent to the DART channel.

Table 4-2. DART Register, I/O Addresses.

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

4.6.1 Channel A Signals

DART Channel A input/output signals are defined in Table 4-3. Note that the high/low baud rate select for Channel A uses the DTRA signal.

Table 4-3. DART Channel A Signal Definitions.

Signal Name	DART Pin	Function
Transmit Data	TXDA	output to RS232C
Receive Data	RXDA	input from RS232C
Handshake Out	RTSA	output to RS232C
Handshake Input	CTSA	input from RS232C
Data Clock	RXCA, TXCA	input from CTC ZC/TC0 pin
Additional function:		
Low Baud Select	DTRA	Serial Port A baud rate mode

4.6.2 Channel B Signals

DART Channel B input/output signals are defined in Table 4-4. Note the additional functions.

Table 4-4. DART Channel B Signal Definitions.

Signal Name	DART Pin	Function
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:		
Drive Ready	DCDB	input from FDC interface
Printer BUSY*	RIB	input from printer interface

4.7 BAUD RATE GENERATION

Both serial ports use signals output by the CTC for baud rates up to 9600. This is accomplished by setting the associated CTC channel to Counter mode, programming the required CTC channel time constant, and programming the DART channel prescale factor (16, 32, or 64).

Channel A, however, can be programmed to use a separate 615.385 kHz signal for its input. This produces baud rate signals of 9600, 19.2k, and 38.4k baud. When the DART DTRA output is active, Channel A is in the low speed mode. When DTRA is inactive, Channel A is in the high speed mode.

4.7.1 Below 9600 Baud

Serial port A baud rate is determined by CTC channel 0, and serial port B baud rate is determined by CTC channel 1. Program each CTC and DART channel as shown in the following tables (see data sheets in Appendix B).

Table 4-5. CTC and DART Modes.

CTC Interrupt	Disable
CTC Mode	Table 4-6
CTC Prescaler	*16
CTC CLK/TRIG edge	Either
CTC Timer Trigger	*Set to automatic
CTC Time Constant	Table 4-6
DART Scale Factor	Table 4-6
* are don´t care in Counter mode	

Table 4-6. CTC and DART Program Values

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

To use this method of baud rate selection, DART Channel A DTRA output must be set active (DTRA = 1).

4.7.2 Above 9600 Baud

To select the high baud range, the DTRA output must be cleared (0), and CTC Channel 0 turned off with a software reset. The values shown in Table 4-7 represent the required DART Scale Factor to be written to the DART. For complete details on DART programming, see the data sheet in Appendix B. To program DART Channel A for the higher baud rates:

CTC Channel 0 Programming:	Must be issued a Software Reset (write a 03H byte as a control word to CTC Channel 0)
DART DTRA Signal:	Cleared (DTRA=0)
DART Scale Factor:	Table 4-7

Table 4-7. High Baud Rate DART Settings

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

4.8 FLOPPY DISK INTERFACE

A Western Digital WD1770 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 WD1770, read and write registers in the FDC occupy unequal addresses (this differs from the WD1770 data sheet description).

The Little Board floppy disk interface is relatively complex to program. It is not recommended that you attempt to write custom routines. The following information is for reference only.

Table 4-8. WD1770 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 4-9. Additional Interface 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

4.9 PARALLEL PRINTER PORT

The parallel interface supports eight data bits (D1 - D8), a Data Strobe, and a printer Busy signal. With the exception of Busy, these signals are accessed as shown in Table 4-10.

Table 4-10. Parallel Printer Port I/O Addresses

Address	Function
01H	8-bit data register written to by CPU. CPU data bit 0 = printer D1 through 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 automatically cleared (Data Strobe = 0) upon 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. In order to correctly read the 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: if Busy = 1, RIB detects a low (inactive) state. Conversely if Busy = 0, RIB detects a high (active) state.

SECTION 5

THEORY OF OPERATION

5.1 INTRODUCTION

This section provides more detailed information on the Little Board theory of operation. No information on the internal operation of the LSI components is included. Please refer to the manufacturers' data (listed in Appendix B) for specific details. Figure 5-1 is a block diagram of the Little Board.

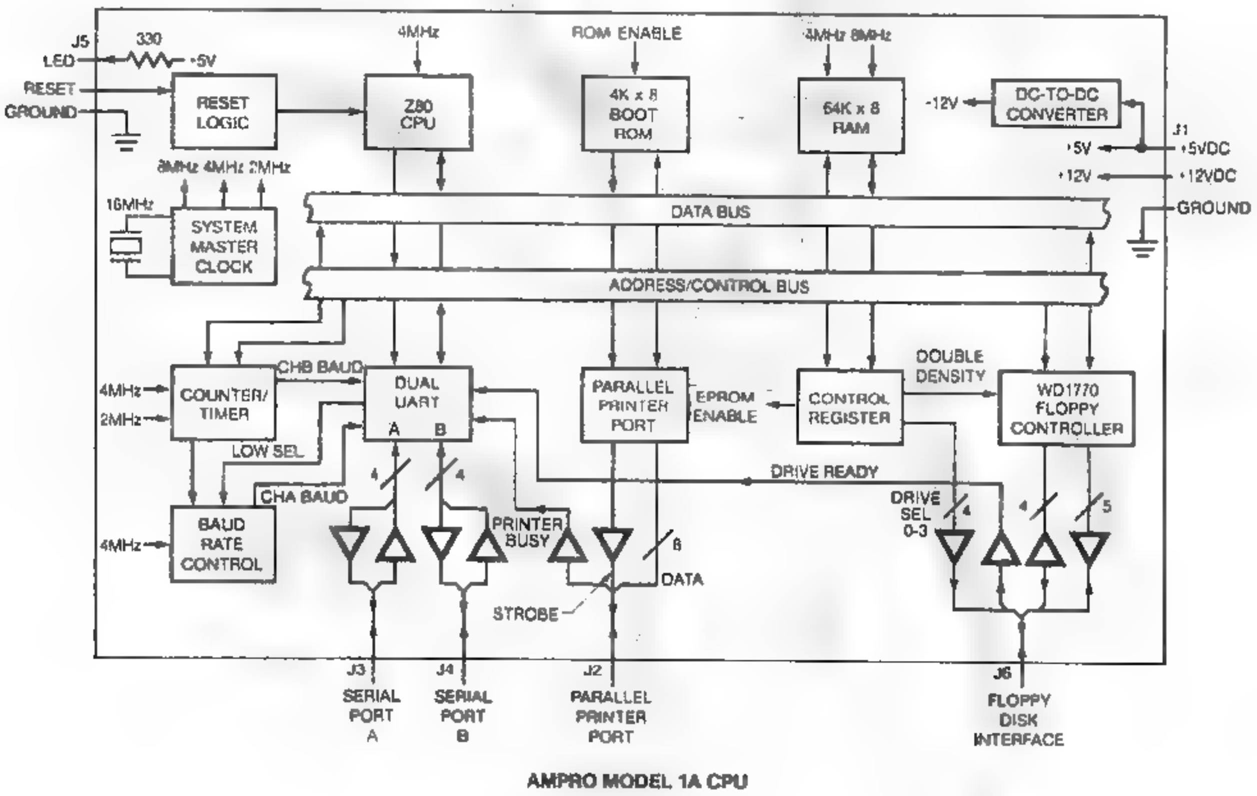


Figure 5-1. Little Board Block Diagram

5.2 CPU, MEMORY, AND TIMING

The main system time base is provided by a 16MHz oscillator module. A binary counter is used to provide three system clocks: 8MHz, 4MHz, and 2MHz. The 4MHz signal is used by the Z80A, Counter Timer Circuit (CTC), and Dual Asynchronous Receiver/Transmitter (DART) devices. The 8MHz signal provides the clock input to the WD1770 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. The FDC interrupt signal is used as the trigger input to the CTC channel 3, thereby allowing the CTC to function as an interrupt controller for the FDC. DART Channel B RIB input is connected to the printer port BUSY signal. This permits the DART to serve as an interrupt controller for the printer port.

All control signals for the 64k dynamic RAM are derived from the system's 4 and 8MHz clocks and the Z80A refresh signal. RAM devices with access times up to 200 nS can be used.

When a memory read or write occurs with address line A15 set to one, 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 pair of two-to-four demultiplexers decode the device select addresses for all of the Little Board's I/O devices. The minimum number of address lines necessary to generate the seven required I/O device select lines is decoded. Table 5-1 shows the device select addresses in binary.

The Board Control Register (BCR), consisting of an octal output latch, is used primarily to control several functions associated with the floppy disk controller. One bit in the BCR also serves to enable or disable the EPROM device. The BCR is cleared by the board's RESET signal, selecting the EPROM at power-up or when the RESET signal is active.

Table 5-1. I/O Device Addresses

Device Select	I/O Address (Binary)
Board Control Register	000XXX00
Parallel Port Data Latch	000XXX01
Parallel Port Strobe Set	000XXX10
Parallel Port Strobe Clear	000XXX11
CTC	01XXXXXX
DART	10XXXXXX
FDC	11XXXXXX
Unused, available to user	0011XXXX
Unused, reserved	0010XXXX

5.3 SERIAL I/O PORTS

The two channels of the Z80A DART are provided with a wide range of baud rates from the CTC device. Channel A of the DART has a second baud rate clock source (615.385kHz), which is obtained by dividing the 16MHz 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 source.

DART RTSA and RTSB output signals generate each channel's output handshake signal. Several of the DART's input and output signals are used for other purposes:

- DTRA is used to select between high and low baud rate modes for serial I/O channel A.
- DCDB is used by the floppy disk interface.
- RIB is used by the parallel I/O port.

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

5.4 PARALLEL I/O PORT

An octal D-latch with a 24mA current sinking capacity is used to drive the eight parallel printer port data lines. 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. An open collector provides the Data Strobe output.

A schmit-trigger buffer conditions the printer Busy input, which is sensed by the RIB input of the DART.

5.5 FLOPPY DISK INTERFACE

Nearly all of the logic required for the floppy disk interface is provided by the WD1770 Floppy Disk Controller (FDC) device. Only the drive and side-selection and interface input signal buffering require additional devices.

The Board Control Register is used to set the state of the four drive select lines, the side select line, and the WD1770 density select input.

A schmit-trigger buffer is used to condition the floppy disk interface Ready input, which is connected to the DART DCDB input signal. This can be optionally used with floppy disk drives that provide this signal.

APPENDIX A

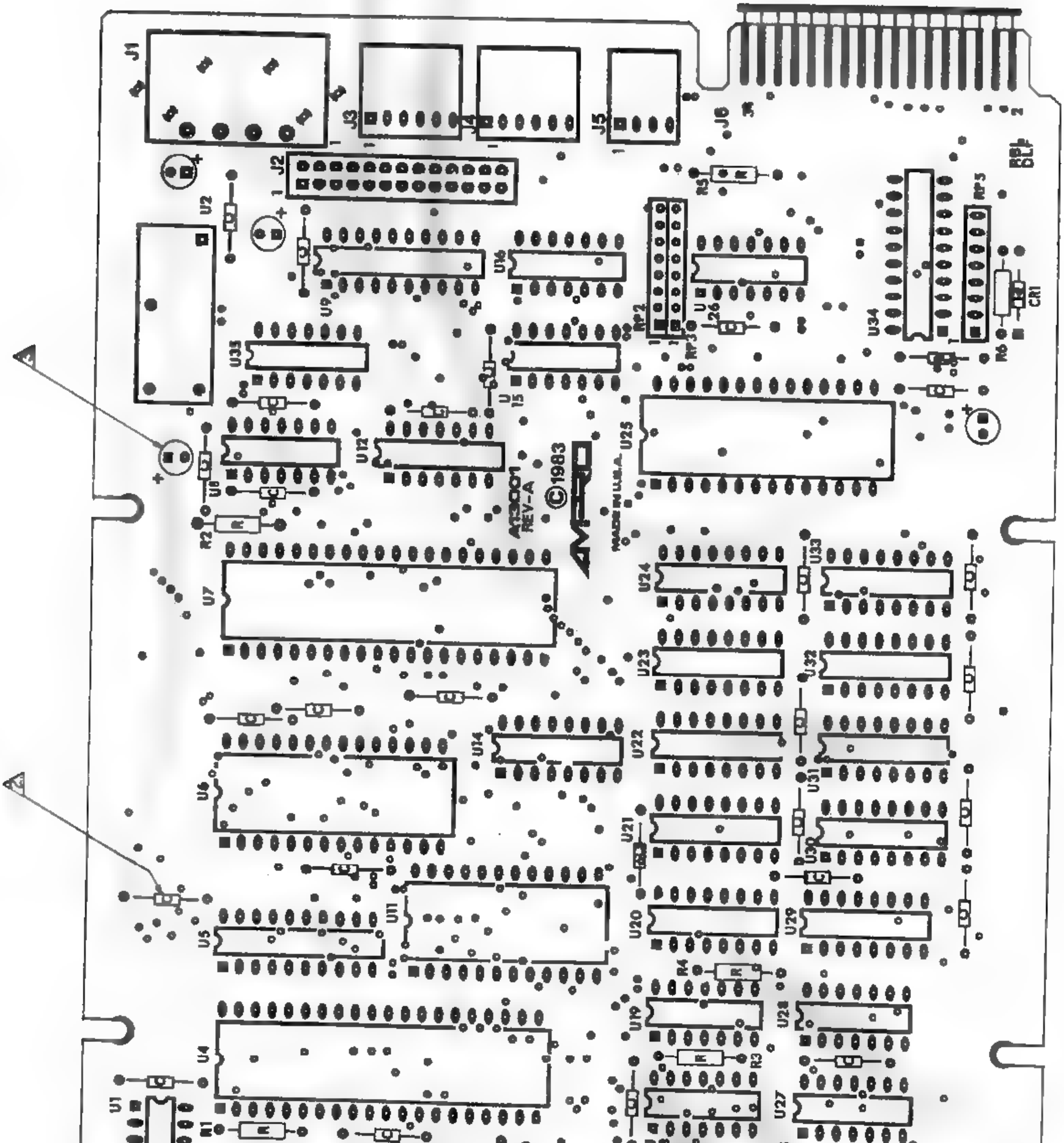
COMPONENT LOCATIONS AND PARTS LIST

LITTLE BOARD PARTS LIST

ASSEMBLY: A60001-001   REV: A   DATE: 2/03/84

REF	PART NUMBER	QTY	DESCRIPTION
	43-00010-010	1	BRD,PC-AMPRO-SER 1 CPU
C1	82-00012-010	31	CAP CER .1UF +80%-20% 50V
C2	86-00001-010	4	CAP ELC 10UF 20% 25V
CR1	60-00005-010	1	DIODE 1N4148
J1	34-00095-010	1	CONN HEADER 4POS SIL RT/AG
J2	34-00085-010	1	CONN HEADER 26POS .100-OC STR
J3,4	34-00082-010	2	CONN HEADER 6POS SIL RT/AG
J5	34-00083-010	1	CONN HEADER 4POS SIL RT/AG
J7	34-00004-010	1	CONN HEADER 2POS SIL .100-OC
J7 (REF)	34-00031-010	1	CONN SHUNT 2POS .100-OC
R1,5	71-00009-010	2	RES CF 1K 05% 1/4W
R2,6	71-00007-010	2	RES CF 4700 05% 1/4W
R3,4	71-00092-010	2	RES CF 39 05% 1/4W
RP1,4	75-00008-010	2	RES PK 8SIP 7-4700 10%
RP2,3	75-00038-010	2	RES PK 8SIP 7-1K 02%
RP5	75-00037-010	1	RES PK 8SIP 7-330 02%
U1	67-00137-010	1	IC 74LS169 (National or Fairchild)
U2	69-00101-010	1	HYBRID -12V CONV (ELPAC #CB3811) +3 to +7 TO -12V +/-0.5% 0 to 25mA
U3	30-00032-010	1	OSC 16.000MHZ
U4	69-00011-010	1	IC Z80A CPU
U4 (REF)	34-00006-010	1	IC SOCKET 40POS D/W
U5	67-00034-010	1	IC 74LS273 OCTAL D F/F
U6	69-00019-010	1	IC Z80A-CTC
U7	69-00020-010	1	IC Z80A-DART
U8	65-00004-010	1	IC 1488
U9	67-00006-010	1	IC 74LS374 OCTAL D F/F TRI S
U10	67-00043-010	1	IC 74S163 SYNC 4-BIT BIN CTR., SYNC CLR
U11	69-00014-010	1	IC 2732 (PROGRAMMED: A75501-302) 350nS
U11 (REF)	34-00005-010	1	IC SOCKET 24POS D/W AM2732-1DC
U12	65-00005-010	1	IC 1489A
U13	67-00041-010	1	IC 74S32 QUAD OR
U14	67-00023-010	1	IC 74LS139 DUAL 2:4 DECODER
U15	67-00009-010	1	IC 74LS08 QUAD AND
U16,26	67-00048-010	2	IC 7406 O.C. HEX INV
U17	67-00015-010	1	IC 74LS74 DUAL D F/F
U18,27	67-00040-010	2	IC 74S74 DUAL D F/F
U19	67-00039-010	1	IC 74S00 QUAD NAND
U20,29	67-00026-010	2	IC 74LS157 QUAD 2:1 MUX
U21-24,30-33	69-00022-010	8	IC 4164
U25	69-00081-010	1	IC WD1770
U28	67-00104-010	1	IC 74S02 QUAD NOR
U34	67-00033-010	1	IC 74LS244 OCTAL NON-INV BUFFER TRI-S
U35	67-00004-010	1	IC 74LS32 QUAD OR

TMS wuz here


J1
J2
J3
J4
J5
J6
J7
1
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
U16
U17
U18
U19
U20
U21
U22
U23
U24
U25
U26
U27
U28
U29
U30
U31
U32
U33
U34
U35
R1
R2
R3
R4
R5
R6
RP1
RP2
RP3
RP4
RP5
CR1
R
C
+
A13001
REV-A
©1983
AMPRO
MADE IN U.S.A.
BBL
BLP

APPENDIX B

SUMMARY OF I/O PORTS AND CONNECTOR PINOUTS

The following figure and tables list the I/O port addresses, signals, and connector pinouts.

Bit	7	6	5	4	3	2	1	0
BCR	X	EEN	DDEN	SD1	DS4	DS3	DS2	DS1

BCR Bit	Signal	Function
7	X	Not used, don´t care.
6	EEN	EPROM Enable, 0=enabled in lower 32k 1=disabled
5	DDEN	Double Density Enable, 0 = enabled
4	SD1	Side 1, 1 = floppy side 1
3	DS4	Drive Select 4, 1 = select drive
2	DS3	Drive Select 3, 1 = select drive
1	DS2	Drive Select 2, 1 = select drive
0	DS1	Drive Select 1, 1 = select drive

Figure B-1. Board Control Register

Table B-1. CTC Register Addresses

Address	CTC Channel
40H	0
50H	1
60H	2
70H	3
All Channels are write-only	

Table B-2. DART Register, I/O Addresses.

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

Table B-3. DART Channel A Signal Definitions.

Signal Name	DART Pin	Function
Transmit Data	TXDA	output to RS232C
Receive Data	RXDA	input from RS232C
Handshake Out	RTSA	ouput to RS232C
Handshake Input	CTSA	input from RS232C
Data Clock	RXCA, TXCA	input from CTC ZC/TC0 pin
Additional function:		
Low Baud Select	DTRA	Serial Port A baud rate mode

Table B-4. DART Channel B Signal Definitions.

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

Table B-5. FDC 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 B-6. Additional Interface 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

Table B-7. Parallel Printer Port I/O Addresses

Address	Function
01H	8-bit data register written to by CPU. CPU data bit 0 = printer D1 through 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.

Table B-8. Power Connector, J1.

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

Table B-9. Parallel Printer Cable Connections, J2.

Pin	Signal Name	Function	Printer Signal Pins
1	DS*	Data Strobe to printer (low)	1
3	Data 1	LSB of printer data	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	9
19	Not Used		
21	BUSY	Printer BUSY input to LBC	11
23	Not Used		
25	Not Used		
2-26	All even	Signal grounds	19-27 & 29

Table B-10. External Serial I/O Cable Connections, J3,J4.

Pin	Signal Name	Function	DB-25 Pin (DCE)
1	Ground	Protective Ground	1
2	Ground	Signal Ground	7
3	TxD	Data Ouput	3
4	HSO	Handshake Signal Out ~~[illegible]~~	5 CTS
5	RxD	Data Input	2
6	HSI	Handshake Signal In ~~[illegible]~~	20 DTR
Note: AMPRO Cable P/N is A60005-001			

Table B-11. Reset/Power-On 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

Table B-12. Floppy Disk Interface Connections, J6.

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

TABLE OF I/O PORTS

ADDRESS	INPUT/OUTPUT	FUNCTION
00H	Output	Board Control Register
01H	Output	Parallel Printer Data Register
02H	Output	Parallel Printer Data Stobe Set
03H	Output	Parallel Printer Data Strobe Clear
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 of the above ports are not unambiguously decoded. This is discussed in the Theory of Operation section of this publication.

APPENDIX C

CP/M USER GROUPS

AMPRO User Group Bulletin Board

Provides a focal point for dissemination of public domain software, and to exchange AMPRO-related hardware and software applications. The bulletin board is not owned by AMPRO Computers, but is provided and maintained by AMPRO users for the benefit of AMPRO users.

(408) 258-8128, 24 hrs/day, 7 days/week

300/1200 baud

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

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.

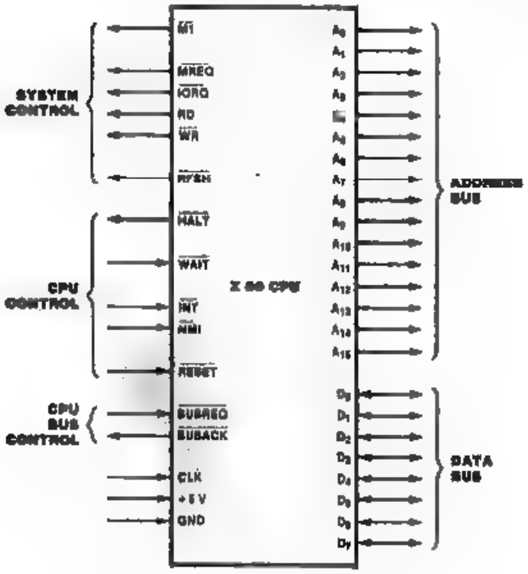


Figure 1. Pin Functions

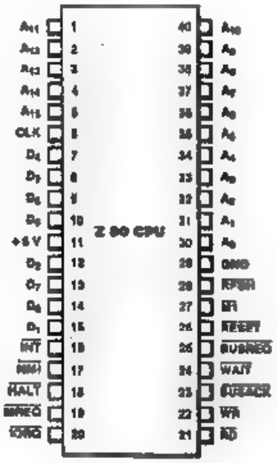


Figure 2. Pin Assignments

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.

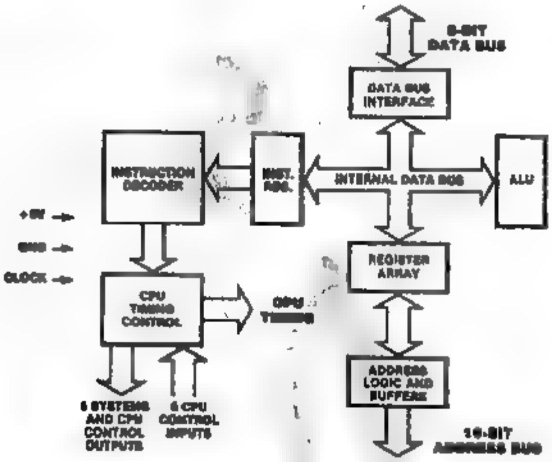


Figure 3. Z80 CPU Block Diagram

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.

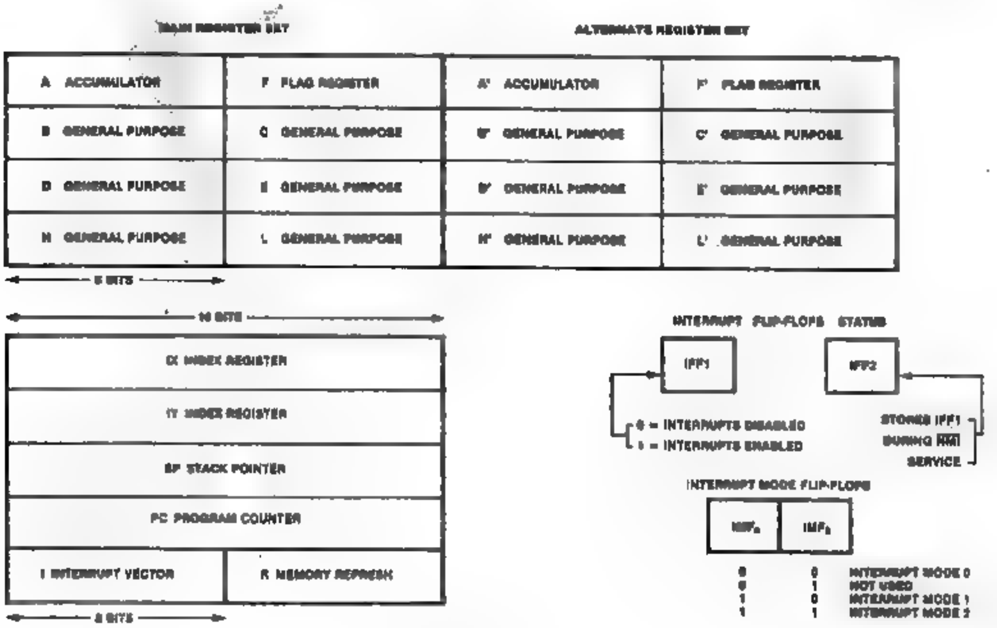


Figure 4. CPU Registers

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: B — High byte C — Low byte D — High byte E — Low byte 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.

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 recognifion 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	$IFF_1 \rightarrow IFF_2$ (Maskable interrupt $\overline{INT}$ disabled)
RETN instruction execution	IFF_2	•	$IFF_2 \rightarrow IFF_1$ at completion of an $\overline{NMI}$ service routine.

Table 2. State of Flip-Flops

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	
LD r, r'	r ← r'	•	•	X	•	X	•	•	•	01 r r'		1	1	4	r, r'	Reg.
LD r, n	r ← n	•	•	X	•	X	•	•	•	00 r 110 ← n →		2	2	7	000 001	B 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 01 r 101 ← d →	DD	3	5	19	011 100 101	E H L
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	X	•	•	•	11 111 101 01 r 110 ← d →	FD	3	5	19	111	A
LD (HL), r	(HL) ← r	•	•	X	•	X	•	•	•	01 110 r		1	2	7		
LD (IX+d), r	(IX+d) ← r	•	•	X	•	X	•	•	•	11 011 101 01 110 r ← d →	DD	3	5	19		
LD (IY+d), r	(IY+d) ← r	•	•	X	•	X	•	•	•	11 111 101 01 110 r ← d →	FD	3	5	19		
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00 110 110 ← n →	36	2	3	10		
LD (IX+d), n	(IX+d) ← n	•	•	X	•	X	•	•	•	11 011 101 00 110 110 ← d → ← n →	DD 36	4	5	19		
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	•	11 111 101 00 110 110 ← d → ← n →	FD 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 ← n → ← 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 ← n → ← n →	32	3	4	13		
LD A, I	A ← I	↕	↕	X	0	X	IFF	0	•	11 101 101 01 010 111	ED 57	2	2	9		
LD A, R	A ← R	↕	↕	X	0	X	IFF	0	•	11 101 101 01 011 111	ED 5F	2	2	9		
LD I, A	I ← A	•	•	X	•	X	•	•	•	11 101 101 01 000 111	ED 47	2	2	9		
LD R, A	R ← A	•	•	X	•	X	•	•	•	11 101 101 01 001 111	ED 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.

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 ← n → ← n →		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11 011 101 00 100 001 ← n → ← n →	DD 21	4	4	14	
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11 111 101 00 100 001 ← n → ← n →	FD 21	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	•	00 101 010 ← n → ← n →	2A	3	5	16	
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	•	•	X	•	X	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	ED	4	6	20	
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	•	•	X	•	X	•	•	•	11 011 101 00 101 010 ← n → ← n →	DD 2A	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	•	•	X	•	X	•	•	•	11 111 101 00 101 010 ← n → ← n →	FD 2A	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00 100 010 ← n → ← n →	22	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	X	•	X	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	ED	4	6	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	X	•	X	•	•	•	11 011 101 00 100 010 ← n → ← n →	DD 22	4	6	20	
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	X	•	X	•	•	•	11 111 101 00 100 010 ← n → ← n →	FD 22	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 11 111 001	DD F9	2	2	10	
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11 111 101 11 111 001	FD F9	2	2	10	
PUSH qq	(SP-2) ← qq_L (SP-1) ← qq_H SP ← SP - 2	•	•	X	•	X	•	•	•	11 qq0 101		1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP-2) ← IX_L (SP-1) ← IX_H SP ← SP - 2	•	•	X	•	X	•	•	•	11 011 101 11 100 101	DD E5	2	4	15	
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H SP ← SP - 2	•	•	X	•	X	•	•	•	11 111 101 11 100 101	FD E5	2	4	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP) SP ← SP + 2	•	•	X	•	X	•	•	•	11 qq0 001		1	3	10	
POP IX	IX_H ← (SP+1) IX_L ← (SP) SP ← SP + 2	•	•	X	•	X	•	•	•	11 011 101 11 100 001	DD E1	2	4	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP) SP ← SP + 2	•	•	X	•	X	•	•	•	11 111 101 11 100 001	FD 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

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

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

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
LDD	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1	•	•	X	0	X	① ↕	0	•	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1 Repeat until BC = 0	•	•	X	0	X	② 0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A − (HL) HL ← HL + 1 BC ← BC − 1	↕	③ ↕	X	↕	X	① ↕	1	•	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A − (HL) HL ← HL + 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0	↕	③ ↕	X	↕	X	① ↕	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A − (HL) HL ← HL − 1 BC ← BC − 1	↕	③ ↕	X	↕	X	① ↕	1	•	11 101 101 10 101 001	ED A9	2	4	16	
CPDR	A − (HL) HL ← HL − 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0	↕	③ ↕	X	↕	X	① ↕	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) 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

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
ADD A, r	A ← A + r	↕	↕	X	↕	X	V	0	↕	10 $\boxed{000}$ r		1	1	4	
ADD A, n	A ← A + n	↕	↕	X	↕	X	V	0	↕	11 $\boxed{000}$ 110 ← n →		2	2	7	
ADD A, (HL)	A ← A + (HL)	↕	↕	X	↕	X	V	0	↕	10 $\boxed{000}$ 110		1	2	7	
ADD A, (IX+d)	A ← A + (IX+d)	↕	↕	X	↕	X	V	0	↕	11 011 101 10 $\boxed{000}$ 110 ← d →	DD	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	↕	↕	X	↕	X	V	0	↕	11 111 101 10 $\boxed{000}$ 110 ← d →	FD	3	5	19	
ADC A, s	A ← A+s+CY	↕	↕	X	↕	X	V	0	↕	$\boxed{001}$					s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the $\boxed{000}$ in the ADD set above.
SUB s	A ← A−s	↕	↕	X	↕	X	V	1	↕	$\boxed{010}$					
SBC A, s	A ← A−s−CY	↕	↕	X	↕	X	V	1	↕	$\boxed{011}$					
AND s	A ← A ∧ s	↕	↕	X	1	X	P	0	0	$\boxed{100}$					
OR s	A ← A ∨ s	↕	↕	X	0	X	P	0	0	$\boxed{110}$					
XOR s	A ← A ⊕ s	↕	↕	X	0	X	P	0	0	$\boxed{101}$					
CP s	A − s	↕	↕	X	↕	X	V	1	↕	$\boxed{111}$					
INC r	r ← r + 1	↕	↕	X	↕	X	V	0	•	00 r $\boxed{100}$		1	1	4	
INC (HL)	(HL) ← (HL)+1	↕	↕	X	↕	X	V	0	•	00 110 $\boxed{100}$		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	↕	↕	X	↕	X	V	0	•	11 011 101 00 110 $\boxed{100}$ ← d →	DD	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	↕	↕	X	↕	X	V	0	•	11 111 101 00 110 $\boxed{100}$ ← d →	FD	3	6	23	
DEC m	m ← m−1	↕	↕	X	↕	X	V	1	•	$\boxed{101}$					m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace $\boxed{100}$ with $\boxed{101}$ in opcode.

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

General-Purpose Arithmetic and CPU Control Groups

Mnemonic	Symbolic Operation	S	Z		Flags H		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 ← $\overline{A}$	•	•	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 01 000 100	ED 44	2	2	8	Negate acc. (two's complement).
CCF	CY ← $\overline{CY}$	•	•	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 01 000 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 01 010 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

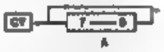
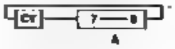
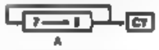
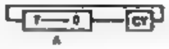
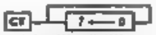
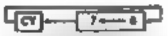
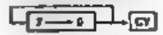
NOTES: IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.
∗ indicates interrupts are not sampled at the end of EI or DI.

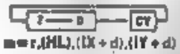
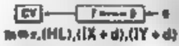
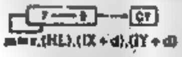
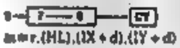
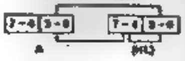
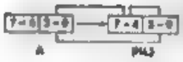
16-Bit Arithmetic Group

Mnemonic	Symbolic Operation	S	Z		H		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. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	HL ← HL + ss + CY	↕	↕	X	X	X	V	0	↕	11 101 101 01 ss1 010	ED	2	4	15	
SBC HL, ss	HL ← HL − ss − CY	↕	↕	X	X	X	V	1	↕	11 101 101 01 ss0 010	ED	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	↕	11 011 101 00 pp1 001	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	↕	11 111 101 00 rr1 001	FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 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 00 100 011	DD 23	2	2	10	
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 111 101 00 100 011	FD 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 00 101 011	DD 2B	2	2	10	
DEC IY	IY ← IY − 1	•	•	X	•	X	•	•	•	11 111 101 00 101 011	FD 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		H		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 00 000 r	CB	2	2	8	Rotate left circular register r.
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 110	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	11 011 101 11 001 011 ← d → 00 000 110	DD CB	4	6	23	
RLC (IY+d)		↕	↕	X	0	X	P	0	↕	11 111 101 11 001 011 ← d → 00 000 110	FD CB	4	6	23	
RL m	 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 or RLC's with shown code.
RRC m	 m≡r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	001					

	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
Rotate and Shift Group (Continued)	RR m	 m≡r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{011}$					
	SLA m	 m≡r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{100}$					
	SRA m	 m≡r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{101}$					
	SRL m	 m≡r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{111}$					
	RLD		↕	↕	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL).
	RRD		↕	↕	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	The content of the upper half of the accumulator is unaffected.
Bit Set, Reset and Test Group	BIT b, r	$Z \leftarrow \overline{r_b}$	X	↕	X	1	X	X	0	•	11 001 011 01 b r	CB	2	2	8	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
	BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	X	↕	X	1	X	X	0	•	11 001 011 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 11 001 011 ← d → 01 b 110	DD CB	4	5	20	
	BIT b, (IY+d)$_b$	$Z \leftarrow \overline{(IY+d)_b}$	X	↕	X	1	X	X	0	•	11 111 101 11 001 011 ← d → 01 b 110	FD CB	4	5	20	b Bit Tested 000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
	SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 $\boxed{11}$ b r	CB	2	2	8	
	SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 $\boxed{11}$ b 110	CB	2	4	15	
	SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 011 101 11 001 011 ← d → $\boxed{11}$ b 110	DD CB	4	6	23	
	SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 111 101 11 001 011 ← d → $\boxed{11}$ b 110	FD CB	4	6	23	
	RES b, m	$m_b \leftarrow 0$ m ≡ r, (HL), (IX+d), (IY+d)	•	•	X	•	X	•	•	•	$\boxed{10}$					To form new opcode replace $\boxed{11}$ of SET b, s with $\boxed{10}$. Flags and time states for SET instruction.

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

	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
Jump Group	JP nn	PC ← nn	•	•	X	•	X	•	•	•	11 000 011 ← n → ← n →	C3	3	3	10	
	JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010 ← n → ← n →		3	3	10	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
	JR e	PC ← PC+e	•	•	X	•	X	•	•	•	00 011 000 ← e−2 →	18	2	3	12	
	JR C, e	If C = 0, continue If C = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00 111 000 ← e−2 →	38	2 2	2 3	7 12	If condition not met. If condition is met.
	JR NC, e	If C = 1, continue If C = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 110 000 ← e−2 →	30	2 2	2 3	7 12	If condition not met. If condition is met.
	JP Z, e	If Z = 0 continue If Z = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00 101 000 ← e−2 →	28	2 2	2 3	7 12	If condition not met. If condition is met.
	JR NZ, e	If Z = 1, continue If Z = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 100 000 ← e−2 →	20	2 2	2 3	7 12	If condition not met. 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 11 101 001	DD E9	2	2	8	

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 11 101 001	FD E9	2	2	8	
DJNZ, e	B ← B−1 If B = 0, continue	•	•	X	•	X	•	•	•	00 010 000 ← e−2 →	10	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 (SP−2) ← PC_L PC ← nn	•	•	X	•	X	•	•	•	11 001 101 ← n → ← n →	CD	3	5	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	X	•	X	•	•	•	11 cc 100 ← n → ← n →		3	3	10	If cc is false.
												3	5	17	If cc is true.
RET	PC_L ← (SP) 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. 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
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 101 101 01 001 101	ED 4D	2	4	14	
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11 101 101 01 000 101	ED 45	2	4	14	
RST p	(SP−1) ← PC_H (SP−2) ← PC_L PC_H ← 0 PC_L ← p	•	•	X	•	X	•	•	•	11 t 111		1	3	11	t p 000 00H 001 08H 010 10H 011 18H 100 20H 101 28H 110 30H 111 38H

NOTE: [1]RETN loads $IFF_2 \rightarrow IFF_1$

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 ← n →	DB	2	3	11	n to A_0 ~ A_7 Acc. to A_8 ~ A_{15}
IN r, (C)	r ← (C) If r = 110 only the flags will be affected	↕	↕	X	↕	X	P	0	•	11 101 101 01 r 000	ED	2	3	12	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INI	(HL) ← (C) B ← B−1 HL ← HL + 1	X	① ↕	X	X	X	X	1	X	11 101 101 10 100 010	ED A2	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INIR	(HL) ← (C) B ← B−1 HL ← HL + 1 Repeat until B = 0	X	② 1	X	X	X	X	1	X	11 101 101 10 110 010	ED B2	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
												2	4 (If B=0)	16	
IND	(HL) ← (C) B ← B − 1 HL ← HL−1	X	① ↕	X	X	X	X	1	X	11 101 101 10 101 010	ED AA	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INDR	(HL) ← (C) B ← B−1 HL ← HL−1 Repeat until B = 0	X	② 1	X	X	X	X	1	X	11 101 101 10 111 010	ED BA	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
												2	4 (If B=0)	16	
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11 010 011 ← n →	D3	2	3	11	n to A_0 ~ A_7 Acc. to A_8 ~ A_{15}
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 101 101 01 r 001	ED	2	3	12	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OUTI	(C) ← (HL) B ← B−1 HL ← HL + 1	X	① ↕	X	X	X	X	1	X	11 101 101 10 100 011	ED A3	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OTIR	(C) ← (HL) B ← B−1 HL ← HL + 1 Repeat until B = 0	X	② 1	X	X	X	X	1	X	11 101 101 10 110 011	ED B3	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
												2	4 (If B=0)	16	
OUTD	(C) ← (HL) B ← B−1 HL ← HL−1	X	① ↕	X	X	X	X	1	X	11 101 101 10 101 011	ED AB	2	4	16	C to A_0 ~ A_7 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.

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) B ← B − 1 HL ← HL − 1 Repeat until B = 0	X	① 1	X	X	X	X	1	X	11 101 101 ED 10 111 011	2 2	5 (If B≠0) 4 (If B=0)	21 16	C to A_0 ~ A_7 B to A_8 ~ A_{15}

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	↕	1	•	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 >.

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.

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

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

$\overline{\mathrm{HALT}}$. *Halt State* (output, active Low). $\overline{\mathrm{HALT}}$ 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.

$\overline{\mathrm{INT}}$. *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. $\overline{\mathrm{INT}}$ is normally wire-ORed and requires an external pullup for these applications.

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

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

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

$\overline{\mathrm{NMI}}$. *Non-Maskable Interrupt* (input, negative edge-triggered). $\overline{\mathrm{NMI}}$ has a higher priority than $\overline{\mathrm{INT}}$. $\overline{\mathrm{NMI}}$ 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.

$\overline{\mathrm{RD}}$. *Read* (output, active Low, 3-state). $\overline{\mathrm{RD}}$ 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.

$\overline{\mathrm{RESET}}$. *Reset* (input, active Low). $\overline{\mathrm{RESET}}$ 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 $\overline{\mathrm{RESET}}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

$\overline{\mathrm{RFSH}}$. *Refresh* (output, active Low). $\overline{\mathrm{RFSH}}$, together with $\overline{\mathrm{MREQ}}$, 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.

$\overline{\mathrm{WAIT}}$. *Wait* (input, active Low). $\overline{\mathrm{WAIT}}$ 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 $\overline{\mathrm{WAIT}}$ periods can prevent the CPU from refreshing dynamic memory properly.

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

CPU Timing

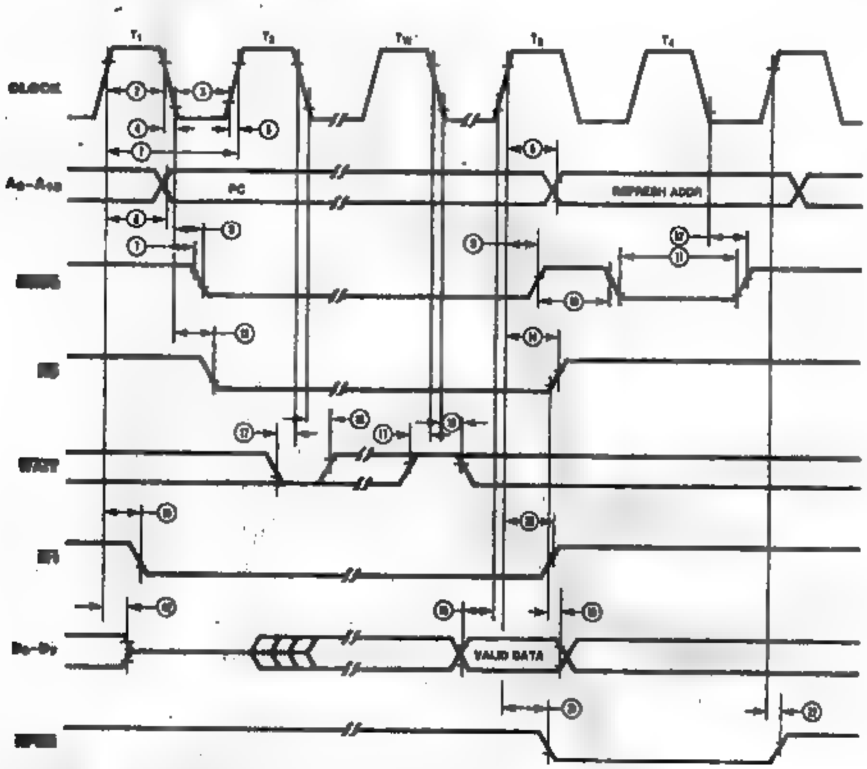
The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

- 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, $\overline{\text{MREQ}}$ goes active. When active, $\overline{\text{RD}}$ indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the $\overline{\text{WAIT}}$ input with the falling edge of clock state T_2. During clock states T_3 and T_4 of an $\overline{\text{M1}}$ 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 ancillary devices.

Figure 5. Instruction Opcode Fetch

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.

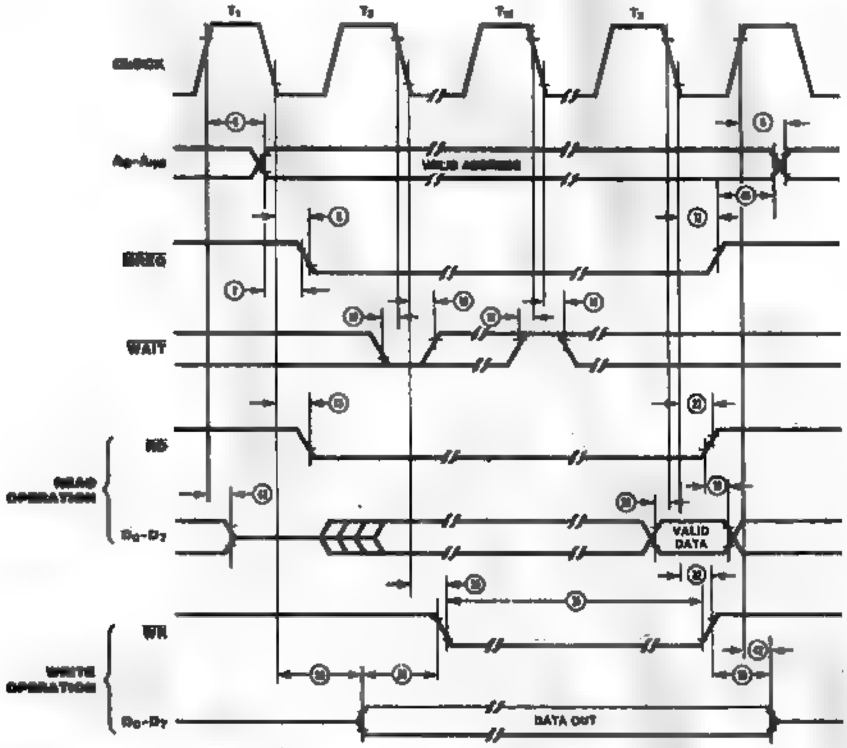
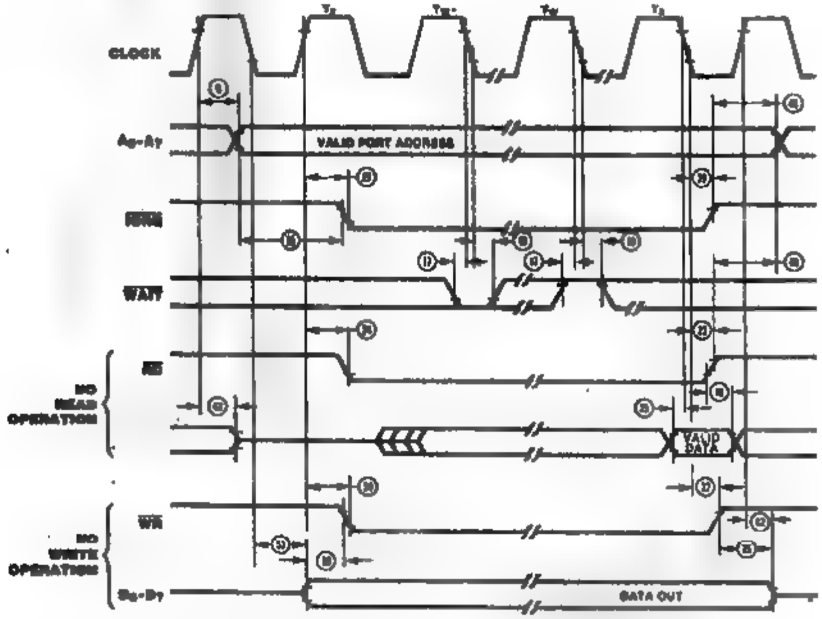


Figure 6. Memory Read or Write Cycles

CPU Timing (Continued)

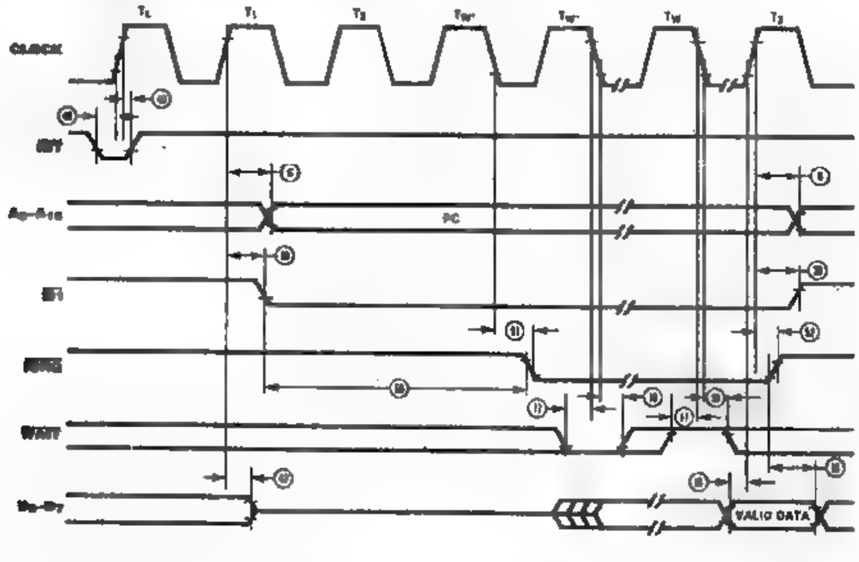
Input or Output Cycles. Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state (T_W). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.



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.



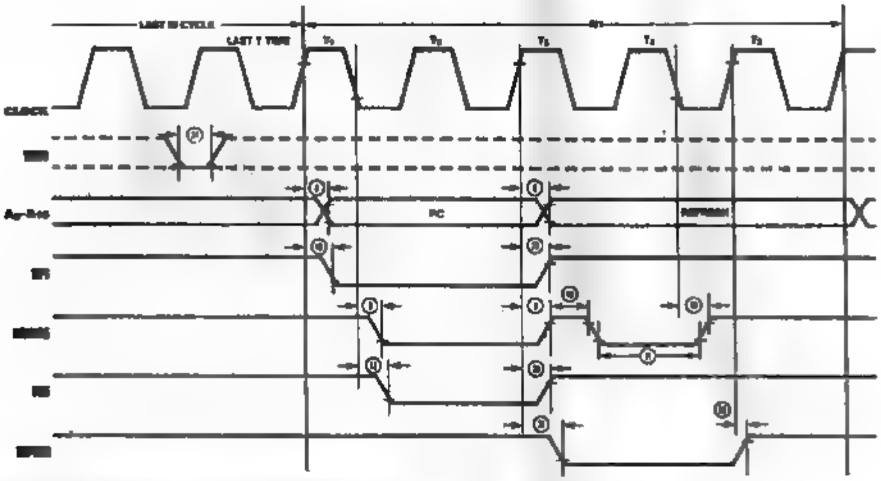
NOTE: 1) T_L = Last state of previous instruction.

2) Two Wait cycles automatically inserted by CPU(*).

Figure 8. Interrupt Request/Acknowledge Cycle

CPU Timing (Continued)

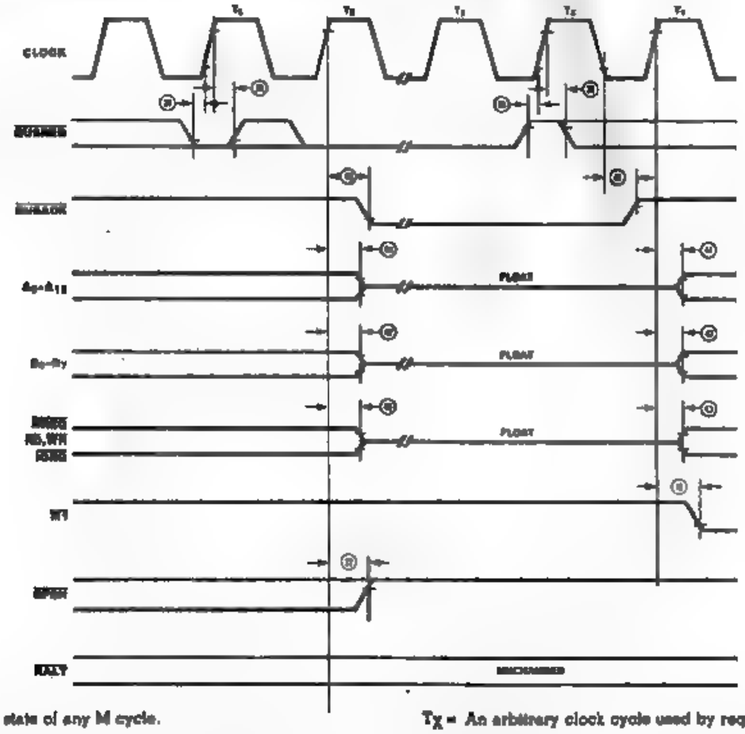
Non-Maskable Interrupt Request Cycle. $\overline{\text{NMI}}$ is sampled at the same time as the maskable interrupt input $\overline{\text{INT}}$ but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal instruction fetch except that data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



*Although $\overline{\text{NMI}}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{\text{NMI}}$'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 $\overline{\text{BUSREQ}}$ with the rising edge of the last clock period of any machine cycle (Figure 10). If $\overline{\text{BUSREQ}}$ is active, the CPU sets its address, data, and $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ 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.



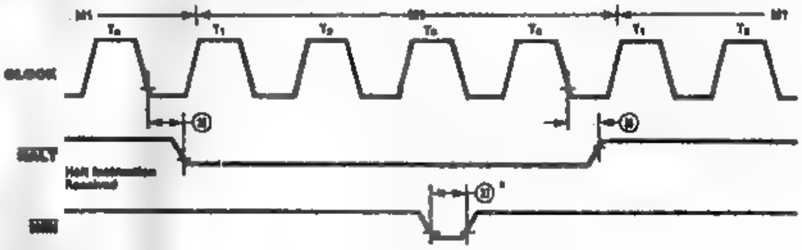
NOTE: T_L = Last state of any M cycle.

T_X = An arbitrary clock cycle used by requesting device.

Figure 10. Z-BUS Request/Acknowledge Cycle

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 ■ 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).

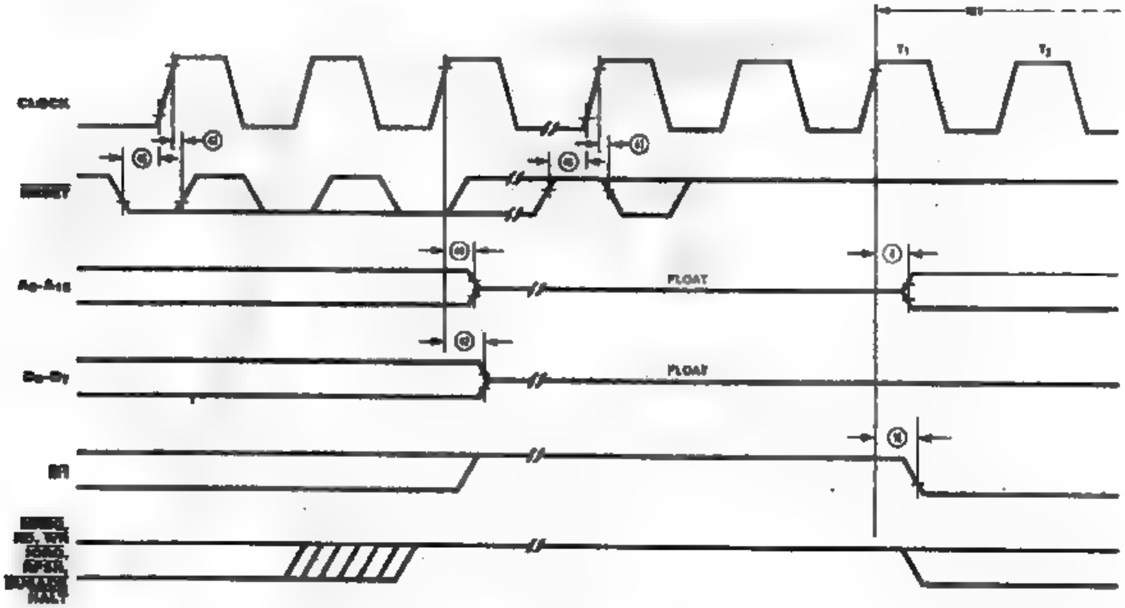


Figure 12. Reset Cycle

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 $\overline{\text{MREQ}}$ ↓ Delay	125*	—	65*	—	35*	—	20*	—
8	TdCf(MREQf)	Clock ↓ to $\overline{\text{MREQ}}$ ↓ Delay	—	100	—	85	—	70	—	60
9	TdCr(MREQr)	Clock ↑ to $\overline{\text{MREQ}}$ ↑ Delay	—	100	—	85	—	70	—	60
10	TwMREQh	$\overline{\text{MREQ}}$ Pulse Width (High)	170*		110*		65*		45*	
11	TwMREQl	$\overline{\text{MREQ}}$ Pulse Width (Low)	360*	—	220*	—	135*	—	100*	—
12	TdCf(MREQr)	Clock ↓ to $\overline{\text{MREQ}}$ ↑ Delay	—	100	—	85	—	70	—	60
13	TdCf(RDf)	Clock ↓ to $\overline{\text{RD}}$ ↓ Delay	—	130	—	95	—	80	—	70
14	TdCr(RDr)	Clock ↑ to $\overline{\text{RD}}$ ↑ Delay	—	100	—	85	—	70	—	60
15	TsD(Cr)	Data Setup Time to Clock ↑	50		35		30		30	
16	ThD(RDr)	Data Hold Time to $\overline{\text{RD}}$ ↑	—	0	—	▮	—	0	—	0
17	TsWAIT(Cf)	$\overline{\text{WAIT}}$ Setup Time to Clock ↓	70	—	70	—	60	—	50	—
18	ThWAIT(Cf)	$\overline{\text{WAIT}}$ Hold Time after Clock ↓	—	0	—	0	—	0	—	0
19	TdCr(M1f)	Clock ↑ to $\overline{\text{M1}}$ ↓ Delay	—	130	—	100	—	80	—	70
20	TdCr(M1r)	Clock ↑ to $\overline{\text{M1}}$ ↑ Delay		130		100		80	—	70
21	TdCr(RFSHf)	Clock ↑ to $\overline{\text{RFSH}}$ ↓ Delay	—	180	—	130	—	110	—	95
22	TdCr(RFSHr)	Clock ↑ to $\overline{\text{RFSH}}$ ↑ Delay	—	150	—	120	—	100	—	85
23	TdCf(RDr)	Clock ↓ to $\overline{\text{RD}}$ ↑ Delay	—	110	—	85	—	70	—	60
24	TdCr(RDf)	Clock ↑ to $\overline{\text{RD}}$ ↓ 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 $\overline{\text{IORQ}}$ ↓	320*	—	180*	—	110*	—	75*	—
27	TdCr(IORQf)	Clock ↑ to $\overline{\text{IORQ}}$ ↓ Delay	—	90	—	75	—	65	—	55
28	TdCf(IORQr)	Clock ↓ to $\overline{\text{IORQ}}$ ↑ Delay	—	110	—	65	—	70	—	60
29	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	190*	—	80*	—	25*	—	5*	—
30	TdCf(WRf)	Clock ↓ to $\overline{\text{WR}}$ ↓ Delay		90		80		70	—	60
31	TwWR	$\overline{\text{WR}}$ Pulse Width	360*	—	220*	—	135*	—	100*	—
32	TdCf(WRr)	Clock ↓ to $\overline{\text{WR}}$ ↑ Delay	—	100	—	80	—	70	—	60
33	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	20*	—	-10*	—	-55*	—	55*	—
34	TdCr(WRf)	Clock ↑ to $\overline{\text{WR}}$ ↓ Delay	—	60	—	65	—	60	—	55
35	TdWRr(D)	Data Stable from $\overline{\text{WR}}$ ↑	120*		60*		30*		15*	
36	TdCf(HALT)	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓	—	300	—	300	—	260	—	225
37	TwNMI	$\overline{\text{NMI}}$ Pulse Width	80	—	80	—	70	—	60*	—
38	TsBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ 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.

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)	$\overline{\text{BUSREQ}}$ Hold Time after Clock ↑	0	—	0	—	0	—	[illegible]	—
40	TdCr(BUSACKf)	Clock ↑ to $\overline{\text{BUSACK}}$ ↓ Delay		120		100		90	—	80
41	TdCf(BUSACKr)	Clock ↓ to $\overline{\text{BUSACK}}$ ↑ Delay	—	110	—	100	—	90	—	80
42	TdCr(Dz)	Clock ↑ to Data Float Delay	—	90	—	90	—	80	—	70
43	TdCr(CTz)	Clock ↑ to Control Output Float Delay ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$)	—	110	—	80	—	70	—	60
44	TdCr(Az)	Clock ↑ to Address Float Delay	—	110	—	90	—	80	—	70
45	TdCTr(A)	$\overline{\text{MREQ}}$ ↑, $\overline{\text{IORQ}}$ ↑, $\overline{\text{RD}}$ ↑, and $\overline{\text{WR}}$ ↑ to Address Hold Time	160*		80*		35*		20*	—
46	TsRESET(Cr)	$\overline{\text{RESET}}$ to Clock ↑ Setup Time	90	—	60	—	60	—	45	—
47	ThRESET(Cr)	$\overline{\text{RESET}}$ to Clock ↑ Hold Time	—	0	—	[illegible]	—	0	—	0
48	TsINTf(Cr)	$\overline{\text{INT}}$ to Clock ↑ Setup Time	80	—	80	—	70	—	55	—
49	ThINTr(Cr)	$\overline{\text{INT}}$ to Clock ↑ Hold Time	—	0	—	0	—	0	—	0
50	TdM1f(IORQf)	$\overline{\text{M1}}$ ↓ to $\overline{\text{IORQ}}$ ↓ Delay	920*		565*		365*		270*	—
51	TdCf(IORQf)	Clock ↓ to $\overline{\text{IORQ}}$ ↓ Delay	—	110	—	85	—	70	—	60
52	TdCf(IORQr)	Clock ↑ to $\overline{\text{IORQ}}$ ↑ 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 − [illegible]	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	$V_{OH} = 2.0$ V
$V_{IL} = 0.8$ V	$V_{OL} = 0.8$ V
$V_{IHC} = V_{CC} - 0.6$ V	FLOAT = ±0.5 V
$V_{ILC} = 0.45$ V	

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 ■ not implied. Exposure ■ absolute maximum rating conditions for extended periods may affect device reliability.

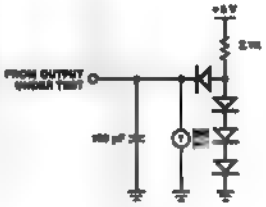
Standard Test Conditions

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, $+4.75\ V \le V_{CC} \le +5.25\ V$
- 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.

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 Z80A Z80B		 150[1] 200[2] 200	 mA mA mA	
I_{LI}	Input Leakage Current		10	µA	V_{IN} = ■ 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} ■ 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.

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.

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 ■ 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 ■ 28-pin plastic or ceramic DIP.

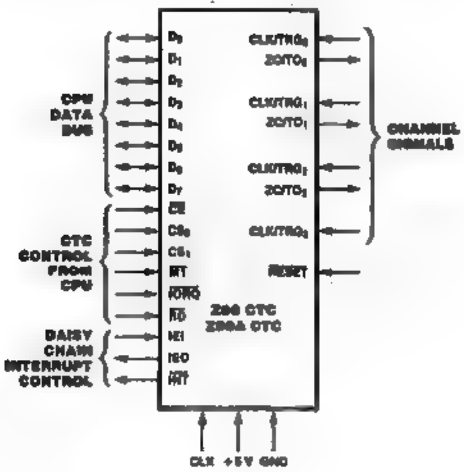


Figure 1. Pin Functions

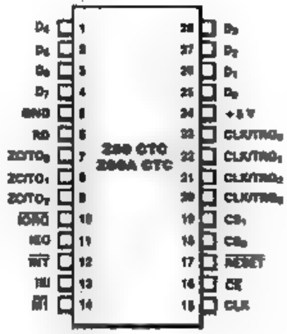


Figure 2. Pin Assignments

2041-0154, 0155

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 ■ 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 ■ 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 ■ automatically reset with the time constant value.

The timer mode determines time intervals as small as 4 μs (Z-80A) or 6.4 μs (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 ($\overline{\text{INT}}$), 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

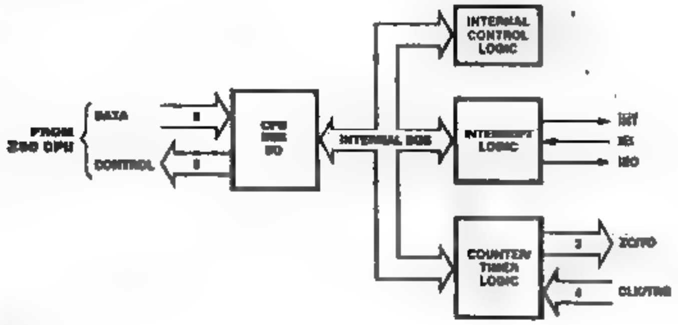


Figure 3. Functional Block Diagram

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 $\overline{\text{INT}}$ signal to the Z-80 CPU. When the Z-80 CPU responds with interrupt acknowledge ($\overline{\text{M1}}$ and $\overline{\text{IORQ}}$), 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 ■ 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 $\overline{\text{M1}}$ cycle. This ensures that all lower priority devices can decode the entire RETI instruction and reset properly.

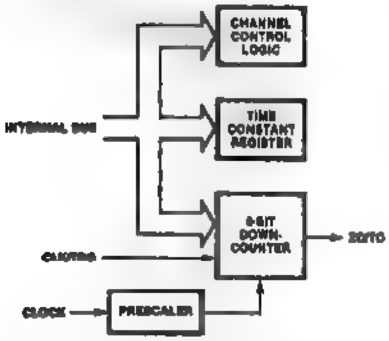


Figure 4. Counter/Timer Block Diagram

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 ($\overline{\text{INT}}$) from the interrupt logic.

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.

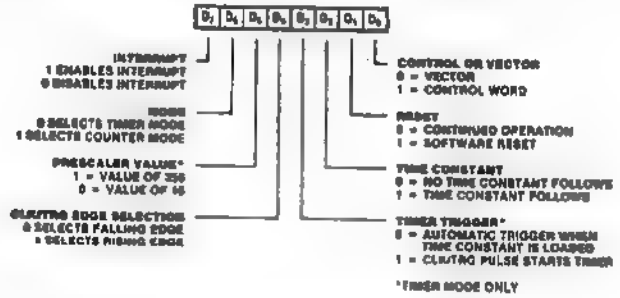


Figure 5. Channel Control Word

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 (ϕ)
- 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 μs 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.

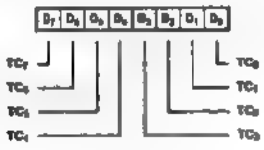


Figure 6. Time Constant Word

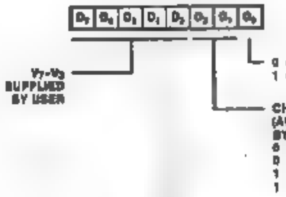


Figure 7. Interrupt Vector Word

Pin Description

$\overline{\text{CE}}$. *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.

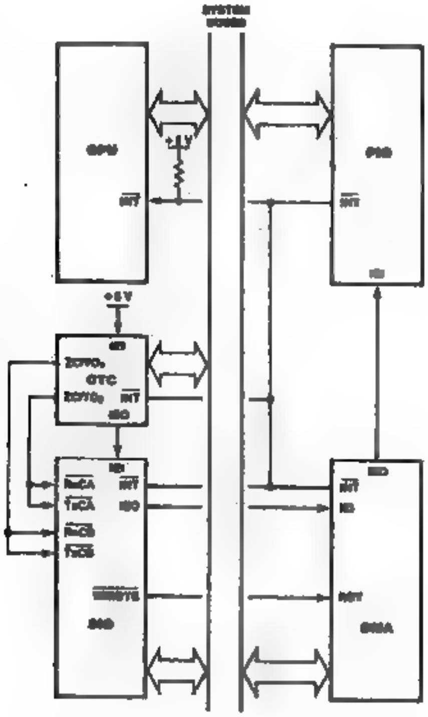


Figure 6. A Typical Z-80 Environment

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.

$\overline{\text{INT}}$. *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.

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

$\overline{\text{M1}}$. *Machine Cycle One* (input from CPU, active Low). When $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ 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 ($\overline{\text{INT}}$).

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

$\overline{\text{RESET}}$. *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 [illegible] (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.

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.

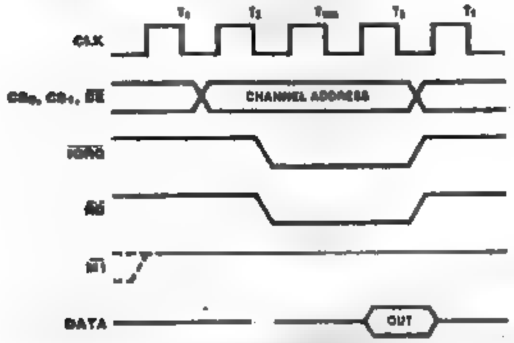


Figure 9. Read Cycle Timing

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.

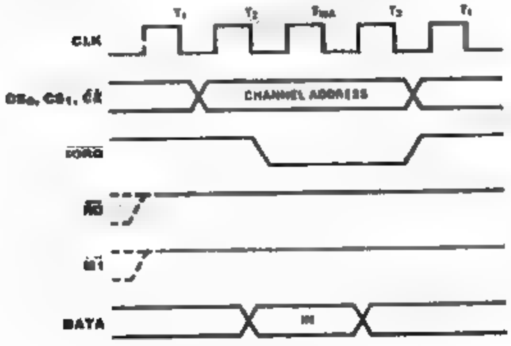


Figure 10. Write Cycle Timing

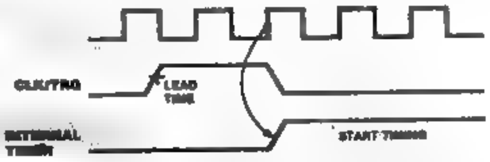


Figure 11. Timer Mode Timing

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.

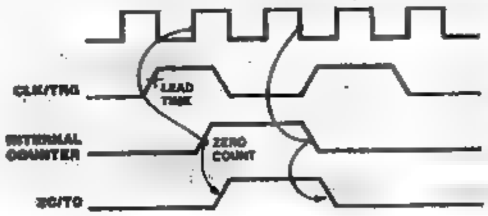


Figure 12. Counter Mode Timing

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.

Z80 CTC

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.*

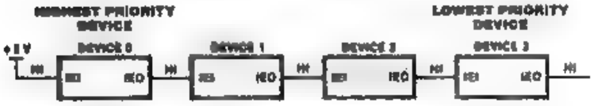


Figure 13. Daisy-Chain Interrupt Priorities

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.

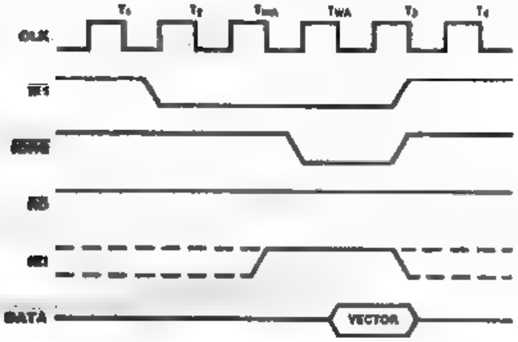


Figure 14. Interrupt Acknowledge Timing

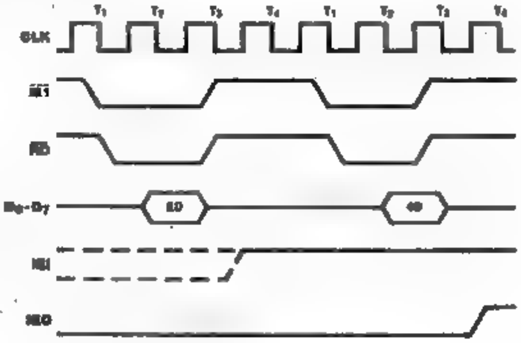


Figure 15. Return From Interrupt Timing

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

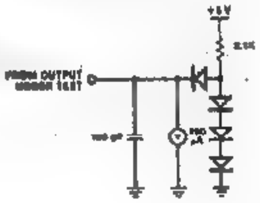
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating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

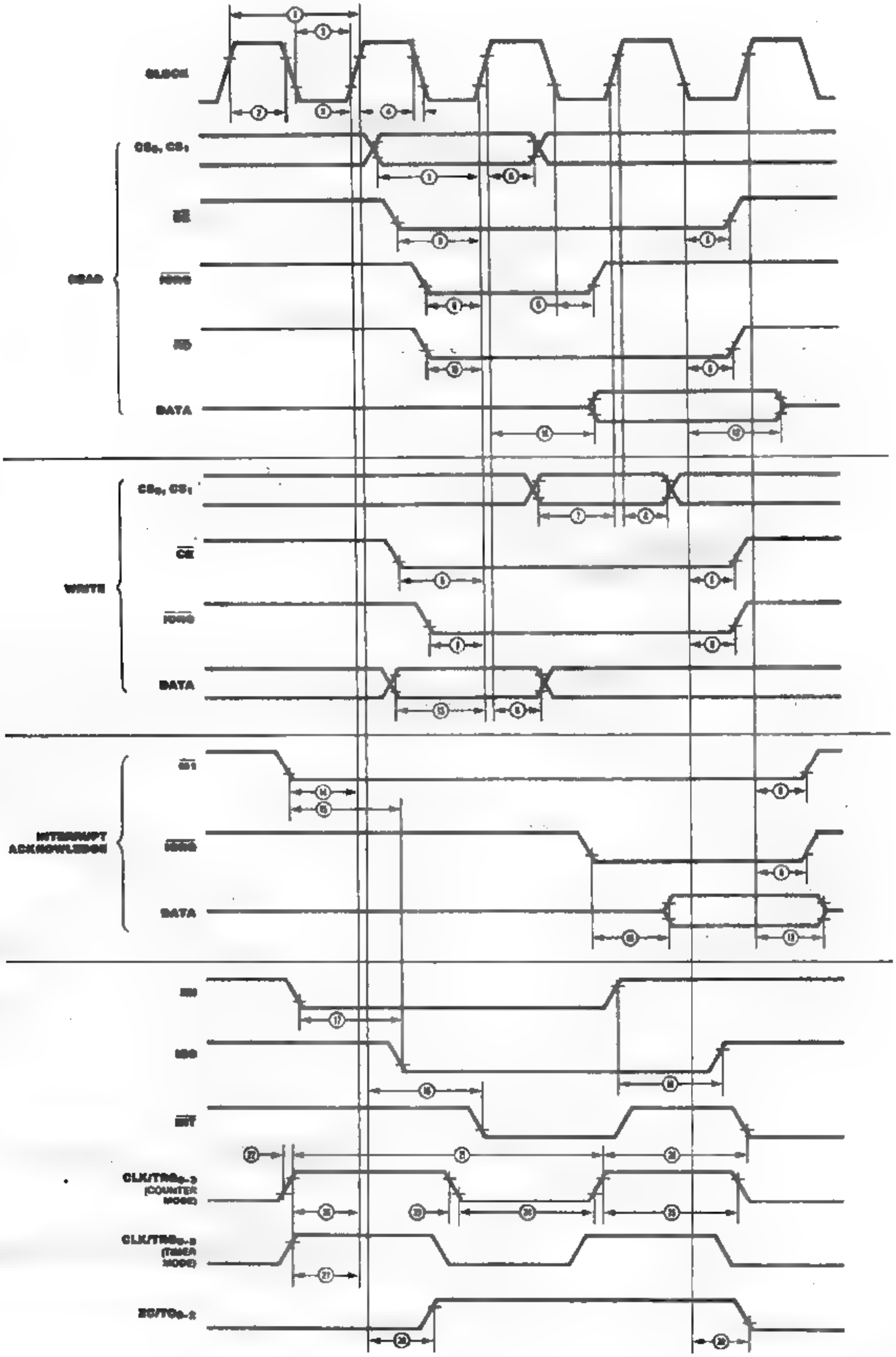


DC Characteristics

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


CLOCK
CS0, CS1
CE
READ
IORQ
RD
DATA
CS0, CS1
CE
WRITE
IORQ
DATA
M1
INTERRUPT ACKNOWLEDGE
IORQ
DATA
IEI
IEO
INT
CLK/TRG0-3 (COUNTER MODE)
CLK/TRG0-3 (TIMER MODE)
ZC/TO0-2

Number	Symbol	Parameter	Z-80 CTC Min (ns)	 Max (ns)	Z-80A CTC Min (ns)	 Max (ns)	Z-80B CTC Min (ns)	 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)	$\overline{\text{CE}}$ to Clock ↑ Setup Time	200		150		100		
9	TsIO(C)	$\overline{\text{IORQ}}$ ↓ to Clock ↑ Setup Time	250		115		70		
10	TsRD(C)	$\overline{\text{RD}}$ ↓ 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)	$\overline{\text{M1}}$ to Clock ↑ Setup Time	210		90		70		
15	TdM1(IEO)	$\overline{\text{M1}}$ ↓ to IEO ↓ Delay (Interrupt immediately preceding $\overline{\text{M1}}$)		300		190		130	[3]
16	TdIO(DOI)	$\overline{\text{IORQ}}$ ↓ 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 $\overline{\text{INT}}$ ↓ Delay	(TcC + 200)		(TcC + 140)		TcC + 120		[4]
20	TdCLK(INT)	CLK/TRG ↑ to $\overline{\text{INT}}$ ↓ tsCTR(C) satisfied tsCTR(C) not satisfied	 (19) + (26) (1) + (19) + (26)		 (19) + (26) (1) + (19) + (26)		 (19) + (26) (1) + (19) + (26)		 [5] [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] $\overline{\text{RESET}}$ 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.

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.

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.

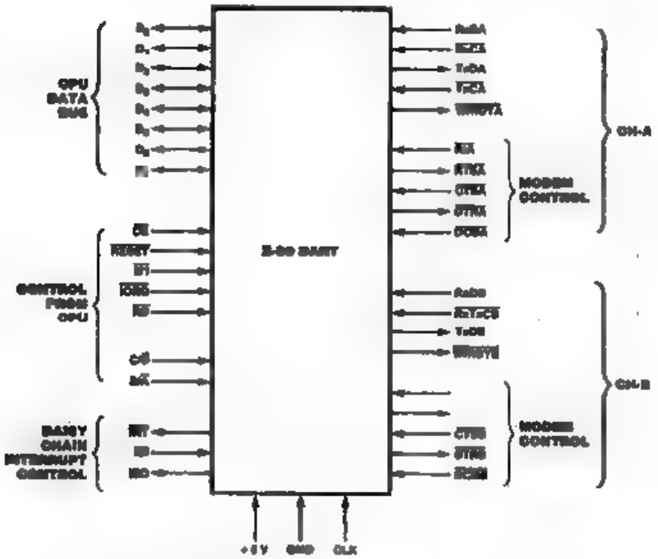


Figure 1. Z80 DART Pin Functions

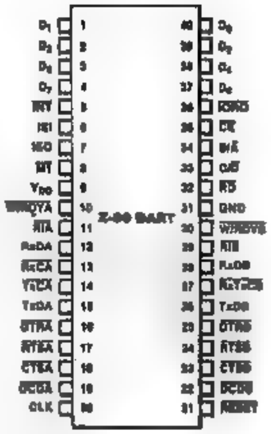


Figure 2. Pin Assignments

Pin Description

B/$\overline{\text{A}}$. *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/$\overline{\text{D}}$. *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.

$\overline{\text{CE}}$. *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.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$. *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.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$. *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.

$\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$. *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 ■ being serviced by its CPU interrupt service routine.

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

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

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

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

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

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

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

$\overline{\text{RIA}}$, $\overline{\text{RIB}}$. *Ring Indicator* (inputs, Active Low). These inputs are similar to $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$. 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.

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

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$. *Transmitter Clocks* (inputs). TxD changes on the falling edge of $\overline{\text{TxC}}$. 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).

$\overline{\text{W/RDYA}}$, $\overline{\text{W/RDYB}}$. *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.

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.

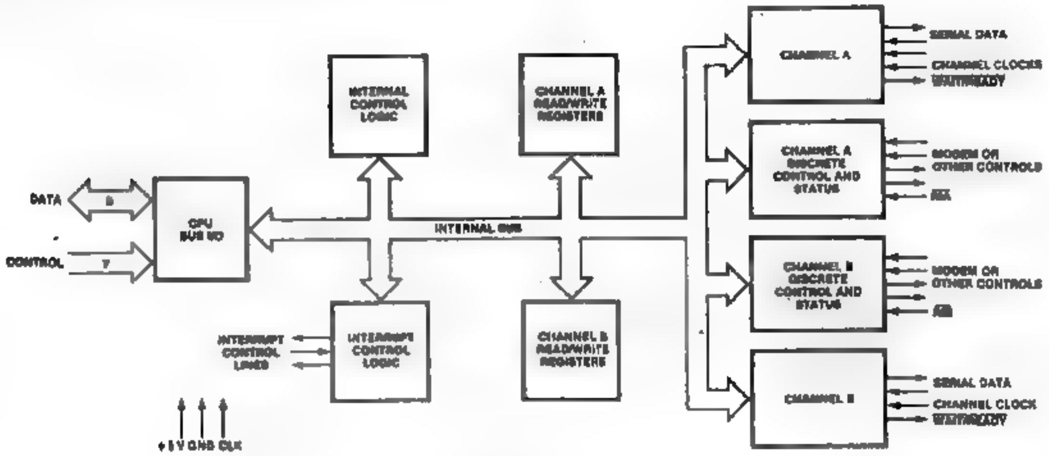


Figure 3. Block Diagram

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.

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.

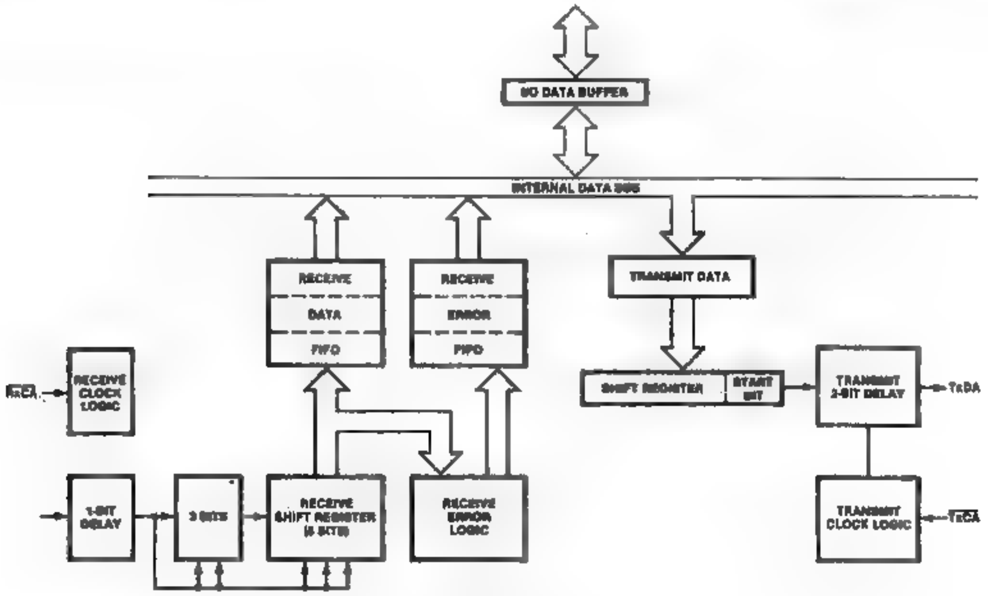


Figure 4. Data Path

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 ($\overline{MI}$ 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 $\overline{MI}$ 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.

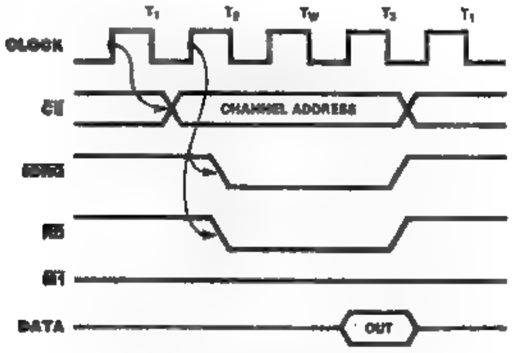


Figure 5a. Read Cycle

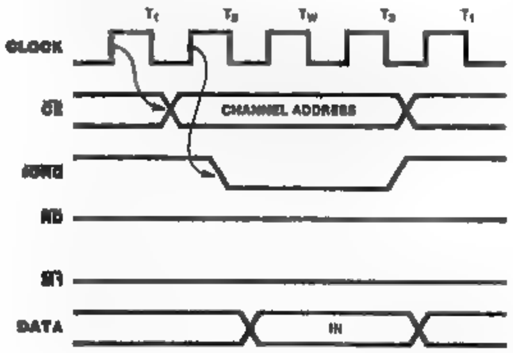


Figure 5b. Write Cycle

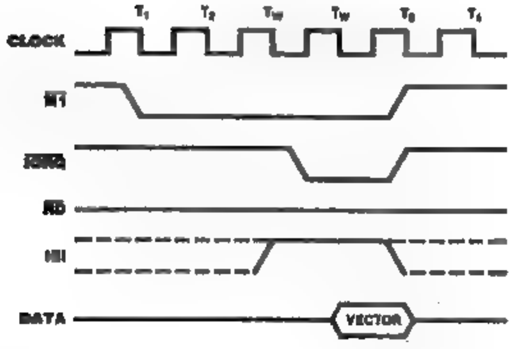


Figure 5c. Interrupt Acknowledge Cycle

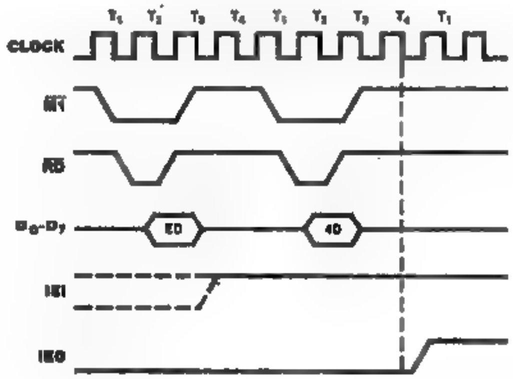


Figure 5d. Return from Interrupt Cycle

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.

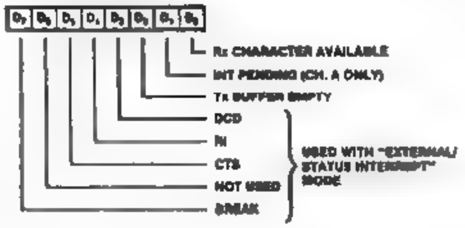
The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

Read Register Functions

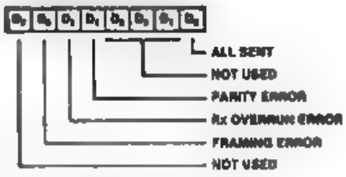
RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)

Z-80 DART Read and Write Registers

READ REGISTER 0

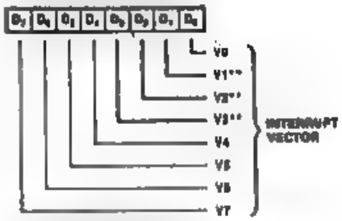


READ REGISTER 1*



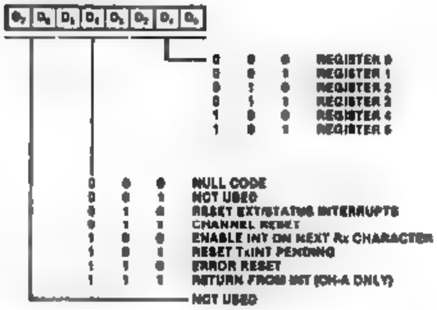
*Used With Special Receive Condition Mode

READ REGISTER 2

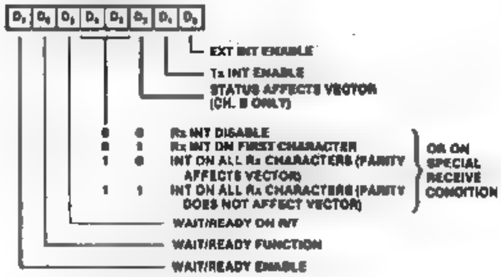


**Variable If "Status Affects Vector" Is Programmed

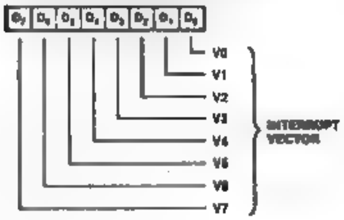
WRITE REGISTER 0



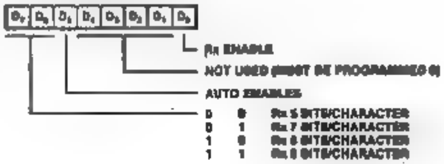
WRITE REGISTER 1


D7 D6 D5 D4 D3 D2 D1 D0
EXT INT ENABLE
Tx INT ENABLE
STATUS AFFECTS VECTOR (CH. B ONLY)
0 0 Rx INT DISABLE
0 1 Rx INT ON FIRST CHARACTER
1 0 INT ON ALL Rx CHARACTERS (PARITY AFFECTS VECTOR)
1 1 INT ON ALL Rx CHARACTERS (PARITY DOES NOT AFFECT VECTOR)
OR ON SPECIAL RECEIVE CONDITION
WAIT/READY ON R/T
WAIT/READY FUNCTION
WAIT/READY ENABLE

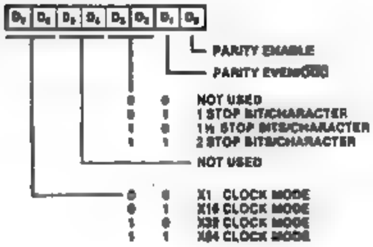
WRITE REGISTER 2 (CHANNEL B ONLY)



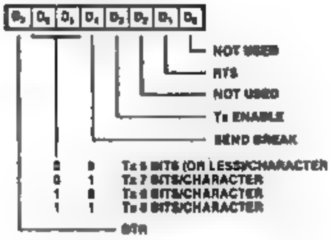
WRITE REGISTER 3



WRITE REGISTER 4



WRITE REGISTER 5



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

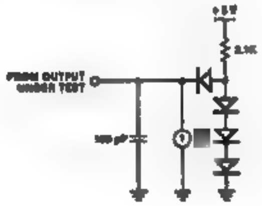
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating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

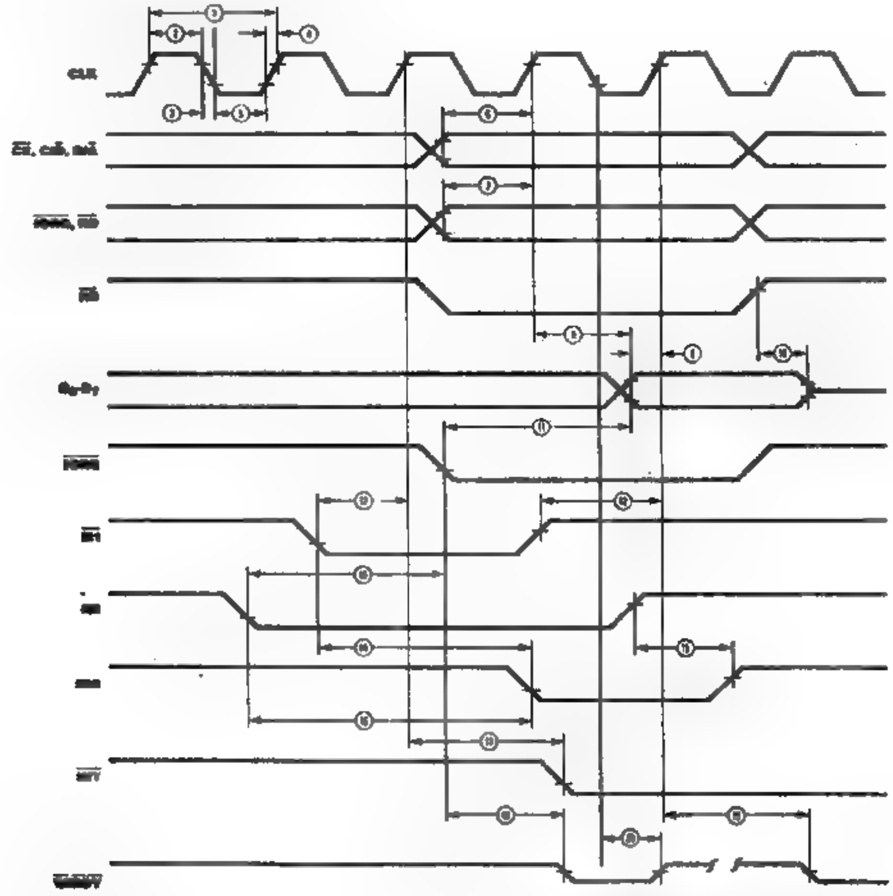


DC Characteristics

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}-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
$I_{L(RI)}$	$\overline{\text{RI}}$ 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%

AC Electrical Characteristics

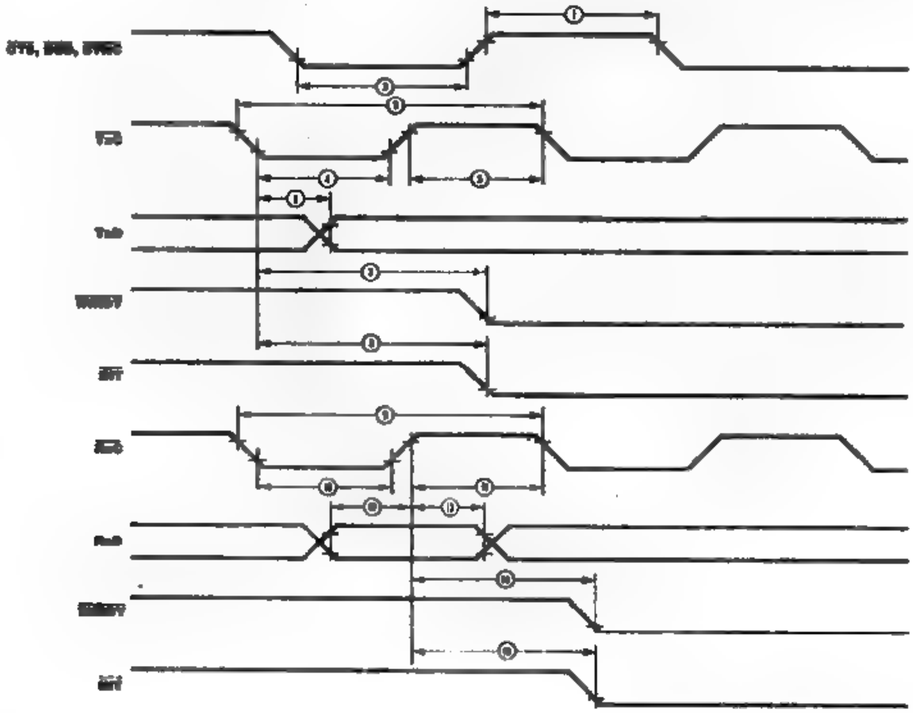


Number	Symbol	Parameter	Z-80 DART Min	Z-80 DART Max	Z-80A DART Min	Z-80A DART Max	Z-80B DART*† Min	Z-80B DART*† Max
1	TcC	Clock Cycle Time	400	4000	250	4000	165	4000
2	TwCh	Clock Width (High)	170	2000	105	2000	70	2000
3	TfC	Clock Fall Time		30		30		15
4	TrC	Clock Rise Time		30		30		15
5	TwCl	Clock Width (Low)	170	2000	105	2000	70	2000
6	TsAD(C)	$\overline{\text{CE}}$, C/$\overline{\text{D}}$, B/$\overline{\text{A}}$ to Clock ↑ Setup Time	160		145		60	
7	TsCS(C)	$\overline{\text{IORQ}}$, $\overline{\text{RD}}$ 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)	$\overline{\text{RD}}$ ↑ to Data Out Float Delay		230		110		90
11	TdIO(DOI)	$\overline{\text{IORQ}}$ ↓ to Data Out Delay (INTACK Cycle)		340		160		100
12	TsM1(C)	$\overline{\text{M1}}$ to Clock ↑ Setup Time	210		90		75	
13	TsIEI(IO)	IEI to $\overline{\text{IORQ}}$ ↓ Setup Time (INTACK Cycle)	200		140		120	
14	TdM1(IEO)	$\overline{\text{M1}}$ ↓ 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 $\overline{\text{INT}}$ ↓ Delay		200		200		150
18	TdIO(W/RWf)	$\overline{\text{IORQ}}$ ↓ or $\overline{\text{CE}}$ ↓ 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.

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	$\overline{\text{TxC}}$ Cycle Time	400	∞	400	∞	330	∞	2
4	TwTxCl	$\overline{\text{TxC}}$ Width (Low)	180	∞	180	∞	100	∞	2
5	TwTxCh	$\overline{\text{TxC}}$ Width (High)	180	∞	180	∞	100	∞	2
6	TdTxC(TxD)	$\overline{\text{TxC}}$ ↓ to TxD Delay		400		300		220	2
7	TdTxC(W/RRl)	$\overline{\text{TxC}}$ ↓ to W/RDY ↓ Delay (Ready Mode)	5	9	5	9	5	9	3
8	TdTxC(INT)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{INT}}$ ↓ Delay	5	9	5	9	5	9	3
9	TcRxC	$\overline{\text{RxC}}$ Cycle Time	400	∞	400	∞	330	∞	2
10	TwRxCl	$\overline{\text{RxC}}$ Width (Low)	180	∞	180	∞	100	∞	2
11	TwRxCh	$\overline{\text{RxC}}$ Width (High)	180	∞	180	∞	100	∞	2
12	TsRxD(RxC)	RxD to $\overline{\text{RxC}}$ ↑ Setup Time (x1 Mode)	0		0		0		2
13	ThRxD(RxC)	RxD Hold Time (x1 Mode)	140		140		100		2
14	TdRxC(W/RRl)	$\overline{\text{RxC}}$ ↑ to W/RDY ↓ Delay (Ready Mode)	10	13	10	13	10	13	3
15	TdRxC(INT)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{INT}}$ ↓ 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. $\overline{\text{RESET}}$ 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.

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.

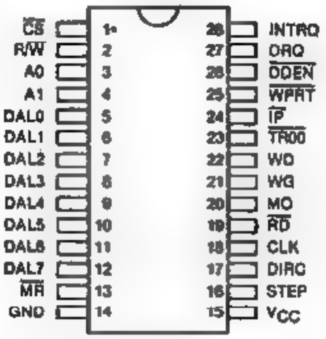
WESTERN DIGITAL
CORPORATION

PRELIMINARY

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 (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.

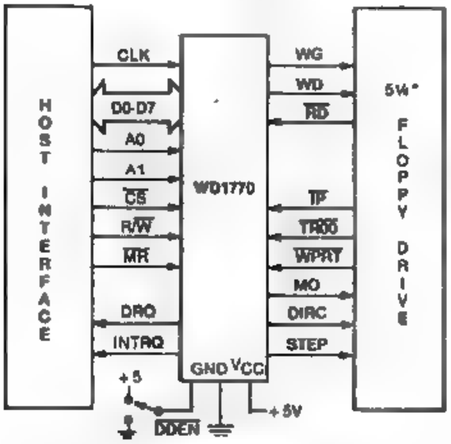
June, 1983

Reproduced with permission from
Western Digital Corporation

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	$\overline{\text{CHIP SELECT}}$	$\overline{\text{CS}}$	A logic low on this input selects the chip and enable Host communication with the device.
2	READ/$\overline{\text{WRITE}}$	R/$\overline{\text{W}}$	A logic high on this input controls the placement of data on the $\overline{\text{D0-D7}}$ 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: $\overline{\text{CS}}$ A1 A0 R/$\overline{\text{W}}$ = 1 R/$\overline{\text{W}}$ = 0 0 0 0 Status Reg Command Reg 0 [illegible] 1 Track Reg Track Reg 0 1 0 Sector Reg Sector Reg 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 $\overline{\text{CS}}$ and R/$\overline{\text{W}}$. Each line will drive one TTL load.
13	$\overline{\text{MASTER RESET}}$	$\overline{\text{MR}}$	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/$\overline{\text{W}}$ 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	$\overline{\text{READ DATA}}$	$\overline{\text{RD}}$	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	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This active low input informs the WD1770 that the drive's R/$\overline{\text{W}}$ heads are positioned over Track zero (internal pull-up).
24	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	This active low input informs the WD1770 when the physical index hole has been encountered on the diskette (internal pull-up).
25	$\overline{\text{WRITE PROTECT}}$	$\overline{\text{WPRT}}$	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	$\overline{\text{DOUBLE DENSITY ENABLE}}$	$\overline{\text{DDEN}}$	This input pin selects either single (FM) or double (MFM) density. When $\overline{\text{DDEN}}$ = 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.

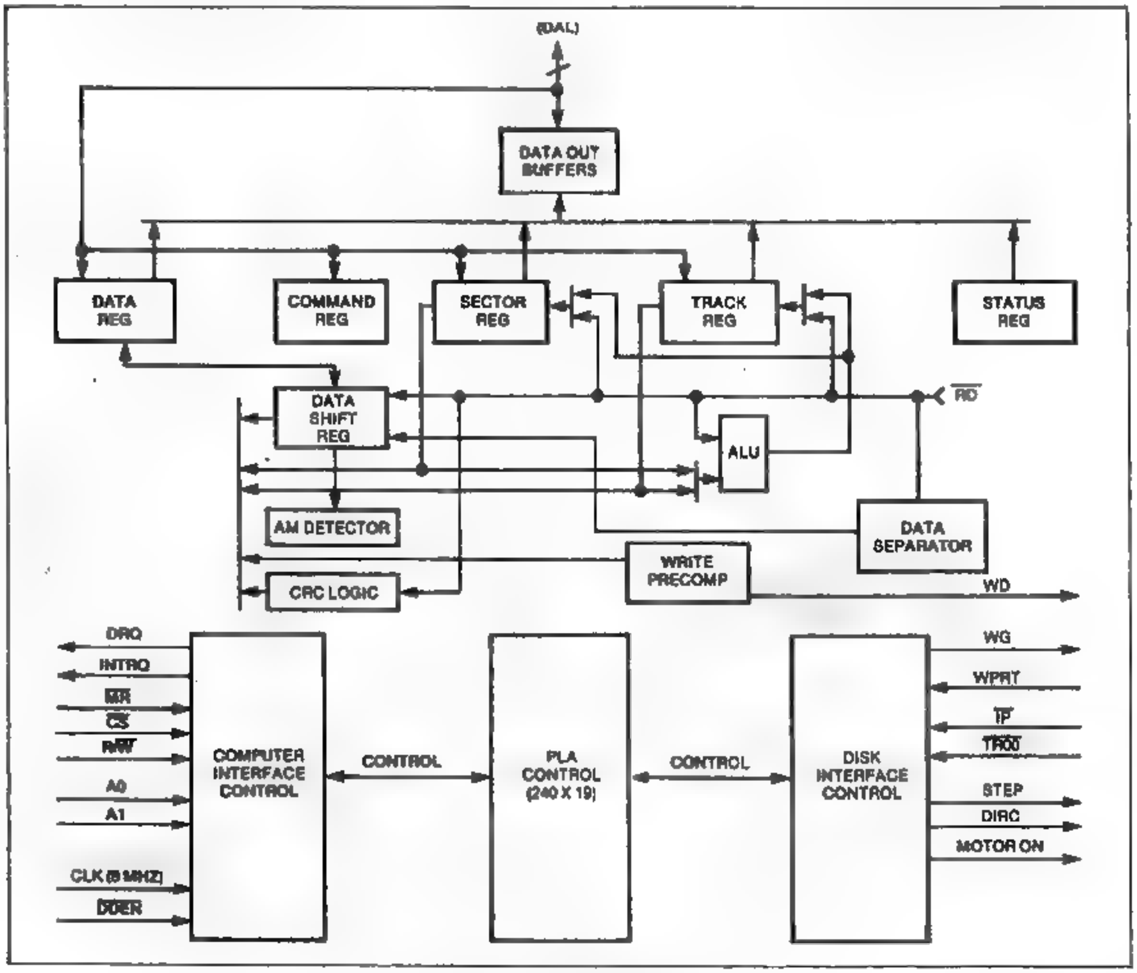
Command Register (CR) — This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

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 $\overline{\text{DDEN}}$ (Pin 26). When $\overline{\text{DDEN}}$ = 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, $\overline{\text{DDEN}}$ should be placed to logical "1." For MFM formats, $\overline{\text{DDEN}}$ 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

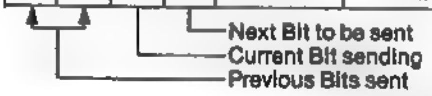
When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the device before the Write Gate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

Writing is inhibited when the $\overline{\text{Write Protect}}$ 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

		BITS							
TYPE	COMMAND	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 4µs (MFM) or [illegible] µs (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 24µs 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. [illegible] 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.

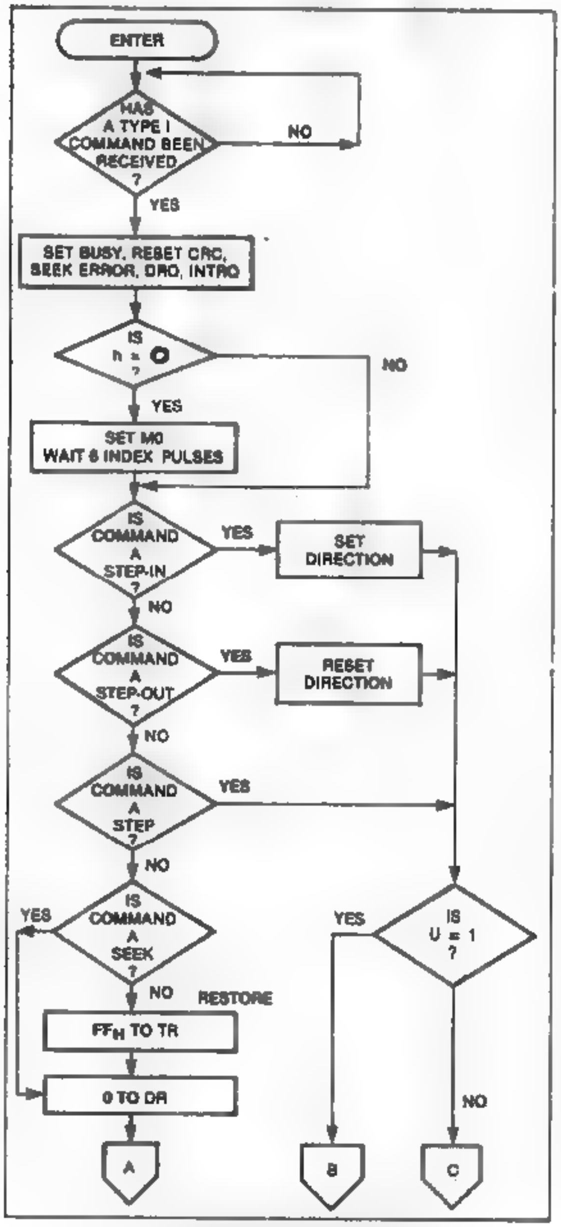
The WD1770 must find an ID field with correct track number and correct CRC within 5 revolutions of the media, otherwise the seek error is set and an INTRQ is generated. If V = 0, no verification is performed.

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 [illegible] 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

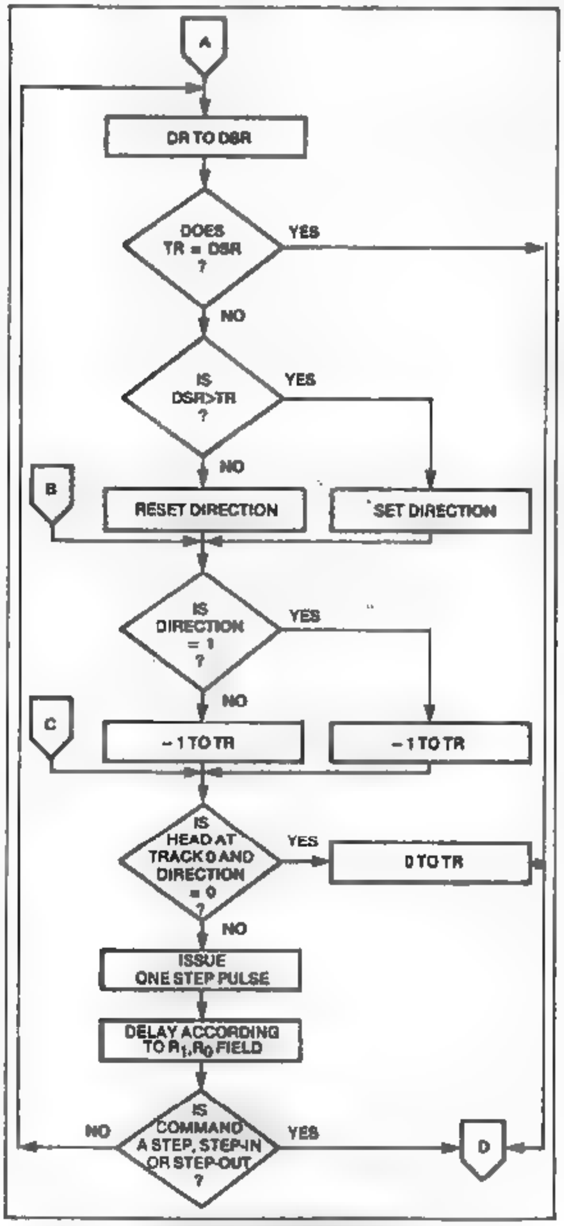
On line will go back to a logic 0. If a command is issued while Motor On is high, the command will execute immediately, defeating the 6 revolution start up. This feature allows consecutive Read or Write commands without waiting for motor start up each time; the WD1770 assumes the spindle motor is up 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



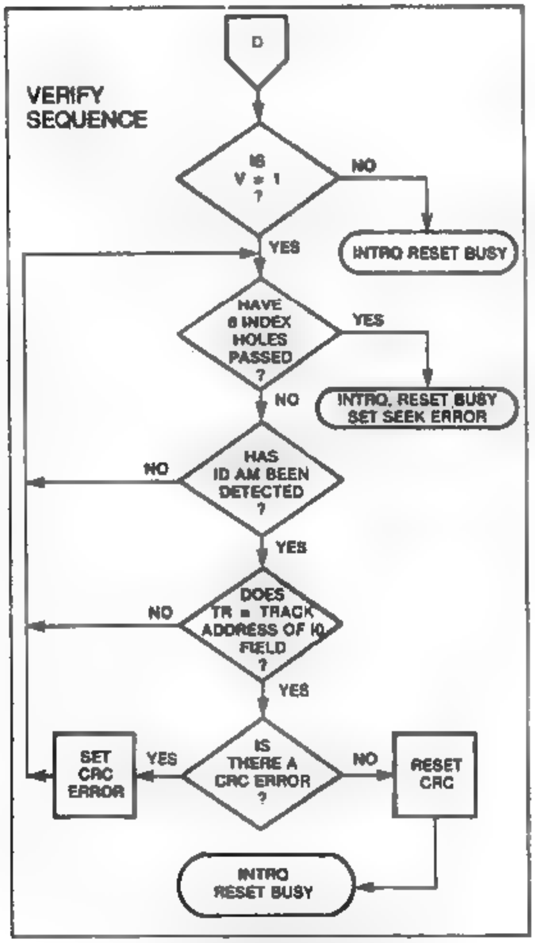
TYPE I COMMAND FLOW

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

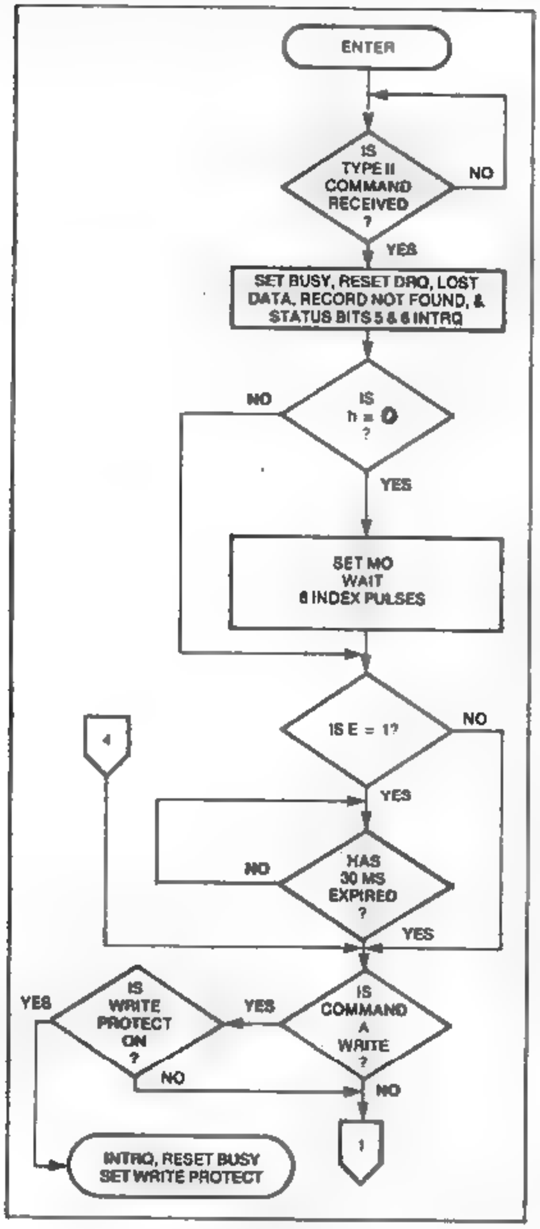
Upon receipt of this command, the WD1770 issues one stepping pulse in the direction towards track 0. If the U flag is on, the Track Register is decremented by one. After delay determined by the r_1,r_0 field, a verification takes place if the V flag is on. The h bit allows the Motor On option at the 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


ENTER
IS TYPE II COMMAND RECEIVED ?
NO
YES
SET BUSY, RESET DRQ, LOST DATA, RECORD NOT FOUND, & STATUS BITS 5 & 6 INTRQ
NO
IS h = 0 ?
YES
SET MO WAIT 6 INDEX PULSES
IS E = 1?
NO
YES
4
HAS 30 MS EXPIRED ?
NO
YES
YES
IS WRITE PROTECT ON ?
YES
IS COMMAND A WRITE ?
NO
NO
INTRQ, RESET BUSY SET WRITE PROTECT
1

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).

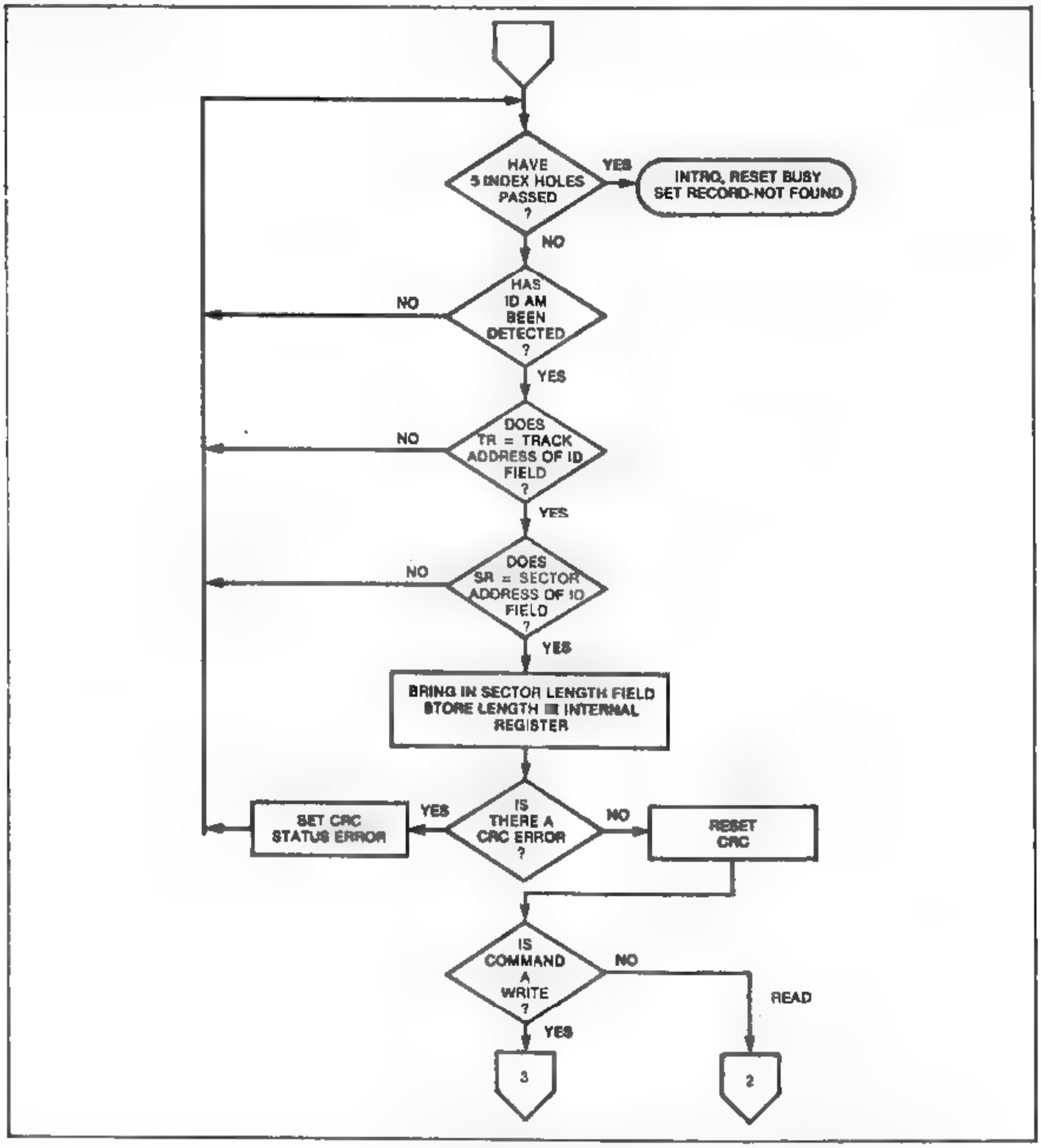
At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown:

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


HAVE 5 INDEX HOLES PASSED ?
YES
INTRQ, RESET BUSY SET RECORD-NOT FOUND
NO
HAS ID AM BEEN DETECTED ?
NO
YES
DOES TR = TRACK ADDRESS OF ID FIELD ?
NO
YES
DOES SR = SECTOR ADDRESS OF ID FIELD ?
NO
YES
BRING IN SECTOR LENGTH FIELD STORE LENGTH IN INTERNAL REGISTER
IS THERE A CRC ERROR ?
YES
SET CRC STATUS ERROR
NO
RESET CRC
IS COMMAND A WRITE ?
NO
READ
YES
3
2

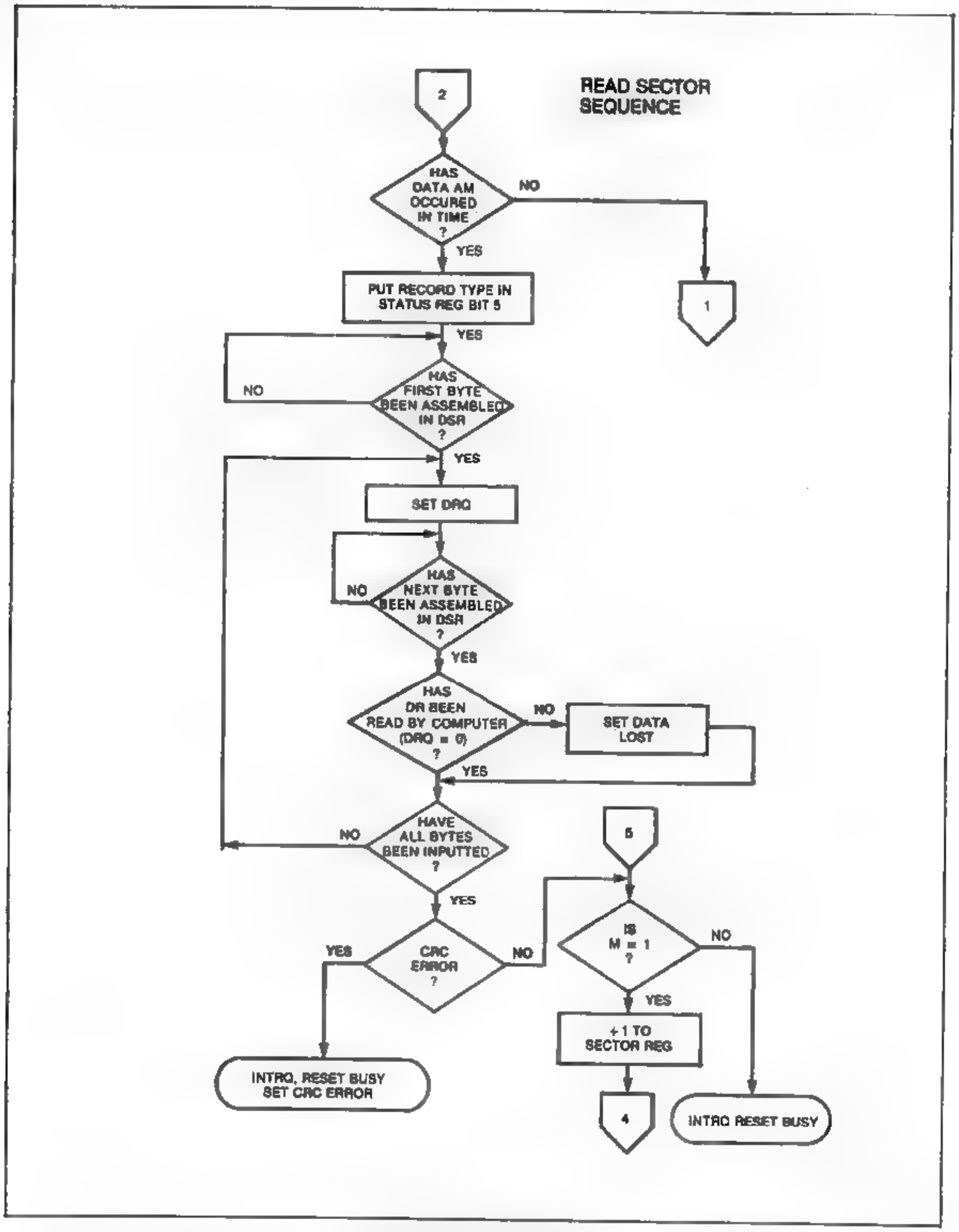
TYPE II COMMAND

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

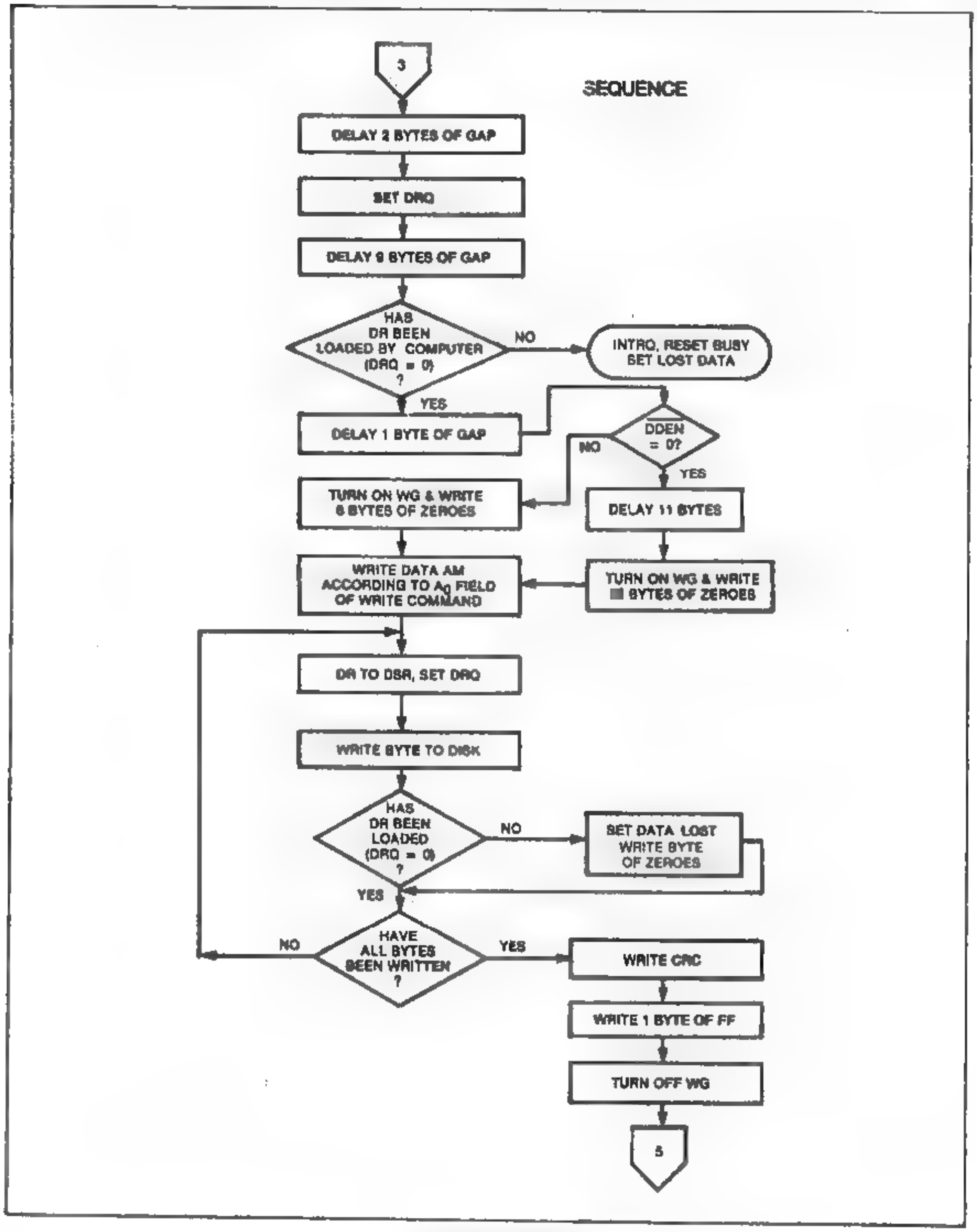
The WD1770 then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit

Reproduced with permission from
Western Digital Corporation


READ SECTOR SEQUENCE
2
HAS DATA AM OCCURED IN TIME ?
NO
1
YES
PUT RECORD TYPE IN STATUS REG BIT 5
YES
NO
HAS FIRST BYTE BEEN ASSEMBLED IN DSR ?
YES
SET DRQ
NO
HAS NEXT BYTE BEEN ASSEMBLED IN DSR ?
YES
HAS DR BEEN READ BY COMPUTER (DRQ = 0) ?
NO
SET DATA LOST
YES
NO
HAVE ALL BYTES BEEN INPUTTED ?
YES
5
YES
CRC ERROR ?
NO
IS M = 1 ?
NO
YES
+1 TO SECTOR REG
INTRQ, RESET BUSY SET CRC ERROR
4
INTRQ RESET BUSY

TYPE II COMMAND

Reproduced with permission from
Western Digital Corporation


3
SEQUENCE
DELAY 2 BYTES OF GAP
SET DRQ
DELAY 9 BYTES OF GAP
HAS DR BEEN LOADED BY COMPUTER (DRQ = 0) ?
NO
INTRQ, RESET BUSY SET LOST DATA
YES
DELAY 1 BYTE OF GAP
$\overline{\text{DDEN}}$ = 0?
NO
YES
TURN ON WG & WRITE 6 BYTES OF ZEROES
DELAY 11 BYTES
WRITE DATA AM ACCORDING TO A_0 FIELD OF WRITE COMMAND
TURN ON WG & WRITE ■ BYTES OF ZEROES
DR TO DSR, SET DRQ
WRITE BYTE TO DISK
HAS DR BEEN LOADED (DRQ = 0) ?
NO
SET DATA LOST WRITE BYTE OF ZEROES
YES
HAVE ALL BYTES BEEN WRITTEN ?
NO
YES
WRITE CRC
WRITE 1 BYTE OF FF
TURN OFF WG
5

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.

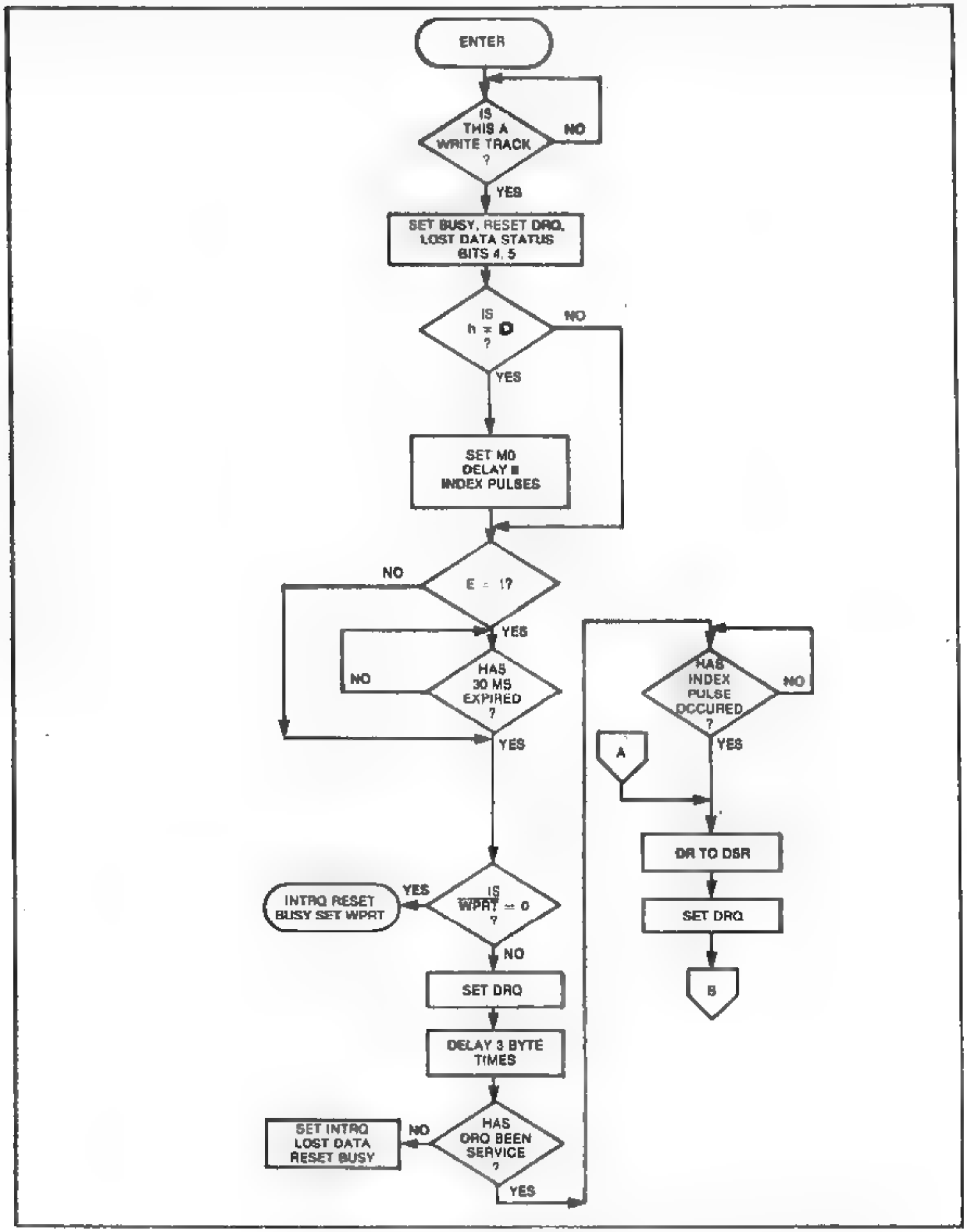
This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the WD1770 detects a data pattern of F5 through FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation.

DATA PATTERN IN DR (HEX)	IN FM ($\overline{\text{DDEN}}$ = 1)	IN MFM ($\overline{\text{DDEN}}$ = 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.

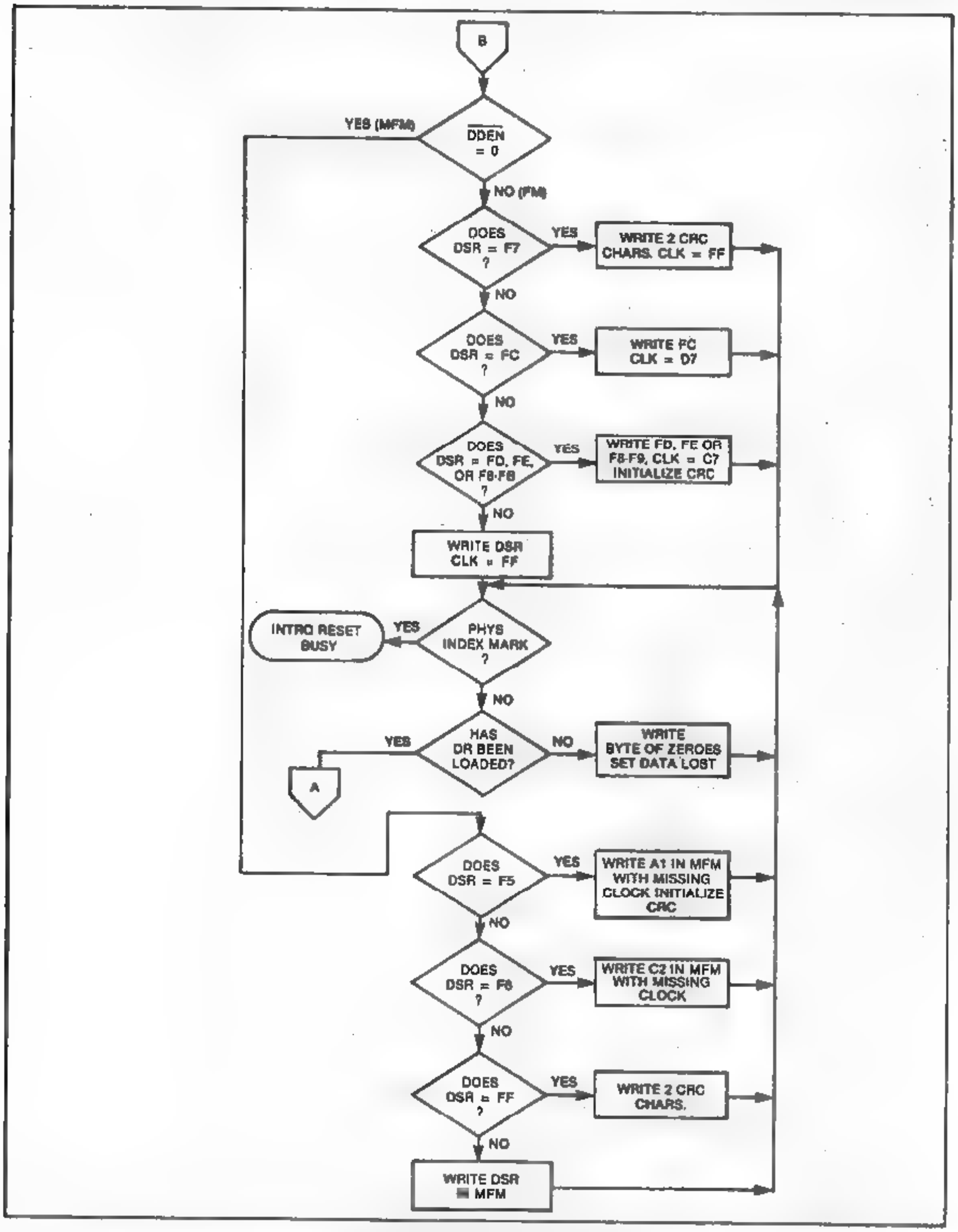
**Missing clock transition between bits 3 and 4.

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 [illegible] 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.

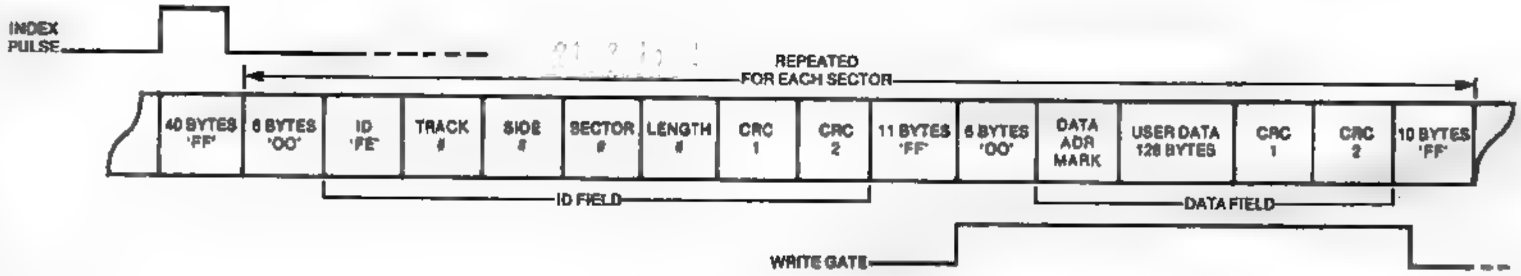
NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00)
6	00
1	FE (ID Address Mark)
1	Track Number
1	Side Number (00 or 01)
1	Sector Number (1 thru 1A)
1	00 (Sector Length)
1	F7 (2 CRC's written)
11	FF (or 00)
6	00
1	FB (Data Address Mark)
128	Data (IBM uses E5)
1	F7 (2 CRC's written)
10	FF (or 00)
369**	FF (or 00)

*Write bracketed field 16 times.
**Continue writing until WD1770 interrupts out. Approx. 369 bytes.

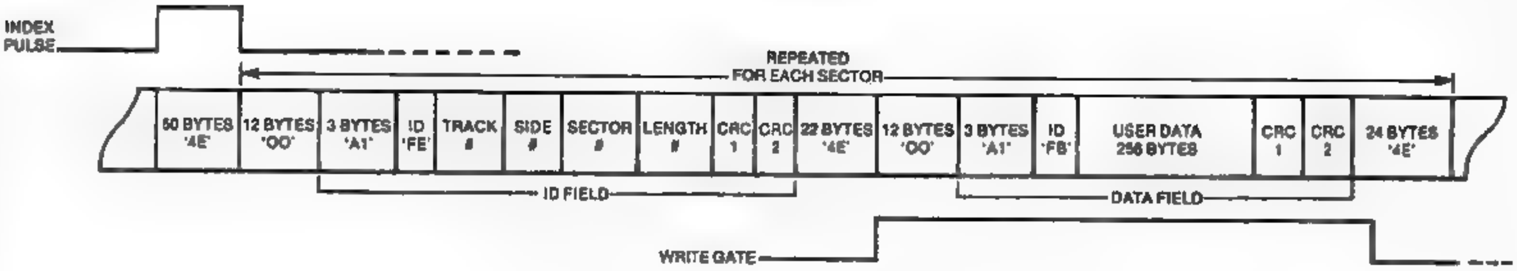
256 BYTES/SECTOR

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



SINGLE DENSITY FORMAT



DOUBLE DENSITY FORMAT

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 * 4 bytes 00	24 bytes 4E 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	$I_O = -100 \mu A$
V_{OL}	Output Low Voltage		0.40	V	$I_O = 1.6 mA$
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.

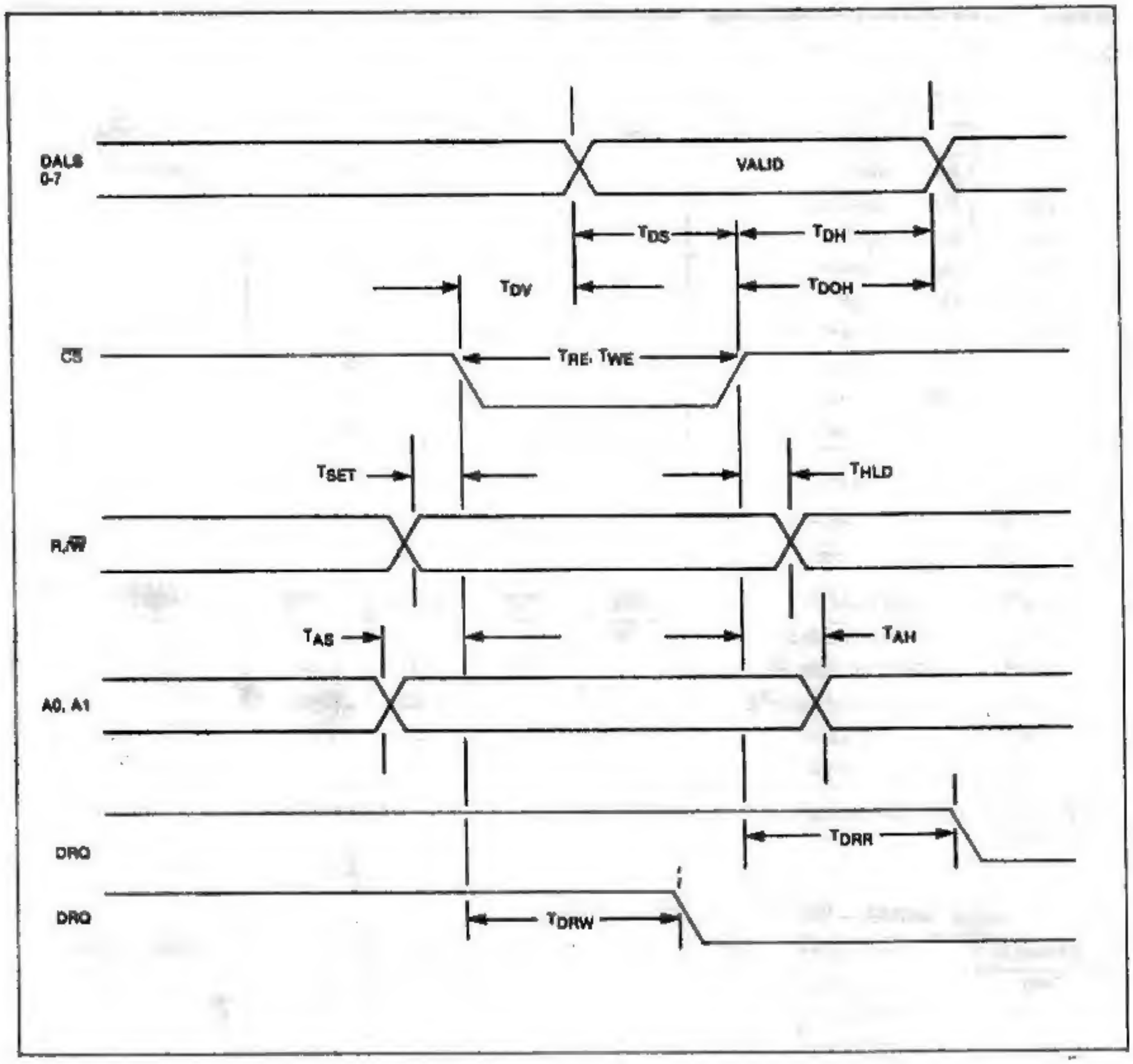
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TRE	RE Pulse Width of $\overline{CS}$	150			nsec	C_L = 50 pf
TDRR	DRQ Reset from $\overline{RE}$		25	100	nsec	
TIRR	INTRQ Reset from $\overline{RE}$			8000	nsec	
TDV	Data Valid from $\overline{RE}$		100	200	nsec	C_L = 50 pf
TDOH	Data Hold from $\overline{RE}$	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.

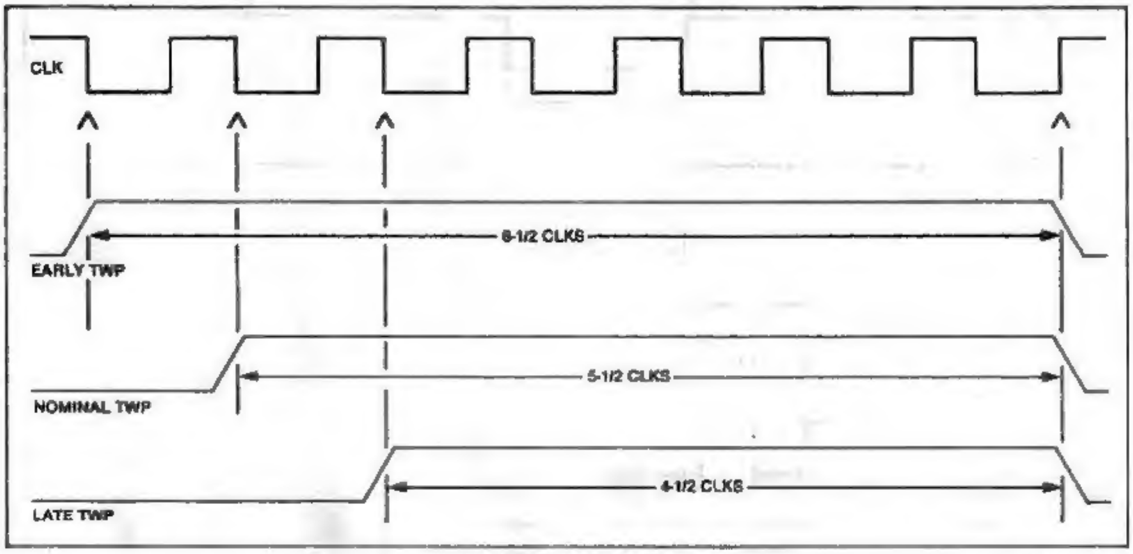
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TAS	Setup ADDR to $\overline{CS}$	50			nsec	
TSET	Setup R/$\overline{W}$ to $\overline{CS}$	0			nsec	
TAH	Hold ADDR from $\overline{CS}$	20			nsec	
THLD	Hold R/$\overline{W}$ from $\overline{CS}$	0			nsec	
TWE	$\overline{WE}$ Pulse Width	150			nsec	
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 2		μsec μsec	FM MFM
Tbc	Write Data Cycle Time		4,6,8		μsec	
Twf	Write Gate off from WD		4 2		μsec μsec	FM MFM
Twp	Write Data Pulse Width		820 690 570 1380		nsec nsec nsec nsec	Early MFM Nominal MFM Late MFM FM

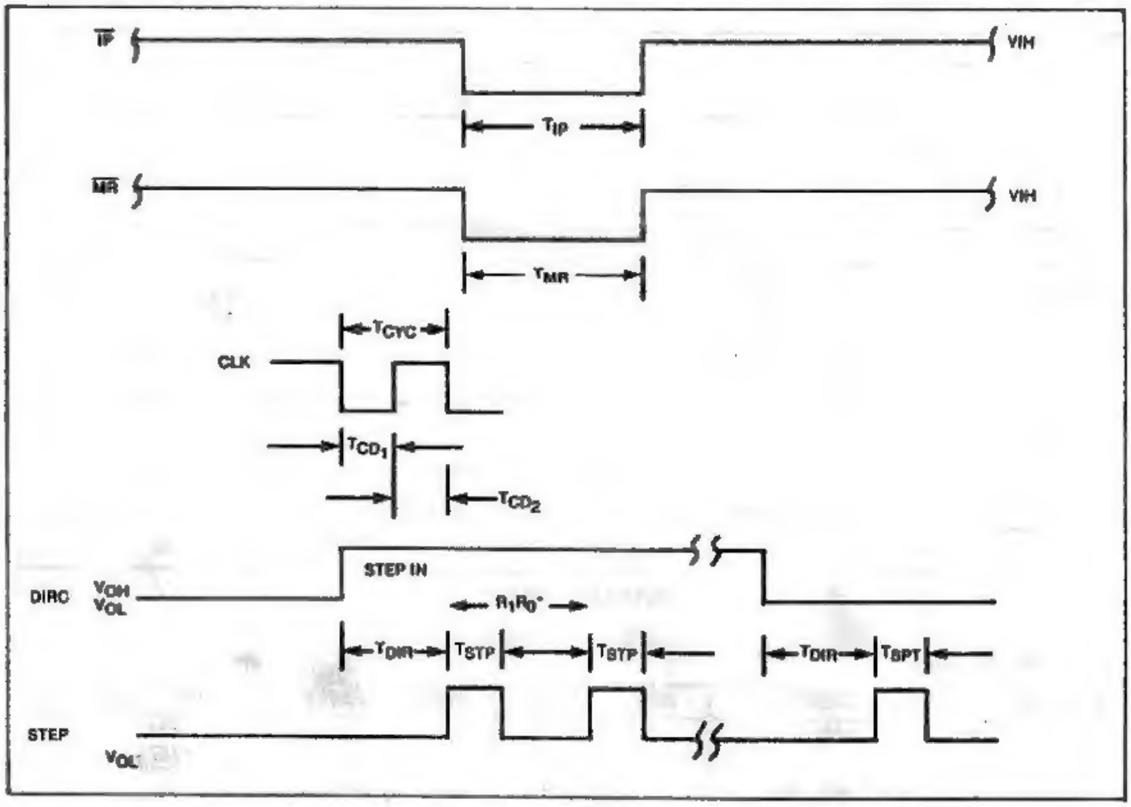
INPUT DATA TIMING:

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 8		μsec	MFM FM
TDIR	Dir Setup to Step		24 48		μsec	MFM 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