SUPER SIX

S-100 Single Board

Computer Technical Manual

ADC

ADVANCED DIGITAL CORPORATION

USA OFFICE
5432 PRODUCTION DRIVE
HUNTINGTON BEACH, CA 92649
TELEPHONE : (714) 891-4004
TELEX: 183210 ADVANCED HTBH

UNITED KINGDOM OFFICE
27 PRINCESS STREET
HANOVER SQUARE, LONDON W1R8NQ
UNITED KINGDOM
409-0077 / 409-3351
TLX 265840 FINEST

SUPER SIX™
S-100 Single Board Computer
Technical Manual

June 1, 1983

ADVANCED DIGITAL CORPORATION

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

SECTION I

INTRODUCTION

1.1 PURPOSE

This Manual provides the technical information necessary to install, operate and maintain the SUPER SIX singleboard computer by Advanced Digital Corporation.

1.2 EQUIPMENT OVERVIEW

SUPER SIX, produced by Advanced Digital Corporation, is the first single board computer for the S-100 bus running at 6MHz. Without the S-100 bus the SUPER SIX can also run as a standalone computer executing a single user CP/M 2.2 or 3.0, or a multiuser MP/M, OASIS, or TurboDOS operating system. SUPER SIX runs substantially faster than any other S-100 single board computer avialable in the market.

The SUPER SIX contains the following set of capabilities:

1. Z80B CPU operating at 6MHZ
2. 128K of dynamic bank select RAM arranged in 16k banks
3. Floppy Disk Controller which supports the 8-inch and 5.25-inch disk drives simultaniously
4. 2/4 K of shadow EPROM (Monitor)
5. 2 serial I/O (RS-232) ports offering software or hardware selectable baud rate, Z80B DART
6. 2 parallel ports (Z80B PIO)
7. Real time clock (Z80B CTC)
8. DMA controller (Z80 DMA)
9. Extended addressing: A16-A23
10. Single 5 volt supply on board
11. One-year warranty
12. Free copy of CP/M 2.2 BIOS supplied.

* NOTE: Items 5 and 6 require external adaptation for RS-232 and Centronics. The adapter boards contain a DB-25 connector on a 2-inch by 2-inch board attached to the back panel of the S-100 system (MODEM and RS 422 Paddle cards are also available).

1.3 DOCUMENT ORGANIZATION

This document is organized into 5 sections and 14 appendices.

Section I serves as an introduction to the entire document, stating the purpose of the document and providing an introduction to the SUPER SIX single board computer. This section also provides a list of acronyms used in the document and provides a statement on the responsibilities of document maintenance.

Section II provides a description of the operations of all components associated with the SUPER SIX single board computer.

Section III lists all SUPER SIX input/output ports and defines the assignments and functions of each port.

Section IV lists and defines all SUPER SIX jumper connections. This section includes the jumper assignments for factory (OEM) installed jumpers.

Section V describes the external connector pins for SUPER SIX connectors J1 through J5.

The appendices provide supplemental material to the body of the text and are referenced in the text at the associated points.

1.4 LIST OF ACRONYMS

Table 1-1 provides a listing and description of the acronyms used within this text.

ACRONYM	DESCRIPTION
CPU	Central Processor Unit
CTC	Controller/Timer Circuit
DART	Dual Asynchronous Receiver/Transmitter
DMA	Direct Memory Access
EPROM	Electrically Erasable Programmable Read-Only Memory
FDC	Floppy Disk Controller
IEEE	Institute of Electrical and Electronic Engineers
MP/M	Multiuser Program For Microcomputers
OEM	Original Equipment Manufacturer
PIO	Parallel Input/Output
PROM	Programmable Read-Only Memory

Table 1-1. List of Acronyms

Table 1-1. List of Acronyms (Continued)

ACRONYM	DESCRIPTION
RAM	Random Access Memory
SIO	Serial Input/Output
TTL	Transistor-transistor Logic
TurboDOS	A Multiuser Networking Operating System used as software with the SUPER SIX

1.5 DOCUMENT MAINTENANCE

This document is the property of Advanced Digital Corporation, who is responsible for its content. Any modifications made to this manual must be made with the express written approval of Advanced Digital Corporation.

1.6 THEORY OF OPERATION - START-UP PROCEDURE

The SUPER SIX Single board computer is shipped configured for 19. baud rate. The installation procedure is as follows:

1. Plug the PSNET/I to connector J5. Use caution; pin 1 is marked.

2. Connect the CRT. Pins 2, 3, 5, 7, and 20 must be used; no parity must be specified,

3. Apply power to the system. The monitor message shown in subsection 2.9.2 appears. Check the CRT baud rate; if 9600 baud is required, unplug jumper area J7, pin 7-8.

4. Install the floppy disk cable, load the CP/M diskette, and bootstrap the system.

Note: CP/M is shipped configured for 64K bytes of memory. The parallel port is configured as the default printer. 1024 bytes per sector read/writes are also supported (or DMA). The plus 8V and the plus/minus 16V on the S-100 bus must be verified prior to installing the SUPER SIX board.

SECTION II
OPERATION

This section describes the operation of all SUPER SIX components.

2.1 FLOPPY DISK CONTROLLER

The floppy disk controller can access up to four 8-inch or four 5.25-inch disk drives or any combination of the two. The controller can read and write IBM 3740 single density format and double density 1024 sector-size formats. Data transfer is performed via Direct Memory Access (DMA). Due to the simultanious operation capability of the SUPER SIX the format compatibility problems with 5.25-inch disks have been eliminated. The floppy disk controller used is the WD2793. The WD2793 has on-chip PLL data separators and on-chip write pre-compensation logic. Adjustments for PLL are factory set and write pre-compensation has been provided with the SUPER SIX. 50 Pin and 34 pin connectors are available for 8-inch and 5.25-inch disk drives respectively.

> NOTE: Customer adjustment of trim pots may result in cancellation of warranty.

2.2 THE 128K DYNAMIC RAM

The 128K RAM array can be switched ON and OFF in 16K increments, (0-16K, 16K-32K, 32K-48K, 48K-64K for both banks) under software control. This feature allows the CPU to access bank switchable external memory on the S-100 bus. The memory has an access time of 150ns. A Refresh operation is performed during Z80 M1 cycles and during WAIT and RESET states. The memory can be accessed by floppy disk via DMA, serial and parallel I/O, or another DMA device on the S-100 bus.

> *NOTE: Any external DMA device that is using continous mode DMA cycles must transfer data at an average rate of 15ms per byte or faster when holding the DMA request line for more than 1.5ms. The RAM row address is the low order address; therefore the entire RAM array is refreshed by DMA device every 128 contiguous memory cycles.

Under CPM 2.2 or CPM 3.0 the additional 64K can be used as a disk buffer. The SUPER SIX is ideal when operating in the bank mode under CP/M 3.0, as 128K RAM is required.

2.3 SYSTEM MONITOR EPROM

The system monitor EPROM is switched ON during reset. It can be disabled and enabled under software control. When enabled, the system monitor resides at locations F800-FFFF (hex) (refer to subsection 2.9.1.2) when using 2716 EPROM or at locations F000-FFFF (hex) when using the 2732 EPROM. The system monitor EPROM contains the cold-start loader for CP/M, MP/M and TurboDOS. In addition it can be used to perform LOAD, I/O READ and I/O WRITE operations. When the EPROM is disabled no system address space is used.

2.4 SERIAL PORTS

A 6MHz Z80B DART is used for the two serial I/O ports; a Z80B S10 or Z80A DART can be used in it's place (if a 4MHz Z80A DART is used the CPU and all other devices must also be 4MHz). This allows asynchronous serial data communication plus a variety of interrupt modes. Modem control signals are available at each serial connector. There are software selectable baud rates as well as hardware selectable baud rates (mini-jumpers J7).

> *NOTE: The serial ports are TTL and must be connected to PSNET/1 (serial adapter interface) for RS-232 communications. The J4 connector is for the CRT; the J5 connector is for the serial printer or CRT.

2.5 PARALLEL PORTS

A 6MHz Z80B PIO is used as the parallel port. The "A" channel of this chip is used to connect the parallel port connector (J2) to PIO. This port has an 8-bit bi-directional data line and two hand-shake lines. The "B" port can be split between the parallel port connector and the S-100 bus vectored interrupts lines by jumper options. This allows the port to be used as an additional parallel port, an interrupt controller, or both of the above. In the output mode the parallel ports can drive one TTL load.

2.6 REAL TIME INTERRUPT CLOCK

A 6MHz Z80B CTC is used for providing a real time system clock for MP/M or TurboDOS operating systems. Three channels of the CTC are available to the user for jumpering to synchronous baud rates or long clock times.

2.7 S-100 BUS INTERFACE

The S-100 bus interface provides the signals necessary for an 8-bit bus master as described by the IEEE-696 bus specification. Vectored interrupt lines VI0-VI7 are supported via jumper options (refer to section IV) and A16-A23 are also supported via an I/O port. The Phantom line is also implemented for the dynamic RAM array.

The SUPER SIX interface with the S-100 bus is depicted in Figure 2-1.

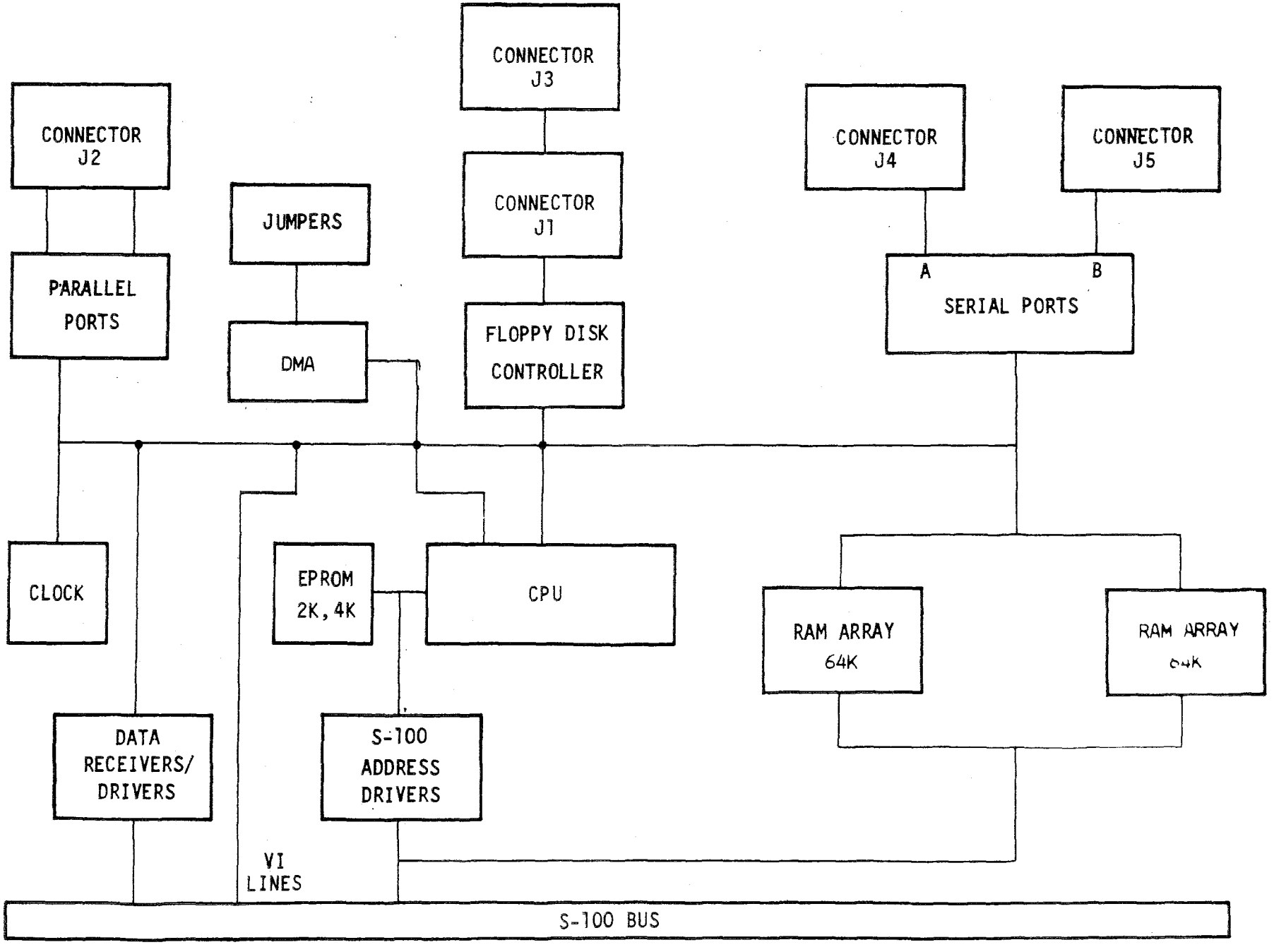


Figure 2-1. SUPER SIX Interface to the S-100 Bus

2.8 BAUD RATE JUMPER

Upon SUPER SIX initialization, the baud rate for the two serial channels can be hardware-selected independently by means of the baud rate jumper (J7). This 7-pole jumper is located between U70 and U68 and is divided into two sets of jumpers containing four and three pins for SIO channels A and B, respectively. Pins 7, 6, 5, and 4 set the baud rate for SIO channel A and are designated as A, B, C, and D, respectively; pins 3, 2, and 1 set the baud rate for SIO channel B and are designated as A, B, and C, respectively. Because this jumper comprises of only seven pin sets, SIO channel B has a hardware limitation of 1200 baud; by means of port 18 the software may be set to allow up to 19.2K baud for channel B. The baud rate settings, as determined by this jumper, are shown in Table 2-1, below. Once the SUPER SIX is initialized, I/O port 18 is used to modify the baud rate. Port 18 is described in subsection 3.2.21.

BIT D	BIT C	BIT B	BIT A	BAUD RATE
0	0	0	0	50
0	0	0	1	75
0	0	1	0	110
0	0	1	1	134.5
0	1	0	0	150
0	1	0	1	300
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	1800
1	0	0	1	2000
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19,200

Table 2-1. SUPER SIX Baud Rate Jumper Settings

2.9 EPROM AND MONITOR OPERATION

The on-board EPROM occupies addresses F000-FFFF (hex). This EPROM is switched ON automatically during RESET or POWER-ON. It contains the serial input/output (SIO) and floppy disk controller (FDC) initialization code along with a simple debugger and floppy disk cold-start loader. After the operating system is loaded the EPROM can be turned OFF to allow access to the RAM at address F000H-FFFFH. The EPROM can be enabled or disabled at any time to permit the calling of hardware dependant I/O routines.

2.9.1 EPROM Enable/Disable

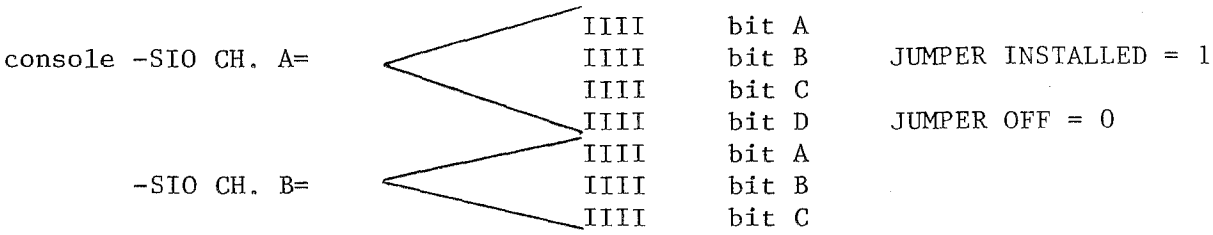
A listing of the program required to enable and disable the EPROM is provided below

BAUD RATE JUMPERS

ADVANCED DIGITAL CORPORATION SUPER SIX SUPPLEMENT

Update for Page 7

With the Super Six component side facing you and the S-100 connector down there are 7 jumpters between U68 and U70 aligned vertically. There are assigned as follows:



Bit D for SIO CH. B is not available as a jumper and must be set in software. Bit 7 of this input port (port 15) is instead used to sense double sided drives. Some software reads this bit and sends it to the SIO CH. B baud rate anyway. If you have double sided drives, this will make the bit a 0 thru creating baud rates from 50 to 1200. If you have single sided drives, this will make the bit a 1 and create baud rates from 1800 to 19.2K. If you experience a problem with this you can make a file under cpm to correctly assign the baud rate to SIO CH. B (used for serial printer and/or modem) as follows:

```
A>ddt
-a100
0100        in 15
0102        ani 7f (for 50 to 1200 baud. for 1800 to
                    19.2k type ori 80)
0104        out 18
0106        jmp 0
0109        <CR>
-g0

A>save 1 setbaud.com
A>setbaud
```

COMMON BAUD RATE SETTINGS

Console = 19.2k SIO CH. B = 300	Console = 19.2k SIO CH. B =1200	Console = 9600	Console = 12
(ON)	(ON)	(off)	(ON)
(ON)	(ON)	(ON)	(ON)
(ON)	(ON)	(ON)	(ON)
(ON)	(ON)	(ON)	(off)
(ON)	(ON)		
(off)	(ON)		
(ON)	(ON)		

2.9.1.1 Enabling the EPROM:

```
F033 3E4F    MVI A,01001111B    ;RESET POWER ON JUMP AND
                                ENABLE MEMORY, EPROM ON
F035 D316    OUT 16H            ;WRITE TO CONTROL PORT
```

2.9.1.2 Disabling the EPROM:

```
F033 3E4F    MVI A,01101111B    ;RESET POWER ON JUMP AND
                                ENABLE MEMORY, EPROM OFF
F035 D316    OUT 16H            ;WRITE TO CONTROL PORT
```

Jumper R25 configures the board to accept a 2716 or 2732 EPROM (as described in section IV).

> NOTE: The EPROM is always addressed at location F800 (hex) and can not be moved. Since the 2716 EPROM is 2K long it appears twice, at location F800-FC00 (hex) and at location FBFF-FFFF (hex).

2.9.2 Monitor Sign-on

The monitor signs-on with the following messages:

```
ADVANCED DIGITAL CORP.
Monitor Version  3.6
April - 1983
Press "H" for help
```

2.9.3 Monitor Commands

The monitor commands are shown in table 2-2.

COMMAND	FUNCTION
B	Loads the disk-boot loader
D ssss qqqq	Dumps memory in hex starting at user-specified address ssss and ending at user-specified address qqqq
F ssss qqqq bb	Fills memory from user-specified address ssss to user-specified address qqqq with bb
G AAAA	Goes to address AAAA
I pp	Input to user-specified port pp
L aaaa	Loads memory starting at user-specified address aaaa
M ssss qqqq dddd	Moves the contents of user-specified starting address ssss through user-specified ending address qqqq to the user-specified starting address of dddd
O pp dd	Output user-specified data dd to port pp
ESC	Terminates any command

Table 2-2. Monitor Commands For the SUPER SIX

The cold-start loader will select and home drive 0. Track 0 sector 1 will be read into memory at location 0. Single density is assumed for track 0. If an error occures an error code will be printed. The error code must be translated using the table in appendex G.

2.9.4 Cold Start Program

The cold start program is listed below.

```
                                        ;READ TRACK 0 SECTION 1 INTO
                                         MEMORY
                              BOOT 5:
                    MVI       A,018H
                    OUT       WAIT      ;set double for 5 inch

                              BOOT:
F4B5 3E0D           MVI       A,0DH     ;RESET FDC
F4B7 D30C           OUT       FDC       ;ISSUE COMMAND
F4B9 00             NOP

                              FDCW1:
F4BA DB0C           IN        FDC       ;CHECK BUSY
F4BC 0F             RRC
F4BD DABAF4         JC        FDCW1
F4C0 00             NOP                 ;KILL TIME
F4C1 00             NOP
F4C2 00             NOP
F4C3 00             NOP
F4C4 3E03           MVI       A,3       ;GET A RESTORE
F4C6 D30C           OUT       FDC       ;ISSUE COMMAND
F4C8 00             NOP
F4C9 DB14           IN        WAIT      ;WAIT FOR
F4CB 00             NOP                 ;INTRQ

                              TK0:
F4CC DB0C           IN        FDC
F4CE E604           ANI       4         ;CHECK TRACK 0
F4D0 CACCF4         JZ        TK0
F4D3 AF             XRA       A
F4D4 6F             MOV       L,A       ;POINT AT LOC 0
F4D5 67             MOV       H,A
F4D6 3C             INR       A
F4D7 D30E           OUT       FDCSEC    ;SET SECTOR
F4D9 3E8C           MVI       A,08CH    ;GET READ COMMAND
F4DB D30C           OUT       FDC       ;ISSUE COMMAND
F4DD 00             NOP

                              FDCRD:
F4DE DB14           IN        WAIT      ;WAIT FOR INTRQ
F4E0 B7             ORA       A         ;OR DRQ
F4E1 F2EBF4         JP        BOOTDN    ;EXIT IF INTRQ
F4E4 DB0F           IN        FDCDATA   ;GET DATA
F4E6 77             MOV       M,A       ;STORE
F4E7 23             INX       H         ;POINT NEXT
F4E8 C3DEF4         JMP       FDCRD
```

```
                                BOOTDN:
F4EB DB0C       IN      FDC     ;CHECK STATUS
F4ED B7         ORA     A       ;0 = NO ERROR
F4EE CA0000     JZ      0       ;OK, GO
F4F1 F5         PUSH    PSW     ;SAVE ERROR
F4F2 210FF6     LXI     H,BTERR ;PRINT
F4F5 CDE6F0     CALL    MSG     ;DISK ERROR
F4F8 F1         POP     PSW     ;GET ERROR
F4F9 CD21F1     CALL    THXB    ;PRINT IT
```

2.10 RAM ORGANIZATION

The SUPER SIX 128K RAM is configured as shown in Figure 2-2.

U55	U46	U47
U57	U48	U56
U58	U49	U50
U60	U51	U59
U61	U52	U53
U63	U54	U62

Figure 2-2. SUPER SIX RAM Configuration

The first 64K bank of RAM comprises of U46, U47, U48, U49, U51, U52, U54, U53, and U50; U46 is the parity chip.

2.11 Z80A DMA FEATURES

The Z80A DMA performs transfers, searches and search/transfers on a full-byte basis in burst or continuous modes. The cycle length and edge timing can be programmed to match the speed of any port. A bit maskable byte search can be performed either concurrently with transfers or as an operation itself.

2.12 PSNET/1 OPERATION

This paddle card converts TTL to RS232 levels. Pin 6 of the 14 pin connector on the card represents TXD; pin 7 is RTS*; pin 8 is DTR*; pin 5 is CTS*; 1 is DCD* (normally GND); 3 is RNG* optional; 2 is DSR*; 4 is RXD. Only pins 3, 5, 20, 2, and 1 are required for most printers or CRT's. Printers employing the BUSY line must be tied to pin 20 of the DB-25 connector on PSNET/1. A PSNET/1 schematic is provided in Appendix L.

2.13 PSNET/PAR

This paddle card connects the SUPER SIX parallel I/O to a Centronics printer or any other device that requires buffered signals. Note that a DB-25 connector is used to simplify the connection for the back panel of the S-100 system. Thirteen wires are required between the DB-25 and the printer. A PSNET/PAR schematic is provided in Appendix M.

2.14 SYNCHRONOUS OPERATIONS

If synchronous operation is required, the CTC channels (all four) are unused; the jumper option on H and T can be used to bring external clock into the SIO. The 14 pin connectors (J4 and J5), pins 1 and 11, are not used and can be employed for RNG. SIO pins 13 and 14 are connected on the SUPER SIX board and must be cut for synchronous operation.

2.15 POWER CONSUMPTION

+8V	2.8 Amp typical
+/-16V	250 ma

SECTION III
INPUT/OUTPUT PORTS

3.1 INPUT/OUTPUT PORT ASSIGNMENTS

Input/Output port assignments are shown in Table 3-1:

ADDRESS		FUNCTION
00	Read/Write	SIO channel A Data port
01	Read/Write	SIO channel A status/control port
02	Read/Write	SIO channel B Data port
03	Read/Write	SIO channel B status/control port
04	Read/Write	PIO channel A Data port
05	Write	PIO channel B Data port
06	Read/Write	PIO channel A control port
07	Write	PIO channel B control port
08	Read/Write	CTC channel 0 control port
09	Read/Write	CTC channel 1 control port
0A	Read/Write	CTC channel 2 control port
0B	Read/Write	CTC channel 3 control port
0C	Read/Write	FDC command/status port
0D	Read/Write	FDC Track register
0E	Read/Write	FDC sector register
0F	Read/Write	FDC data port
10	Read/Write	DMA control port
11	Read/Write	Same as port 10
12	Read/Write	Same as port 10
13	Read/Write	Same as port 10
14	Read/Write	FDC synchronization/Drive/Density
15	Write	S-100 bus extended address A16-A23
15	Read	On-board Baud Rate jumpers
16	Write	On-board memory control port #0
17	Write	On-board memory control port #1
18	Write	Set Baud Rate
19	Write	Same as port 18
1A	Write	Same as port 18
1B	Write	Same as port 18

Table 3-1. I/O Port Assignments for the SUPER SIX Board

Note: All Address in table 3-1 are listed in Hex. The unused input/output ports are internally decoded and should not be used by external S-100 I/O boards.

The individual ports are described in detail in the following subsection.

3.2 I/O PORT DESCRIPTIONS

This subsection discribes the function of all SUPER SIX I/O ports.

3.2.1 Port 00

This read/write port acts as the serial input/output channal A data port and is described in detail in Appendix A.

3.2.2 Port 01

This read/write port acts as the serial input/output channel A status/control port and is discribed in Appendix A.

3.2.3 Port 02

This read/write port acts as the serial input/output channel B data port and is discribed in detail in Appendix A.

3.2.4 Port 03

This read/write port acts as the serial input/output channel B status/control port and is described in detail in Appendix A.

3.2.5 Port 04

This read/write port acts as the parallel input/output channel A data port and is discribed in detail in Appendix B.

3.2.6 Port 05

This write-only port acts as the parallel input/output channel B data port and is described in detail in Appendix B. This port can be jumpered to the S-100 vectored intercept lines onto connector J2 (refer to Section IV).

3.2.7 Port 06

This read/write port acts as the parallel input/output channel A control is discribed in detail in Appendix B.

3.2.8 Port 07

This write-only port acts as the parallel input/output channel B control port and is described in detail in Appendix B. This port can be jumpered to the S-100 vectored interrupt lines onto connector J2 (refer to Section IV).

3.2.9 Port 08

This read/write port acts as the counter/timer circuit channel zero control port and is discribed in detail in Appendix C.

3.2.10 Port 09

This read/write port acts as the counter/timer circuit channel 1 control port and is discribed in detail in Appendix C.

3.2.11 Port 0A

This read/write port acts as the counter/timer circuit channel 2 control port and is discribed in detail in Appendix C.

3.2.12 Port 0B

This read/write port acts as the counter/timer circuit channel 3 control port and is described in detail in Appendix C.

3.2.13 Port 0C

This read/write port acts as the floppy disk control command and status port and is described in detail in Appendix F.

3.2.14 Port 0D

This read/write port acts as the floppy disk control track register port and is discribed in detail in Appendix F.

3.2.15 Port 0E

This read/write port acts as the floppy disk control sector register port and is discribed in detail in Appendix F.

3.2.16 Port 0F

This read/write port acts as the floppy disk control data port and is described in detail in Appendix F.

3.2.17 Ports 10-13

These ports act as direct memory access (DMA) control ports. The Z80A DMA controller requires only one control/status port; ports 11, 12, and 13 (hex) are therefore identical to port 10 (hex). The floppy disk controller (FDC) data request line is tied to the Z80 DMA ready input and is active when the signal is high. The DMA controller can move data to and from the FDC and memory, any I/O channel and memory, or memory and memory, on or off the board.

> *NOTE: All DMA other than FDC must be started by ready = low (e.g., FDC DRQ = low) and sycronized using the S-100 wait line, pin 72. The DMA interrupt output is connected to interrupt jumper J6. INT from the DMA is active when low.

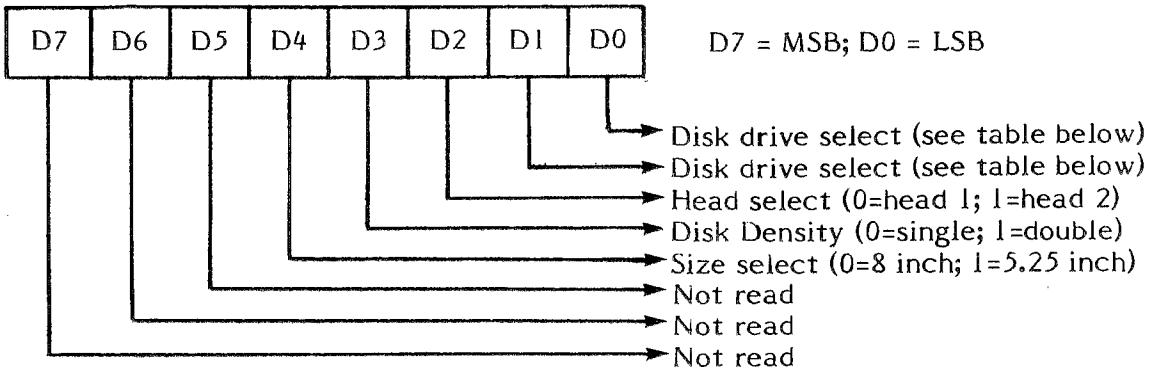
3.2.18 Port 14

This read/write port acts as the floppy disk controller synchronization, drive identification, and density port.

3.2.18.1 Port 14, Write Operation

The two low-order bits (D0 and D1) specify the disk drive (see table below). Bit D2 selects the disk head. Bit D3 specifies the disk density, where zero (0) specifies single density and 1 specifies double density. Bit D4 specifies the disk drive size, where zero (0) specifies 8-inch disk and 1 specifies 5.25-inch disk.

The bit significance is shown below.



The two low-order bits (D0 and D1) specify the selected disk drive, where:

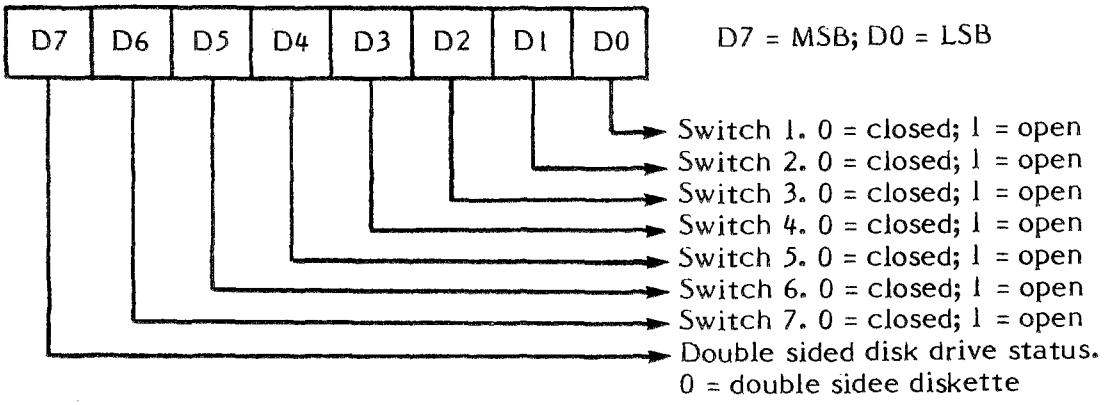
D1	D0	RESULT
0	0	Disk Drive 0 selected
0	1	Disk Drive 1 selected
1	0	Disk Drive 2 selected
1	1	Disk Drive 3 selected

Two connectors are provided on the SUPER SIX for the floppy disk: J1 and J3. J1 drives the 5.25-inch floppy disk; pin 1 is clearly marked to this effect. J3 drives the 8 inch disk.

3.2.19 Port 15

3.2.19.1 Port 15, Read Operation

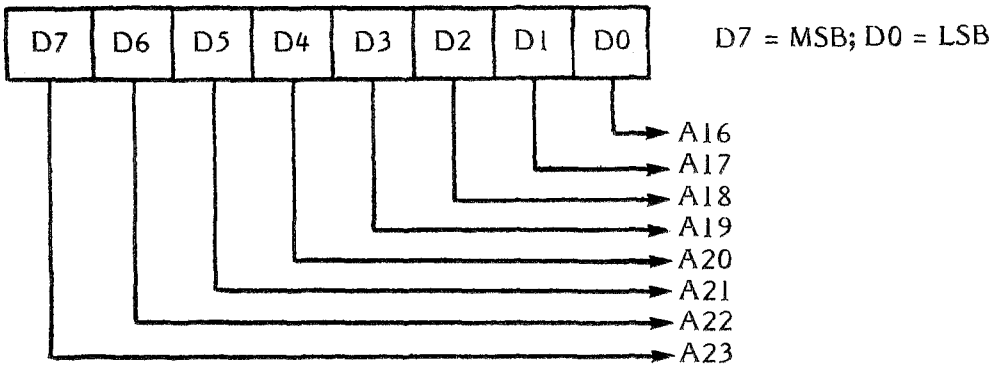
This port reads the on-board baud rate. The bit significance is shown below.



Note: Jumper J7 specifies the hardware setting for the baud rate. Port 15 reads jumper J7 to determine the hardware specified rate.

3.2.19.2 Port 15, Write Operation

This port controls the S-100 extended address lines. The bit significance is shown below:



3.2.20 Ports 16 and 17

These write-only ports act as the on-board memory control ports zero (0) and 1 respectively. These ports control the on-board memory management circuit, PROM enable and disable circuit, power-on jump reset circuit, and the parity check enable. Port zero (0) controls the first 64K bytes of memory; port 1 controls the second 64K of memory.

3.2.20.1 Port 16, Write Operation

The four low-order bits (D0, D1, D2, and D3) switch the on-board memory ON and OFF in 16K banks. Relative addresses of these banks are as shown below, (in hex).

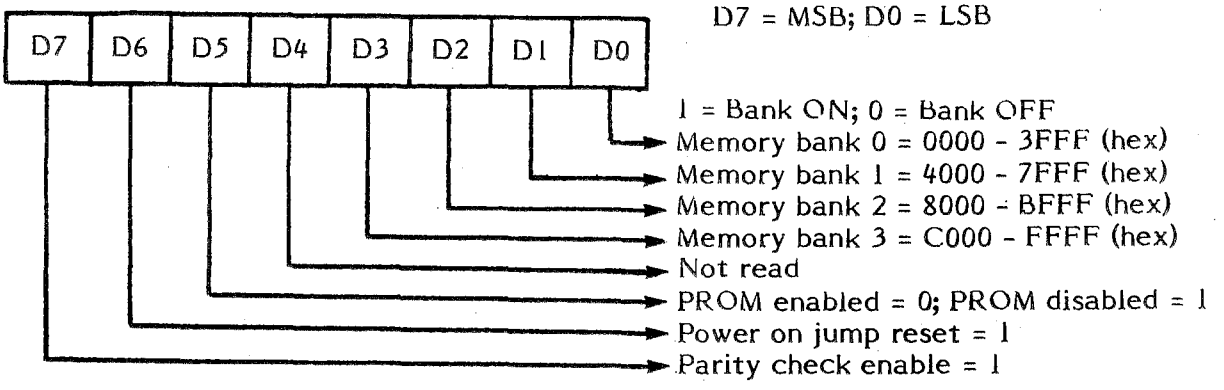
BANK	FROM	TO
D0	0000	3FFF
D1	4000	7FFF
D2	8000	BFFF
D3	C000	FFFF

Bit 5 of this port switches the on-board EPROM ON and OFF. This EPROM occupies addresses F000 through FFFF (Hex). After the operating system is loaded the EPROM can be set to OFF to enable accessing of the RAM addresses F000 through FFFF (hex); it is switched to ON Automatically during reset or power-on. This EPROM contains the SIO and FDC initialization code, a simple debugger, and the floppy disk cold-start loader.

Bit D6 resets the power-on jump circuit; it must be reset high before RAM can be accessed upon completion of a reset or power-on operation.

Bit D7 enables and disables the on-board memory parity check latch. On power-up or reset, this bit is set to low and disables the parity circuit. Prior to enabling the parity circuit, all memory must be written; this initializes the parity bit for each location. If a parity error occurs an on-board LED is turned ON. If interrupts are enabled and the board is jumpered for parity error interrupts, the location in error can be located by clearing bit D7 (to clear the error), resetting the bit, and then reading each location again; If no parity error is encountered it is likely that the error encountered was a soft read error. Parity error interrupts can be jumpered to create a non-maskable or counter/timer channel interrupt.

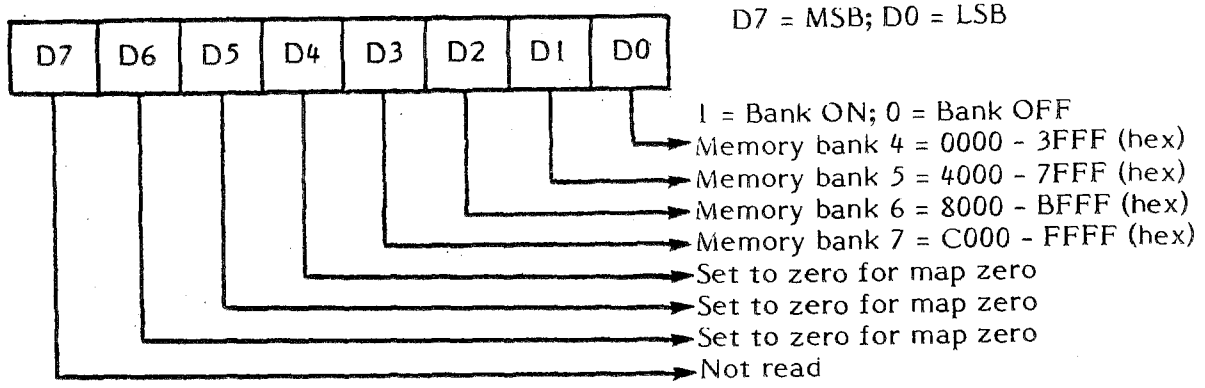
The bit significance of this port is shown below.



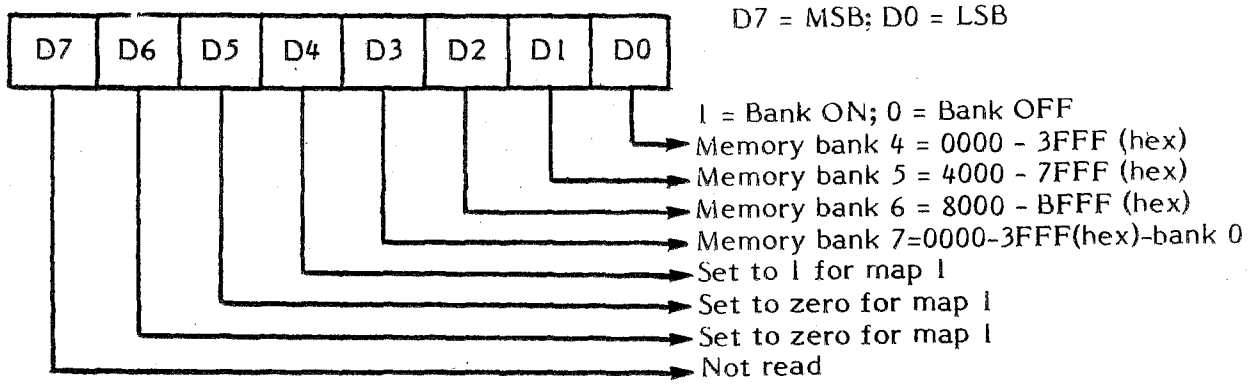
3.2.20.2 Port 17, Write Operation

Bits D0, D1, D2, and D3 switch the second 64K set of on-board memory similarly to the corresponding bits for port 16. Bits D4, D5, and D6 control memory mapping of the second set of memory. The first set of 64K bytes is not affected by these settings. Five memory maps are available and are depicted below. Note that if any 16K segment from the first bank is selected to overlay a 16K segment from the second bank, only the first bank's 64K memory is selected.

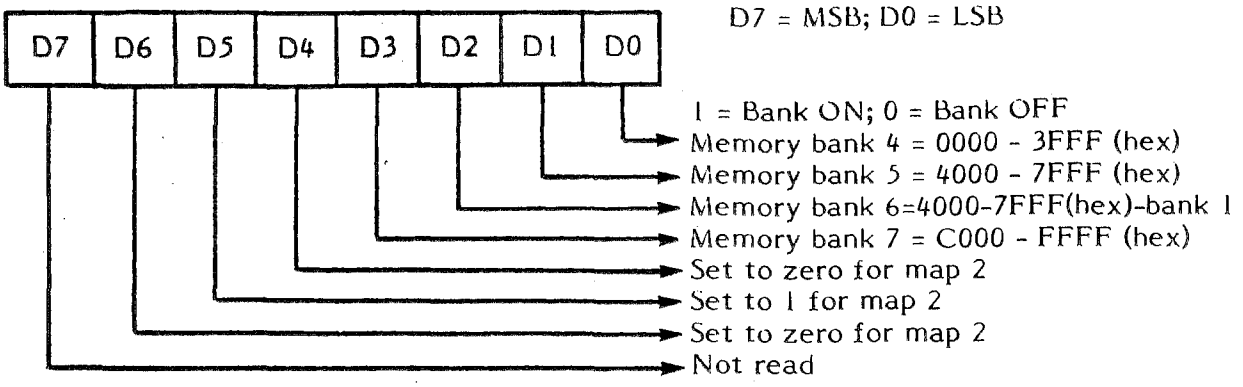
The bit significance of map 0 is shown below. No overlays may be performed on this bank.



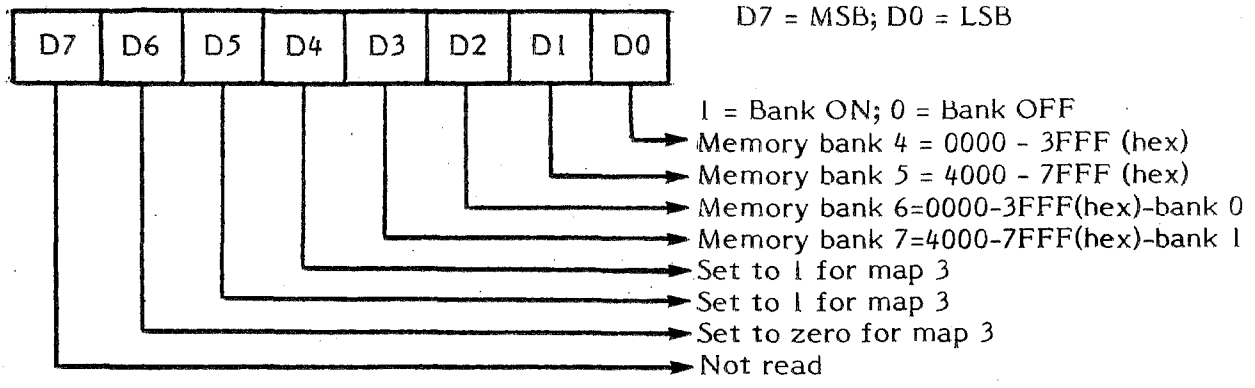
The bit significance of map 1 is shown below. Bank zero of the first memory set may overlay bank 7.



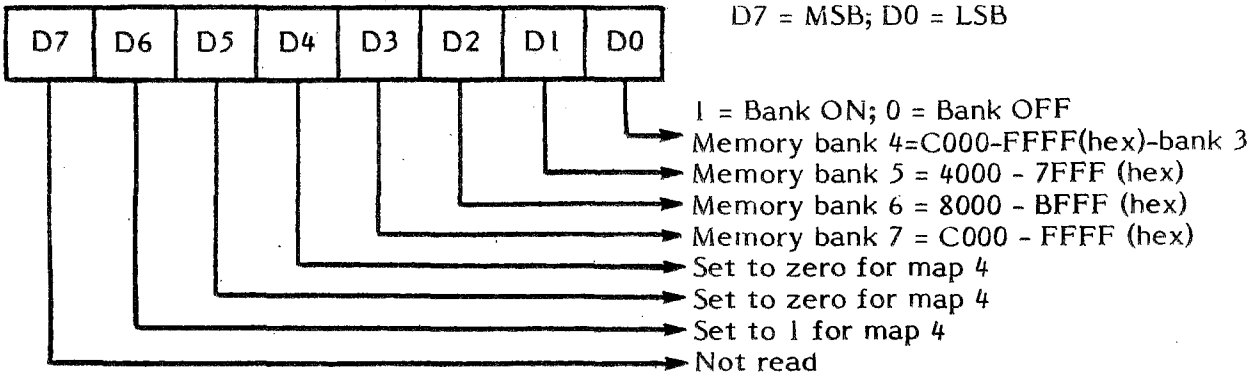
The bit significance of map 2 is shown below. Bank 1 of the first memory set may overlay bank 6.



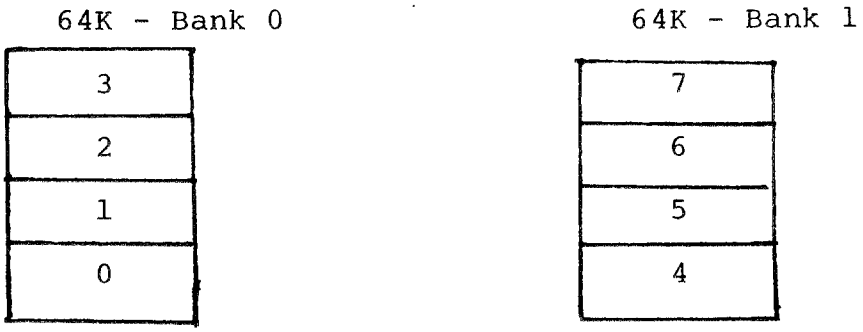
The bit significance of map 3 is shown below. Banks zero and 1 of the first memory set overlay banks 6 and 7, respectively.



The bit significance of map 4 is shown below. Bank 3 of the first memory set overlays bank 4.



EXAMPLES OF BANKING:



Bank 0 - Controlled by Port 16
There are five maps possible in Bank 1.
Bank 0 always overrides. Once a bit in Bank 0 is set, you will always get the corresponding memory in that Bank only.

Bank 1 - Port 17.
Depending on the map used.

Map 0 - You will get the corresponding memory in Map 0, Bank 1 if you don't have the Bank 0 bit set.

Map 1 - If you set bit 0 , then you will get 0-16K, if you set just bit 3 , then you get 0-16K which corresponds with 48-64K in Map 0.

MAP 0

3/7	C000
2/6	8000
1/5	4000
0/4	0

MAP 1

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

MAP 2

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

MAP 3

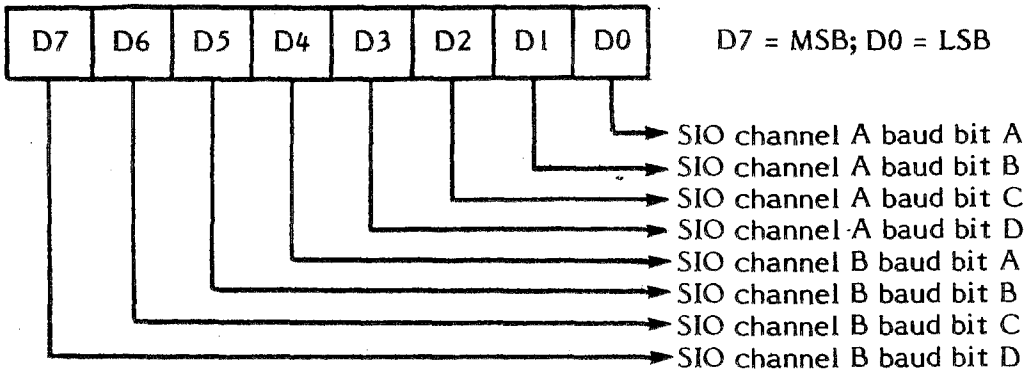
3
2
1/5/7
0/4/6

MAP 4

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

3.2.21 Port 18

This write-only port acts as the set-baud-rate port and allows the user to perform a software reset of the baud rate for SIO channels A and B. The bit significance of this port is shown below.



The baud rates equated with bit settings A, B, C, and D are shown in Table 3-2. These are identical for both channel A and channel B.

BIT D	BIT C	BIT B	BIT A	BAUD RATE
0	0	0	0	50
0	0	0	1	75
0	0	1	0	110
0	0	1	1	134.5
0	1	0	0	150
0	1	0	1	300
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	1800
1	0	0	1	2000
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19,200

Table 3-2. SUPER SIX Port 18 Baud Rate Settings

SECTION IV
JUMPER CONNECTIONS

4.1 JUMPER DEFINITIONS

Table 4-1 presents the available jumpers and abbreviates their assigned functions. Detailed descriptions of individual jumper functions are presented following this table. To facilitate jumper insertion, each three (or more) pin jumper is depicted in an illustration contained within the narrative for that jumper.

JUMPER	FUNCTION
A	Generates Wait States
B	Selects between on-board Tri-state and S-100 bus buffering (MWRITE)
C	Selects 6MHz or 4MHz operation
D	Selects 6MHz or 4MHz operation For DMA ONLY (2-3)
E	Selects between the S-100 bus interrupt (INT) and the vectored interrupt zero (VI0)
F	Used in conjunction with jumpers P, N, M, K, R, S, and J, this jumper area selects between vectored interrupts and the parallel I/O second port
G	Used in adjusting the floppy disk controller phase line level (PLL)
H & T	This jumper area selects synchronous or asynchronous serial I/O
J	Used in conjunction with jumpers P, N, M, K, R, S, and F, this jumper area selects between vectored interrupts and the parallel I/O second port
J6	Used for interrupts of the CTC to 1.5 MHz, DMA or parity error, NMI or INTRQ
J7	Used to set the hardware baud rate
K	Used in conjunction with jumpers P, N, M, J, R, S, and F, this jumper area selects between vectored interrupts and the parallel I/O second port
M	Used in conjunction with jumpers P, N, J, K, R, S, and F, this jumper area selects between vectored interrupts and the parallel I/O second port

Table 4-1. SUPER SIX Jumpers and Abbreviated Functions

JUMPER	FUNCTION
N	Used in conjunction with jumpers P, M, J, K, R, S, and F, this jumper area selects between vectored interrupts and the parallel I/O second port
P	Used in conjunction with jumpers N, M, J, K, R, S, and F, this jumper area selects between vectored interrupts and the parallel I/O second port
R	Used in conjunction with jumpers P, N, M, J, K, S, and F, this jumper area selects between vectored interrupts and the parallel I/O second port
R25	Used to select the 2716 or the 2732 EPROM
S	Used in conjunction with jumpers P, N, M, J, K, R, and F, this jumper area selects between vectored interrupts and the parallel I/O second port

Table 4-1. SUPER SIX Jumpers and Abbreviated Functions (Continued)

4.2 JUMPER DESCRIPTIONS

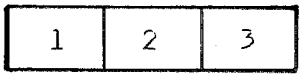
4.2.1 Jumper A

This two-pin jumper must be installed when running DMA.

4.2.2. Jumper B

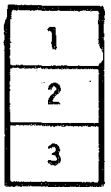
When installed this two-pin jumper causes buffering of the MWRITE signal to be performed through the on-board tri-state buffer. When not installed, buffering is performed through the S-100 bus. In most cases this jumper is installed.

4.2.3 Jumper C



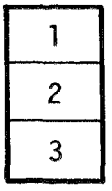
When installed between 1 and 2, this jumper sets the board for 6MHz operation. When installed between 2 and 3, 4MHz operation is specified. Also see jumper D, below.

4.2.4 Jumper D



When installed between 1 and 2, this jumper sets the board for 6MHz operation. When installed between 2 and 3, 4MHz operation is specified. Also see jumper C, above. For DMA Operation plug between 2-3 always.

4.2.5 Jumper E

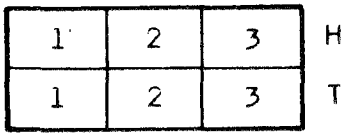


This jumper determines whether the S-100 bus interrupt (INT) or the vectored interrupt zero (VI0) is to be used. When installed between pins 1 and 2, the VI0 is tied to priority interrupt DAISY CHAIN. When installed between pins 2 and 3, the priority interrupt (S-100 pin 73) is tied to the interrupt DAISY CHAIN.

4.2.6 Jumper G

This jumper adjusts the floppy disk controller phase-locked loop (PLL) and is not installed unless adjusting the PLL.

4.2.7 Jumpers H and T



This jumper area specifies that serial I/O is to be performed synchronously or asynchronously. When installed between pins 1 and 2 of jumpers H and T, synchronous operation is selected; when installed between pins 2 and 3, asynchronous operation is selected. Pins 1 and 11 of the 14 pin J4 and J5 connectors are not factory connected. These may be used in synchronous operations to bring the external clock or ring indicator into the SIO.

4.2.8 Jumper J6

This jumper is for CTC external interrupts. Pin 14 is the 1.5MHz clock; Pin 12 is INTDMA; pin 10 is PERR (parity error); pin 9 is NMI; pin 8 is INTRQ. Jumper J6 internal interrupts are 7-6, 5-4, and 3-2.

4.2.9 Jumper R25

This jumper allows selection between the 2716 and 2732 EPROMs. When installed between pins 1 and 2, the 2716 EPROM is addressed; when installed between pins 2 and 3, the 2732 EPROM is addressed.

4.2.10 Jumper J7

GND	8	7	BD_0
GND	9	6	BD5
GND	10	5	BD4
GND	11	4	BD3
GND	12	3	BD2
GND	13	2	BD1
GND	14	1	BD0

This jumper sets the hardware baud rate. When the jumper is inserted an inverted ground condition exists (1); when not inserted a zero (0) value is assumed. This setting must agree with the Set Baud Rate Port (port 18 - refer to subsection 3.2.21) e.g., when 7, 6, 5, and 4 are installed, 19.2K baud is read.

4.2.11 Jumpers P, N, M, K, J, R, S, and F

	3	2	1
P			
N			
M			
K			
J			
R			
S			
F			

These jumpers select between vectored interrupts or the Parallel I/O second port. When installed between pins 1 and 2, the vectored interrupts zero (0) to 7, as selected, are enabled and connected to Parallel I/O port B. When jumpers are installed between pins 2 and 3, Parallel I/O port B is connected to the J2 connector.

4.3 FACTORY AND OEM INSTALLED JUMPERS

4.3.1 8-Inch Floppy Disk Drive

Install jumpers as indicated below.

JUMPER	BETWEEN PINS	FUNCTION
B		Enables the S-100 bus memory write signal
C	1 - 2	Set for 6MHz operation
D	2-3	Set for 4MHZ DMA
E	1 - 2	Vectored Interrupt zero
H	2 - 3	Specifies asynchronous SIO
T	2 - 3	Specifies asynchronous SIO
J6	1-14, 7-8	Between 1-14 Clock/TRG3 to TO2 of the CTC
	3 - 2	Between 7-8 Clock/TRG2 to TO1 of the CTC
		Between 2 - 3: Clock TRG1 to TO0 of the CTC
J7	5, 6, 7, 8	19.2K baud rate
P	2 - 3	Selects parallel port B, Bit D0 on J2-25
N	2 - 3	Selects parallel port B, Bit D1 on J2-27
M	2 - 3	Selects parallel port B, Bit D2 on J2-29
K	2 - 3	Selects parallel port B, Bit D3 on J2-31
J	2 - 3	Selects parallel port B, Bit D4 on J2-33
R	2 - 3	Selects parallel port B, Bit D5 on J2-35
S	2 - 3	Selects parallel port B, Bit D6 on J2-37
F	2 - 3	Selects parallel port B, Bit D7 on J2-39

4.3.2 Shugart SA 800/801 Disk Drive

Remove all factory installed jumpers and install as follows:

```
Jumper  C
Jumper  T2
Jumper  DS1 (Drive Select 1)
Jumper  DS
Jumper  RR
Jumper  800
Jumper  A
Jumper  B
Jumper  T1
Jumper  T3, T4, T5, T6, for end of cable
```

4.3.3 Shugart 850 Disk Drive

Install the terminator at the cable end. Remove all factory installed jumpers and install as follows:

Jumper 2S
Jumper C
Jumper A
Jumper B
Jumper R
Jumper I
Jumper Y
Jumper 850
Jumper S2
Jumper IT
Jumper FS
Jumper RM
Jumper DS
Jumper S
Jumper M
Jumper FM, MFM
Jumper DS1 (for drive 1)
Jumper DS2 (for drive 2)

4.3.4 MFE Model 700 Disk Drive

Install jumpers as indicated below:

Jumper J3
Jumper RHL
Jumper J4
Jumper L-1
Jumper J3
Jumper WP1
Jumper J1-4, J1-8
Jumper J10
Jumper J7
Jumper SSI
Jumper SE2
Jumper J11, HS1

4.3.5 Tandon Slim Line Disk Drive

Install the terminator at the cable end. Install jumpers as indicated below:

Jumper DS1 (for drive 1)
Jumper DS2 (for drive 2)

4.3.6 NEC Model FD1160 Disk Drive

Install jumpers as indicated below:

Jumper C
Jumper N
Jumper HLS
Jumper M
Jumper PRI
Jumper DLD
Jumper FU

4.3.7 QUME Data Track 8 Disk Drive

Install and cut jumpers as indicated below:

2S, S2
Jumper DS1 (for drive 1)
Jumper DS2 (for drive 2)
Jumper Y
Jumper C
Cut X
Cut Z
Cut L

4.3.8 Tandon 5.25-inch Disk Drive (48TPI)

Install jumpers on the DIP shunt as indicated below:

Jumper pin 7
Jumper pin 8
Jumper pin 2 (for drive 1)
Jumper pin 3 (for drive 2)

4.3.9 Mitsubishi Model 2894 Disk Drive

Install and cut jumpers as indicated below:

Jumper Y
Jumper E
Jumper off
Jumper R
Jumper DS (drive select)
Jumper S2
Jumper D
Jumper H
Jumper I

4.3.10 Mitsubishi Model 2896 Disk Drive

Install jumpers as indicated below:

Jumper	JFG	Jumper	B
Jumper	PS	Jumper	RS
Jumper	S1 next to SE	Jumper	HUD
Jumper	S2	Jumper	Z
Jumper	M1	Jumper	WP
Jumper	C	Jumper	HY
Jumper	I		
Jumper	R		
Jumper	It		
Jumper	DS (drive select)		
Jumper	RFa		
Jumper	A		

4.3.11 Shugart SA 860 Disk Drive

Install jumpers as indicated below:

Jumper DS
Jumper S2
Jumper SR
Jumper TR

4.3.12 Siemens FDD-100-8 Disk Drive

Install jumpers as indicated below:

Jumper	RR	Jumper	SS
Jumper	U	Jumper	GND
Jumper	H	Jumper	SE
Jumper	D	Jumper	Cut jumper 32 and connect jumper 0
Jumper	L	Jumper	E
		Jumper	2 for radial step
		Jumper	Cut jumper G & connect H
		Jumper	R1

Note: Jumper H is normally shipped as G. Jumper 0 is set for 8 or 16 sectors.

4.3.13 Qume Trak 592 5.25-inch Disk Drive

Install jumpers as indicated below:

Jumper HS
Jumper DS0

4.3.14 All Other Disk Drives

Refer to manufacturer-supplied technical manual for specific drive.

Note: The last drive at the end of the cable must be terminated.

SECTION V
EXTERNAL CONNECTOR PINS

5.1 CONNECTOR S-100

This connector interfaces SUPER SIX to the S-100 bus. Pin numbers and names are shown in Table 5-1.

PIN	NAME	PIN	NAME	PIN	NAME
1	+8V	25	PSTVAL*	58	SXTRQ*
2	+16V	26	PHLDA	59	A19
3	XRDY	27-28	RFU	60	SIXTN*
4	VI0*	29-34	A5,4,3,	61-64	A20-A23
5	VI1*		15,12,9	65-66	NDEF
6	VI2*	35	DO1/DATA1	67	PHANTOM*
7	VI3*	36	DO0/DATA0	68	MRWT
8	VI4*	37	A10	69	RFU
9	VI5*	38	DO4	70	GND
10	VI6*	39	DO5	71	RFU
11	VI7*	40-43	DO6,12	72	RDY
12	NMI*		13,17	73	INT*
13	PWRFAIL*	44	SMI	74	HOLD*
14	DMA3*	45	SOUT	75	RESET*
15	A18	46	SINP	76	PSYNCH
16	A17	47	SMEMR	77	PWR*
17	A16	48	SHLTA	78	PDBIN
18	SDSB*	49	CLOCK	79-87	A0-A11
19	CDSB*	50	GND	88-95	DO2-DI0
20	GND	51	+8V	96	SINTA
21	NDEF	52	-16V	97	SWO*
22	ADSB*	53	GND	98	ERROR*
23	DODSB*	54	SLAVE CLR*	99	POC*
24	0	55-57	DMA0*-DMA2*	100	GND

Table 5-1. S-100 Connector Pins and Pin Functions

5.2 CONNECTOR J2

This connector interfaces the SUPER SIX to the parallel port. Pin numbers, names, and functions are shown in Table 5-2.

PIN	NAME	FUNCTION
1	ARDY	PIO Channel A Ready Signal
2	ARDY RET	Ground
3	ASTRB*	PIO Channel A Strobe
4	ASTRB RET	Ground
5	PA0	PIO Channel A Data Bit 0
6	PA0 RET	Ground
7	PA1	PIO Channel A Data Bit 1
8	PA1 RET	Ground
9.	PA2	PIO Channel A Data Bit 2
10	PA2 RET	Ground
11	PA3	PIO Channel A Data Bit 3
12	PA3 RET	Ground
13	PA4	PIO Channel A Data Bit 4
14	PA4 RET	Ground
15	PA5	PIO Channel A Data Bit 5
16	PA5 RET	Ground
17	PA6	PIO Channel A Data Bit 6
18	PA6 RET	Ground
19	PA7	PIO Channel A Data Bit 7
20	PA7 RET	Ground
21	BRDY	PIO Channel B Ready Signal
22	BRDY RET	Ground
23	BSTRB*	PIO Channel B Strobe
24	BSTRB RET	Ground
25*	PB0	PIO Channel B Data Bit 0
26	PB0 RET	Ground
27*	PB1	PIO Channel B Data Bit 1
28	PB1 RET	Ground
29*	PB2	PIO Channel B Data Bit 2
30	PB2 RET	Ground
31*	PB3	PIO Channel B Data Bit 3
32	PB3 RET	Ground
33*	PB4	PIO Channel B Data Bit 4
34	PB4 RET	Ground
35*	PB5	PIO Channel B Data Bit 5
36	PB5 RET	Ground
37*	PB6	PIO Channel B Data Bit 6
38	PB6 RET	Ground
39*	PB7	PIO Channel B Data Bit 7
40	+5 Volts	

* - Can be jumpered to the S-100 Bus.

Table 5-2. SUPER SIX J2 Connector Pin Functions

5.3 CONNECTOR J3

This connector interfaces the SUPER SIX to the floppy disk controller. Pin numbers and functions are shown in Table 5-3.

8 INCH DISK PIN	5.25 INCH DISK PIN	FUNCTION
1		Ground
2		Alternate Head 2*
3		Ground
4		N/C
5		Ground
6		N/C
7		Ground
8		N/C
9		Ground
10		N/C
11		Ground
12		N/C
13		Ground
14		Head 2*
15		Ground
16		N/C
17	1	Ground
18		Head Load*
19	3	Ground
20	8	Index*
21	5	Ground
22		Ready*
23	7	Ground
24	8	Above Track 43*
25	9	ground
26	10	Drive Select 0*
27	11	Ground
28	12	Drive Select 1*
29	13	Ground
30	14	Drive Select 2*
31	15	Ground
32	6	Drive Select 3*
33	17	Ground
34	18	Direction
35	19	Ground
36	20	Step*
37	21	Ground
38	22	Write Data*
39	23	Ground
40	24	Write Gate*
41	25	Ground
42	26	Track 0*
43	27	Ground
44	28	Write Protect*
45	29	Ground
46	30	Read Data*
47	31	Ground
48	16	Motor ON*
49	33	Ground
50	34	N/C

Table 5-3. SUPER SIX J3 Connector Pin Functions

5.4 CONNECTOR J4

This connector interfaces the SUPER SIX to the serial port, channel A. Pin numbers, names, and functions are shown in Table 5-4.

PIN	NAME	FUNCTION
1	N/C	
2	DCDA*	Data Carrier Detect, Channel A*
3	SYNCA*	Sync Detect
4	RxDA	Receive data
5	CTSA*	Clear To Send
6	TxDA	Transmit Data
7	RTSA*	Request To Send
8	DTRA*	Data Terminal Ready
9	Tx/RxCA*	Transmit/receive Clock
10	GND	Ground
11	N/C	
12	+16 Volts	
13	-16 Volts	
14	+5 Volts	

Table 5-4. SUPER SIX J4 Connector Pin Functions

5.5 CONNECTOR J5

This connector interfaces the SUPER SIX with serial port channel B. Pin numbers, names, and functions are shown in Table 5-5.

PIN	NAME	FUNCTION
1	N/C	
2	DCDA*	Data Carrier Detect, Channel A*
3	SYNCA*	Sync Detect
4	RxDA	Receive data
5	CTSA*	Clear To Send
6	TxDA	Transmit Data
7	RTSA*	Request To Send
8	DTRA*	Data Terminal Ready
9	Tx/RxCA*	Transmit/receive Clock
10	GND	Ground
11	N/C	
12	+16 Volts	
13	-16 Volts	
14	+5 Volts	

Table 5-5. SUPER SIX J5 Connector Pin Functions

APPENDIX A

Z80A/Z80B SERIAL INPUT/OUTPUT (SIO) AND DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DART)

Z8470 Z80® DART Dual Asynchronous Receiver/Transmitter

Zilog

Product Specification

March 1981

Features

- Two independent full-duplex channels with separate modem controls. Modem status can be monitored.
- Receiver data registers are quadruply buffered; the transmitter is doubly buffered.
- 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.
- 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.
- 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.

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.

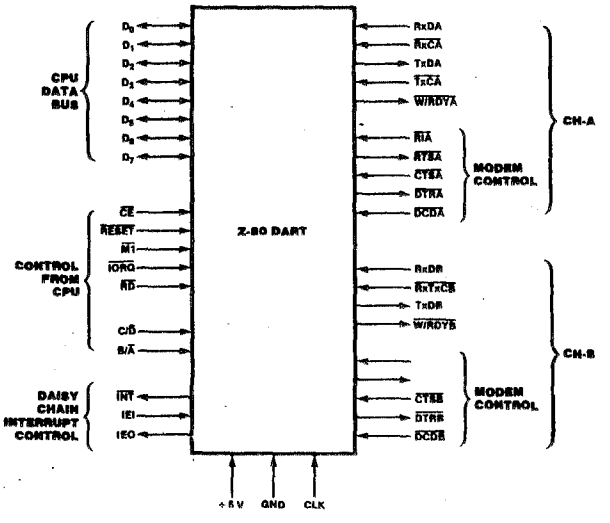


Figure 1. Z80 DART Pin Functions

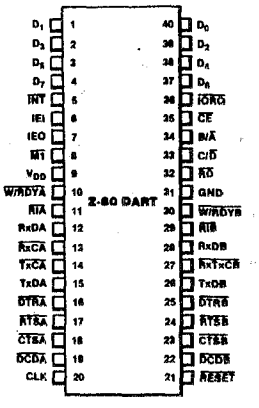


Figure 2. Pin Assignments

Pin Description

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 hand-shake 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.

A more detailed explanation of Z-80 DART operation can be found in the *Z-80 SIO Technical Manual* (Document Number 03-3033-01). Because this manual was written for the Z-80 SIO, it contains information about synchronous as well as asynchronous operation.

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.

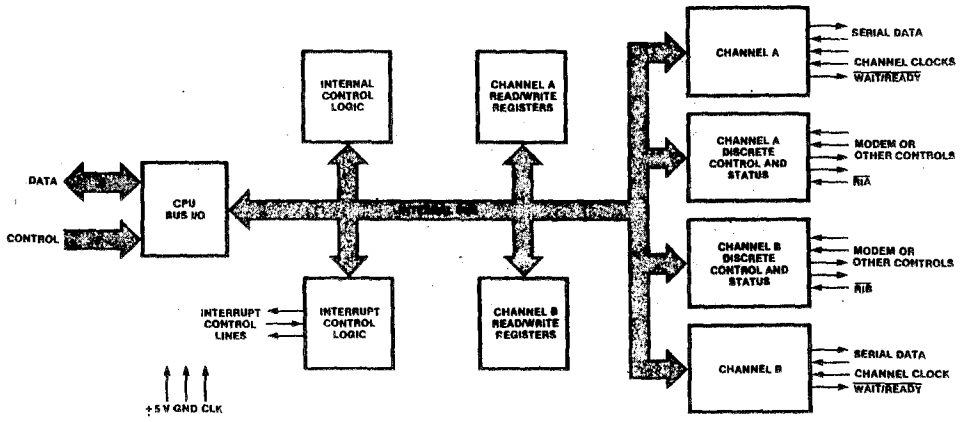


Figure 3. Block Diagram

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.

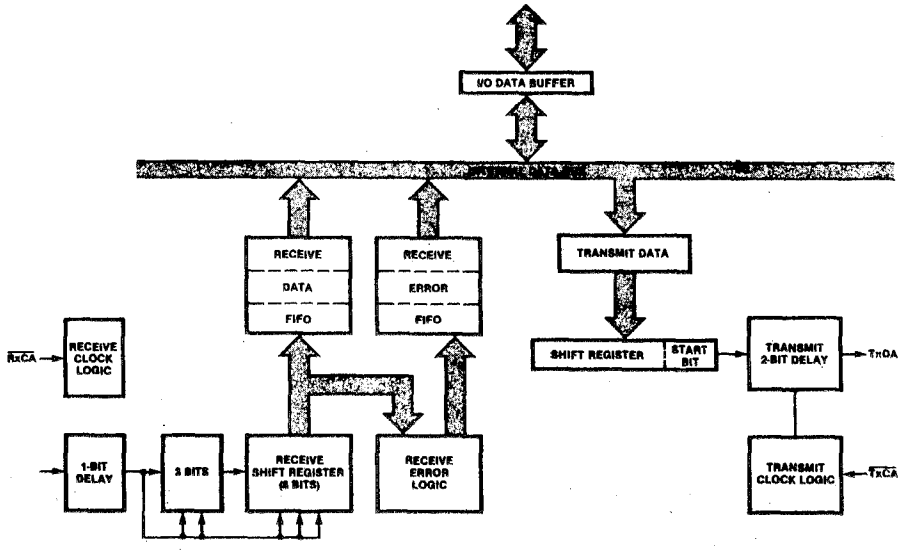


Figure 4. Data Path

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 ($\overline{M1}$ and $\overline{IORQ}$ both Low). The daisy-chained interrupt circuits determine the highest priority interrupt requestor. The IEI of the highest priority peripheral is terminated High. For any peripheral that has no interrupt pending or under service, IEO = IEI. Any peripheral that does have an interrupt pending or under service forces its IEO Low.

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

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

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

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

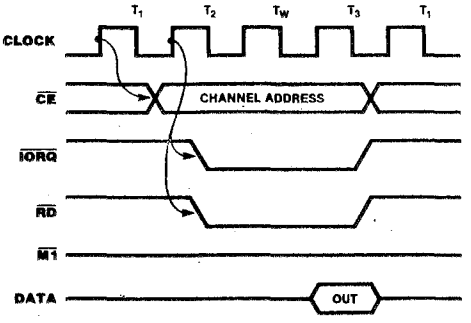


Figure 5a. Read Cycle

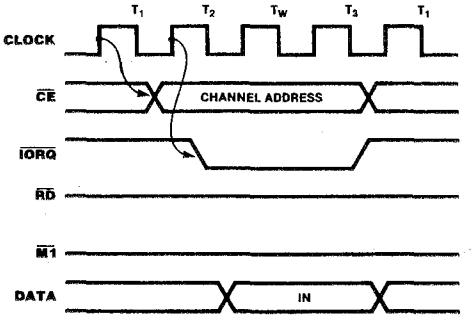


Figure 5b. Write Cycle

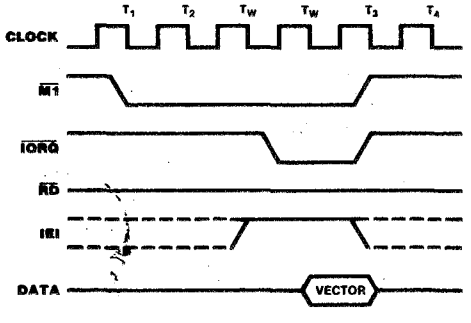


Figure 5c. Interrupt Acknowledge Cycle

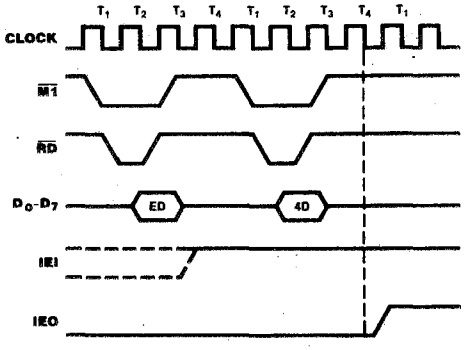


Figure 5d. Return from Interrupt Cycle

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.

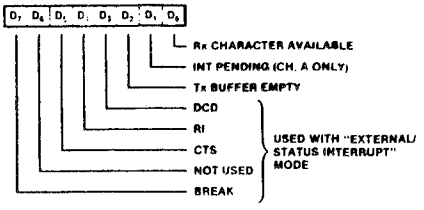
To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing an input instruction, the contents of the addressed read register can be read by the CPU.

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).

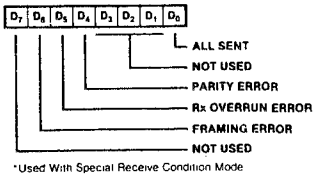
Read Register Functions	
RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)

Z-80 DART Read and Write Registers

READ REGISTER 0

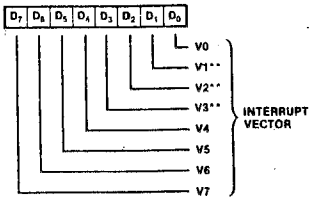


READ REGISTER 1*



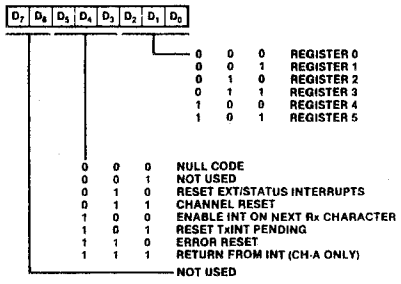
*Used With Special Receive Condition Mode

READ REGISTER 2

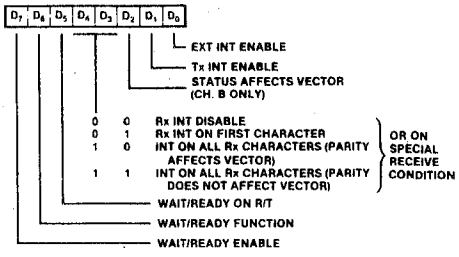


**Variable If Status Affects Vector Is Programmed

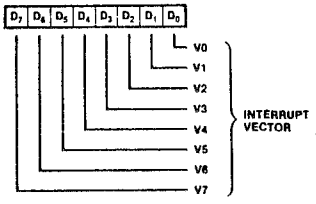
WRITE REGISTER 0



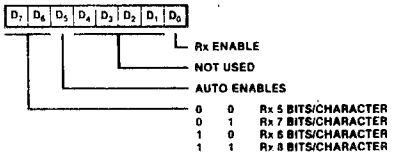
WRITE REGISTER 1



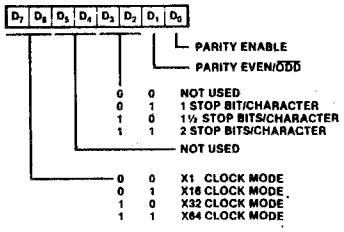
WRITE REGISTER 2 (CHANNEL B ONLY)



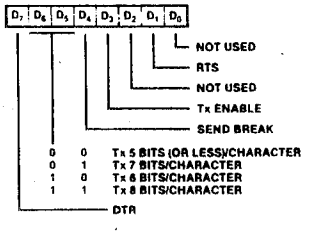
WRITE REGISTER 3



WRITE REGISTER 4



WRITE REGISTER 5



APPENDIX B

Z80A/Z80B PARALLEL INPUT/OUTPUT (PIO)

Z8420
Z80® PIO Parallel Input/Output Controller

Zilog

Product Specification

March 1981

Features

- Provides a direct interface between Z-80 microcomputer systems and peripheral devices.
- Both ports have interrupt-driven handshake for fast response.
- Four programmable operating modes: byte input, byte output, byte input/output (Port A only), and bit input/output.
- Programmable interrupts on peripheral status conditions.
- Standard Z-80 Family bus-request and prioritized interrupt-request daisy chains implemented without external logic.
- The eight Port B outputs can drive Darlington transistors (1.5 mA at 1.5 V).

General Description

The Z-80 PIO Parallel I/O Circuit is a programmable, dual-port device that provides a TTL-compatible interface between peripheral devices and the Z-80 CPU. The CPU configures the Z-80 PIO to interface with a wide range of peripheral devices with no other external logic. Typical peripheral devices that are compatible with the Z-80 PIO include most keyboards, paper tape readers and punches, printers, PROM programmers, etc.

One characteristic of the Z-80 peripheral controllers that separates them from other interface controllers is that all data transfer between the peripheral device and the CPU is accomplished under interrupt control. Thus, the interrupt logic of the PIO permits full use of the efficient interrupt capabilities of the Z-80 CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO.

Another feature of the PIO is the ability to interrupt the CPU upon occurrence of specified status conditions in the peripheral device. For example, the PIO can be programmed to interrupt if any specified peripheral alarm conditions should occur. This interrupt capability reduces the time the processor must spend in polling peripheral status.

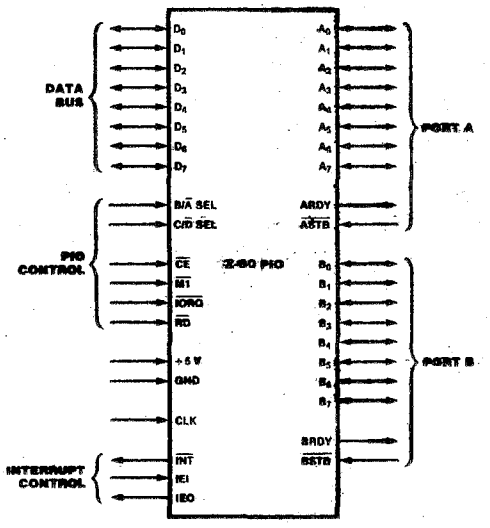


Figure 1. Pin Functions

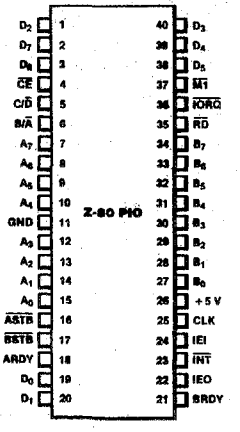


Figure 2. Pin Assignments

General Description (Continued)

The Z-80 PIO interfaces to peripherals via two independent general-purpose I/O ports, designated Port A and Port B. Each port has eight data bits and two handshake signals, Ready and Strobe, which control data transfer. The Ready output indicates to the peripheral that the port is ready for a data transfer. Strobe is an input from the peripheral that indicates when a data transfer has occurred.

Operating Modes. The Z-80 PIO ports can be programmed to operate in four modes: byte output (Mode 0), byte input (Mode 1), byte input/output (Mode 2) and bit input/output (Mode 3).

In Mode 0, either Port A or Port B can be programmed to output data. Both ports have output registers that are individually addressed by the CPU; data can be written to either port at any time. When data is written to a port, an active Ready output indicates to the external device that data is available at the associated port and is ready for transfer to the external device. After the data transfer, the external device responds with an active Strobe input, which generates an interrupt, if enabled.

In Mode 1, either Port A or Port B can be configured in the input mode. Each port has an input register addressed by the CPU. When the CPU reads data from a port, the PIO sets the Ready signal, which is detected by the external device. The external device then places data on the I/O lines and strobes the I/O port, which latches the data into the Port Input Register, resets Ready, and triggers the Interrupt Request, if enabled. The CPU can read the input data at any time, which again sets Ready.

Mode 2 is bidirectional and uses Port A, plus the interrupts and handshake signals from both ports. Port B must be set to Mode 3 and masked off. In operation, Port A is used for both data input and output. Output operation is similar to Mode 0 except that data is allowed out onto the Port A bus only when $\overline{ASTB}$ is Low. For input, operation is similar to Mode 1, except that the data input uses the Port B handshake signals and the Port B interrupt (if enabled).

Both ports can be used in Mode 3. In this mode, the individual bits are defined as either input or output bits. This provides up to eight separate, individually defined bits for each port. During operation, Ready and Strobe are not used. Instead, an interrupt is generated if the condition of one input changes, or if all inputs change. The requirements for generating an interrupt are defined during the programming operation; the active level is specified as either High or Low, and the logic condition is specified as either one input active (OR) or all inputs active (AND). For example, if the port is programmed for active Low inputs and the logic function is AND, then all inputs at the specified port must go Low to generate an interrupt.

Data outputs are controlled by the CPU and can be written or changed at any time.

- Individual bits can be masked off.
- The handshake signals are not used in Mode 3; Ready is held Low, and Strobe is disabled.
- When using the Z-80 PIO interrupts, the Z-80 CPU interrupt mode must be set to Mode 2.

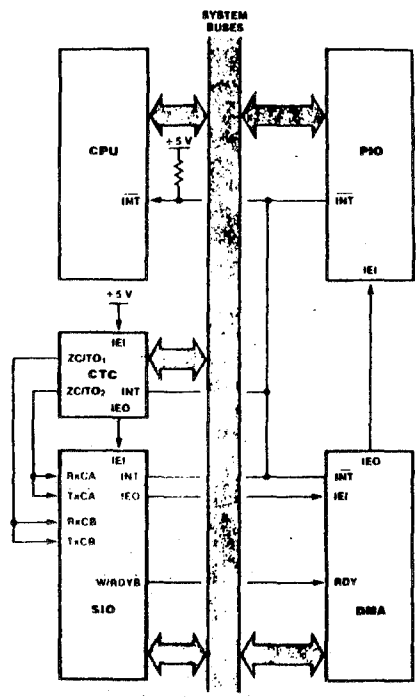


Figure 3. PIO in a Typical Z80 Family Environment

Internal Structure

The internal structure of the Z-80 PIO consists of a Z-80 CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic (Figure 4). The CPU bus interface logic allows the Z-80 PIO to interface directly to the Z-80 CPU with no other external logic. The internal control logic synchronizes the CPU data bus to the peripheral device interfaces (Port A and Port B). The two I/O ports (A and B) are virtually identical and are used to interface directly to peripheral devices.

Port Logic. Each port contains separate input and output registers, handshake control logic, and the control registers shown in Figure 5. All data transfers between the peripheral unit and the CPU use the data input and output registers. The handshake logic associated with each port controls the data transfers through the input and the output registers. The mode control register (two bits) selects one of the four programmable operating modes.

The control mode (Mode 3) uses the remaining registers. The input/output control register specifies which of the eight data bits in the port are to be outputs and enables these bits; the remaining bits are inputs. The mask register and the mask control register control Mode 3 interrupt conditions. The mask register specifies which of the bits in the port are active and which are masked or inactive.

The mask control register specifies two conditions: first, whether the active state of the input bits is High or Low, and second, whether an interrupt is generated when any *one* unmasked input bit is active (OR condition) or if the interrupt is generated when *all* unmasked input bits are active (AND condition).

Interrupt Control Logic. The interrupt control logic section handles all CPU interrupt protocol for nested-priority interrupt structures. Any device's physical location in a daisy-chain configuration determines its priority. Two lines (IEI and IEO) are provided in each PIO to form this daisy chain. The device closest to the CPU has the highest priority. Within a PIO, Port A interrupts have higher priority than those of Port B. In the byte input, byte output, or bidirectional modes, an interrupt can be generated whenever the peripheral requests a new byte transfer. In the bit control mode, an interrupt can be generated when the peripheral status matches a programmed value. The PIO provides for complete control of nested interrupts. That is, lower priority devices may not interrupt higher priority devices that have not had their interrupt service routines completed by the CPU. Higher priority devices may interrupt the servicing of lower priority devices.

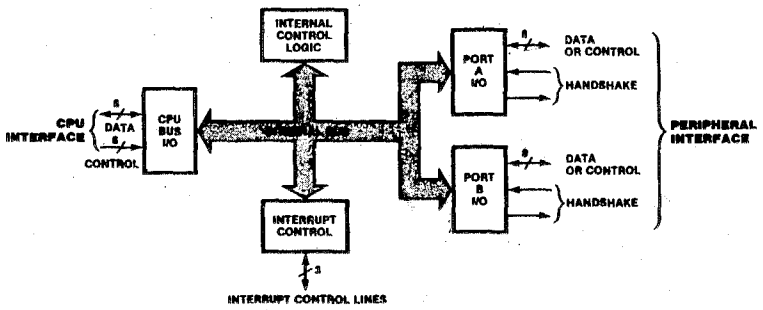


Figure 4. Block Diagram

Internal Structure (Continued)

If the CPU (in interrupt Mode 2) accepts an interrupt, the interrupting device must provide an 8-bit interrupt vector for the CPU. This vector forms a pointer to a location in memory where the address of the interrupt service routine is located. The 8-bit vector from the interrupting device forms the least significant eight bits of the indirect pointer while the I Register in the CPU provides the most significant eight bits of the pointer. Each port (A and B) has an independent interrupt vector. The least significant bit of the vector is automatically set to 0 within the PIO because the pointer must point to two adjacent memory locations for a complete 16-bit address.

Unlike the other Z-80 peripherals, the PIO does not enable interrupts immediately after programming. It waits until $\overline{M1}$ goes Low (e.g., during an opcode fetch). This condition is unimportant in the Z-80 environment but might not be if another type of CPU is used.

The PIO decodes the RETI (Return From Interrupt) instruction directly from the CPU data bus so that each PIO in the system knows at all times whether it is being serviced by the CPU interrupt service routine. No other communication with the CPU is required.

CPU Bus I/O Logic. The CPU bus interface logic interfaces the Z-80 PIO directly to the Z-80 CPU, so no external logic is necessary. For large systems, however, address decoders and/or buffers may be necessary.

Internal Control Logic. This logic receives the control words for each port during programming and, in turn, controls the operating functions of the Z-80 PIO. The control logic synchronizes the port operations, controls the port mode, port addressing, selects the read/write function, and issues appropriate commands to the ports and the interrupt logic. The Z-80 PIO does not receive a write input from the CPU; instead, the $\overline{RD}$, $\overline{CE}$, C/$\overline{D}$ and $\overline{IORQ}$ signals generate the write input internally.

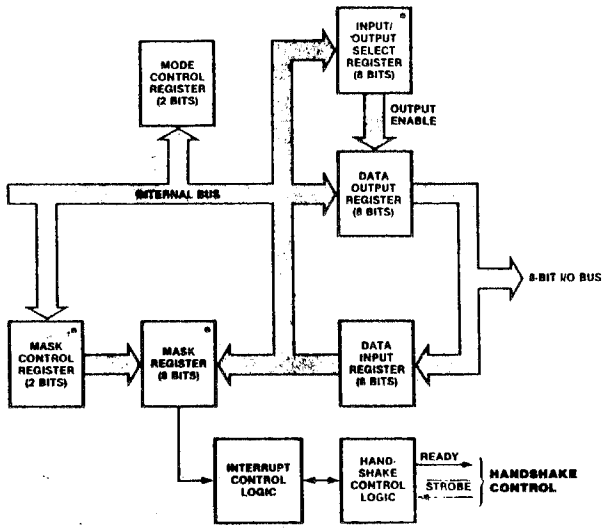


Figure 5. Typical Port I/O Block Diagram

Programming

Mode 0, 1, or 2. *(Byte Input, Output, or Bidirectional).* Programming a port for Mode 0, 1, or 2 requires two words per port. These words are:

A Mode Control Word. Selects the port operating mode (Figure 6). This word may be written any time.

An Interrupt Vector. The Z-80 PIO is designed for use with the Z-80 CPU in interrupt Mode 2 (Figure 7). When interrupts are enabled, the PIO must provide an interrupt vector.

Mode 3. *(Bit Input/Output).* Programming a port for Mode 3 operation requires a control word, a vector (if interrupts are enabled), and three additional words, described as follows:

I/O Register Control. When Mode 3 is selected, the mode control word must be followed by another control word that sets the I/O control register, which in turn defines which port lines are inputs and which are outputs (Figure 8).

Interrupt Control Word. In Mode 3, handshake is not used. Interrupts are generated as a logic function of the input signal levels. The interrupt control word sets the logic conditions and the logic levels required for generating an interrupt. Two logic conditions or functions are available: AND (if all input bits change to the active level, an interrupt is triggered), and OR (if any one of the input bits changes to the active level, an interrupt is triggered). Bit D_6 sets the logic function, as shown in Figure 9. The active level of the input bits can be set either High or Low. The active level is controlled by Bit D_5.

Mask Control Word. This word sets the mask control register, allowing any unused bits to be masked off. If any bits are to be masked, then D_4 must be set. When D_4 is set, the next word written to the port must be a mask control word (Figure 10).

Interrupt Disable. There is one other control word which can be used to enable or disable a port interrupt. It can be used without changing the rest of the interrupt control word (Figure 11).

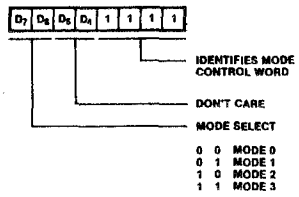


Figure 6. Mode Control Word

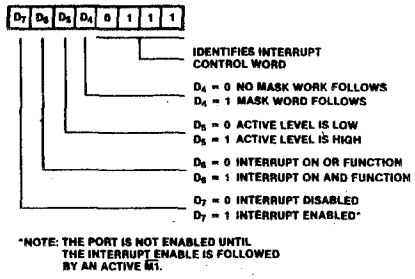


Figure 9. Interrupt Control Word

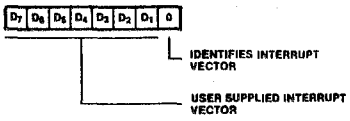


Figure 7. Interrupt Vector Word

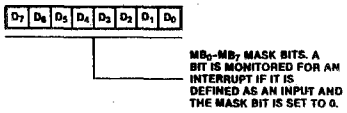


Figure 10. Mask Control Word

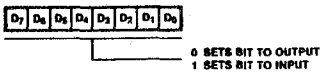


Figure 8. I/O Register Control Word

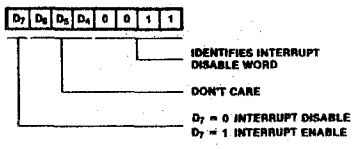


Figure 11. Interrupt Disable Word

Pin Description

A_0-A_7. *Port A Bus* (bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port A of the PIO and a peripheral device. A_0 is the least significant bit of the Port A data bus.

ARDY. *Register A Ready* (output, active High). The meaning of this signal depends on the mode of operation selected for Port A as follows:

Output Mode. This signal goes active to indicate that the Port A output register has been loaded and the peripheral data bus is stable and ready for transfer to the peripheral device.

Input Mode. This signal is active when the Port A input register is empty and ready to accept data from the peripheral device.

Bidirectional Mode. This signal is active when data is available in the Port A output register for transfer to the peripheral device. In this mode, data is not placed on the Port A data bus, unless $\overline{ASTB}$ is active.

Control Mode. This signal is disabled and forced to a Low state.

$\overline{ASTB}$. *Port A Strobe Pulse From Peripheral Device* (input, active Low). The meaning of this signal depends on the mode of operation selected for Port A as follows:

Output Mode. The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the PIO.

Input Mode. The strobe is issued by the peripheral to load data from the peripheral into the Port A input register. Data is loaded into the PIO when this signal is active.

Bidirectional Mode. When this signal is active, data from the Port A output register is gated onto the Port A bidirectional data bus. The positive edge of the strobe acknowledges the receipt of the data.

Control Mode. The strobe is inhibited internally.

B_0-B_7. *Port B Bus* (bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port B and a peripheral device. The Port B data bus can supply 1.5 mA at 1.5 V to drive Darlington transistors. B_0 is the least significant bit of the bus.

B/$\overline{A}$. *Port B Or A Select* (input, High = B). This pin defines which port is accessed during a data transfer between the CPU and the PIO. A Low on this pin selects Port A; a High selects Port B. Often address bit A_0 from the CPU is used for this selection function.

BRDY. *Register B Ready* (output, active High). This signal is similar to ARDY, except that in the Port A bidirectional mode this signal is High when the Port A input register is empty and ready to accept data from the peripheral device.

$\overline{BSTB}$. *Port B Strobe Pulse From Peripheral Device* (input, active Low). This signal is similar to $\overline{ASTB}$, except that in the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.

C/$\overline{D}$. *Control Or Data Select* (input, High = C). This pin defines the type of data transfer to be performed between the CPU and the PIO. A High on this pin during a CPU write to the PIO causes the Z-80 data bus to be interpreted as a *command* for the port selected by the B/$\overline{A}$ Select line. A Low on this pin means that the Z-80 data bus is being used to transfer data between the CPU and the PIO. Often address bit A_1 from the CPU is used for this function.

$\overline{CE}$. *Chip Enable* (input, active Low). A Low on this pin enables the PIO to accept command or data inputs from the CPU during a write cycle or to transmit data to the CPU during a read cycle. This signal is generally decoded from four I/O port numbers for Ports A and B, data, and control.

CLK. *System Clock* (input). The Z-80 PIO uses the standard single-phase Z-80 system clock.

D_0-D_7. *Z-80 CPU Data Bus* (bidirectional, 3-state). This bus is used to transfer all data and commands between the Z-80 CPU and the Z-80 PIO. D_0 is the least significant bit.

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

IEO. *Interrupt Enable Out* (output, active High). The IEO signal is the other signal required to form a daisy chain priority scheme. It is High only if IEI is High and the CPU is not servicing an interrupt from this PIO. Thus this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

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

$\overline{IORQ}$. *Input/Output Request* (input from Z-80 CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{RD}$ to transfer commands and data between the Z-80 CPU and the Z-80 PIO. When $\overline{CE}$, $\overline{RD}$, and $\overline{IORQ}$ are active, the port addressed by B/$\overline{A}$ transfers data to the CPU (a read operation). Conversely, when $\overline{CE}$ and $\overline{IORQ}$ are active but $\overline{RD}$ is not, the port addressed by B/$\overline{A}$ is written into from the CPU with either data or control information, as specified by C/$\overline{D}$. Also, if $\overline{IORQ}$ and $\overline{M1}$ are active simultaneously, the CPU is acknowledging an interrupt; the interrupting port automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

Pin Description (Continued)

$\overline{M1}$. *Machine Cycle* (input from CPU, active Low). This signal is used as a sync pulse to control several internal PIO operations. When both the $\overline{M1}$ and $\overline{RD}$ signals are active, the Z-80 CPU is fetching an instruction from memory. Conversely, when both $\overline{M1}$ and $\overline{IORQ}$ are active, the CPU is acknowledging an interrupt. In addition, $\overline{M1}$ has two other functions within the Z-80 PIO: it synchronizes the PIO interrupt logic; when $\overline{M1}$ occurs without an active $\overline{RD}$ or $\overline{IORQ}$ signal, the PIO is reset.

$\overline{RD}$. *Read Cycle Status* (input from Z-80 CPU, active Low). If $\overline{RD}$ is active, or an I/O operation is in progress, $\overline{RD}$ is used with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{IORQ}$ to transfer data from the Z-80 PIO to the Z-80 CPU.

Timing

The following timing diagrams show typical timing in a Z-80 CPU environment. For more precise specifications refer to the composite ac timing diagram.

Write Cycle. Figure 12 illustrates the timing for programming the Z-80 PIO or for writing data to one of its ports. No Wait states are allowed for writing to the PIO other than the automatically inserted T_{WA}. The PIO does not receive a specific write signal; it internally generates its own from the lack of an active $\overline{RD}$ signal.

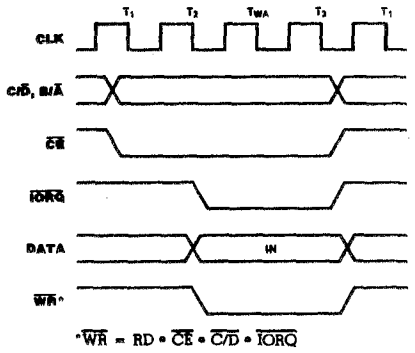


Figure 12. Write Cycle Timing

Read Cycle. Figure 13 illustrates the timing for reading the data input from an external device to one of the Z-80 PIO ports. No Wait states are allowed for reading the PIO other than the automatically inserted T_{WA}.

Output Mode (Mode 0). An output cycle (Figure 14) is always started by the execution of an output instruction by the CPU. The $\overline{WR}$* pulse from the CPU latches the data from the CPU data bus into the selected port's output register. The $\overline{WR}$* pulse sets the Ready flag after a Low-going edge of CLK, indicating data is available. Ready stays active until the positive edge of the strobe line is received, indicating that data was taken by the peripheral. The positive edge of the strobe pulse generates an $\overline{INT}$ if the interrupt enable flip-flop has been set and if this device has the highest priority.

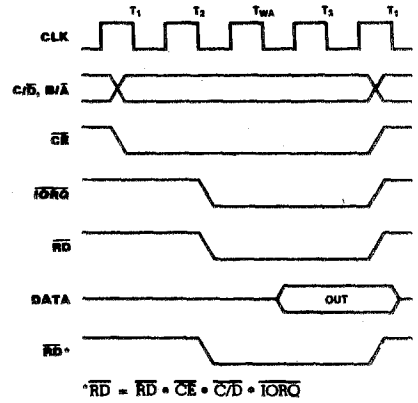


Figure 13. Read Cycle Timing

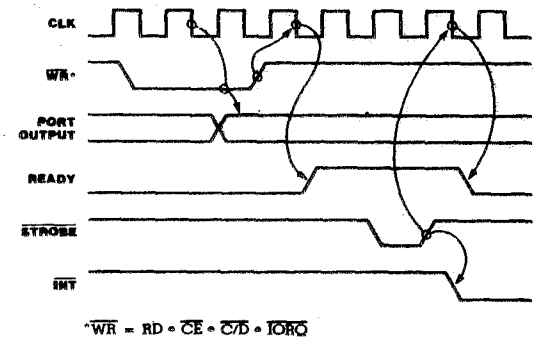


Figure 14. Mode 0 Output Timing

Timing (Continued)

Input Mode (Mode 1). When $\overline{\text{STROBE}}$ goes Low, data is loaded into the selected port input register (Figure 15). The next rising edge of strobe activates $\overline{\text{INT}}$, if Interrupt Enable is set and this is the highest-priority requesting device. The following falling edge of CLK resets Ready to an inactive state, indicating that the input register is full and cannot accept any more data until the CPU completes a read. When a read is complete, the positive edge of $\overline{\text{RD}}$ sets Ready at the next Low-going transition of CLK. At this time new data can be loaded into the PIO.

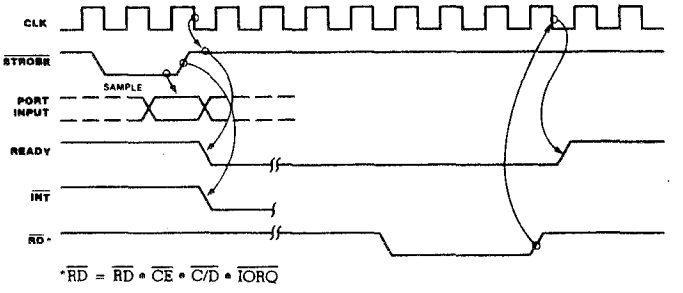


Figure 15. Mode 1 Input Timing

Bidirectional Mode (Mode 2). This is a combination of Modes 0 and 1 using all four handshake lines and the eight Port A I/O lines (Figure 16). Port B must be set to the bit mode and its inputs must be masked. The Port A handshake lines are used for output control and the Port B lines are used for input control. If interrupts occur, Port A's vector will be used during port output and Port B's will be used during port input. Data is allowed out onto the Port A bus only when $\overline{\text{ASTB}}$ is Low. The rising edge of this strobe can be used to latch the data into the peripheral.

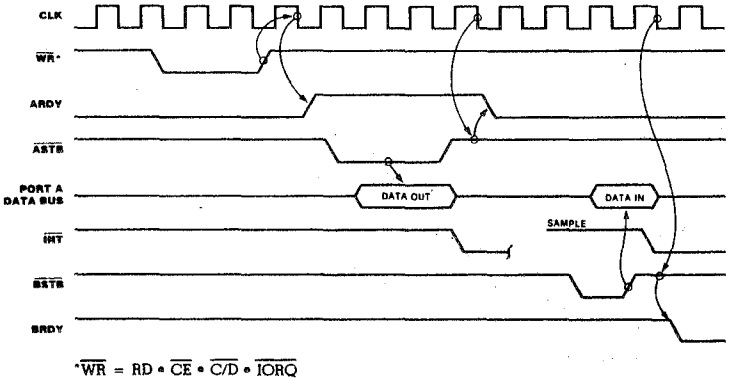


Figure 16. Mode 2 Bidirectional Timing

Timing (Continued)

Bit Mode (Mode 3). The bit mode does not utilize the handshake signals, and a normal port write or port read can be executed at any time. When writing, the data is latched into the output registers with the same timing as the output mode (Figure 17).

When reading the PIO, the data returned to the CPU is composed of output register data from those port data lines assigned as outputs and input register data from those port data lines assigned as inputs. The input register contains data that was present immediately prior to the falling edge of $\overline{RD}$. An interrupt is generated if interrupts from the port are enabled and the data on the port data lines satisfy the logical equation defined by the 8-bit mask and 2-bit mask control registers. However, if Port A is programmed in bidirectional mode, Port B does not issue an interrupt in bit mode and must therefore be polled.

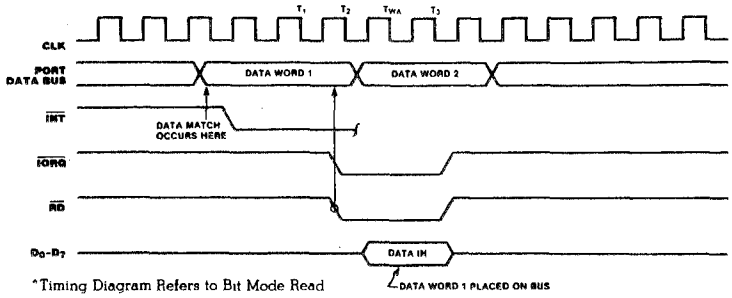


Figure 17. Mode 3 Bit Mode Timing

Interrupt Acknowledge Timing. During $\overline{M1}$ time, peripheral controllers are inhibited from changing their interrupt enable status, permitting the Interrupt Enable signal to ripple through the daisy chain. The peripheral with IEI High and IEO Low during $\overline{INTACK}$ places a preprogrammed 8-bit interrupt vector on the data bus at this time (Figure 18). IEO is held Low until a Return From Interrupt (RETI) instruction is executed by the CPU while IEI is High. The 2-byte RETI instruction is decoded internally by the PIO for this purpose.

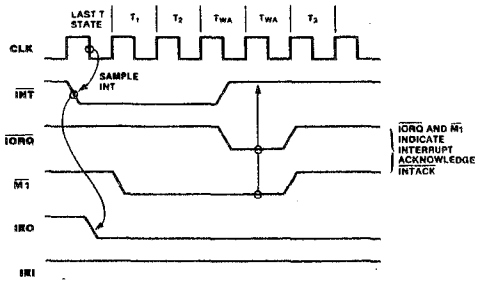


Figure 18. Interrupt Acknowledge Timing

Return From Interrupt Cycle. If a Z-80 peripheral has no interrupt pending and is not under service, then its IEO = IEI. If it has an interrupt under service (i.e., it has already interrupted and received an interrupt acknowledge) then its IEO is always Low, inhibiting lower priority devices from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO is Low unless an "ED" is decoded as the first byte of a 2-byte opcode (Figure 19). In this case, IEO goes High until the next opcode byte is decoded, whereupon it goes Low again. If the second byte of the opcode was a "4D," then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service has its IEI High and its IEO Low. This device is the highest-priority device in the daisy chain that has received an interrupt acknowledge. All other peripherals have IEI = IEO. If the next opcode byte decoded is "4D," this peripheral device resets its "interrupt under service" condition.

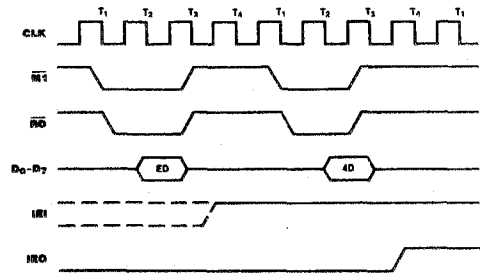
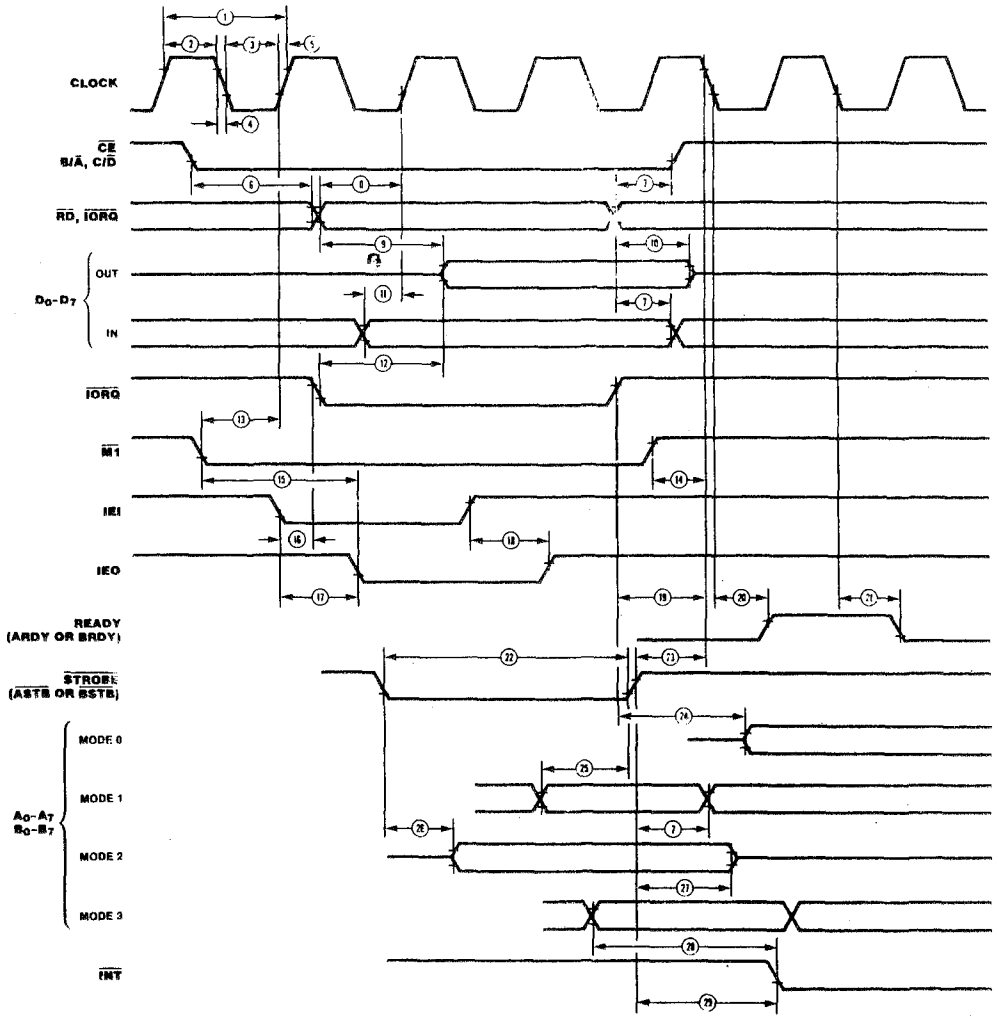


Figure 19. Return From Interrupt

AC Characteristics



Number	Symbol	Parameter	Z-80 PIO Min (ns)	Z-80 PIO Max (ns)	Z-80A PIO Min (ns)	Z-80A PIO Max (ns)	Z-80B PIO[9] Min (ns)	Z-80B PIO[9] Max (ns)	Comment
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	TsCS(RI)	$\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ to $\overline{RD}$, $\overline{IORQ}$ ↓ Setup Time	50		50		50		[6]
7	Th	Any Hold Times for Specified Setup Time	0		0		0	0	
8	TsRI(C)	$\overline{RD}$, $\overline{IORQ}$ to Clock ↑ Setup Time	115		115		70		
9	TdRI(DO)	$\overline{RD}$, $\overline{IORQ}$ ↓ to Data Out Delay		430		380		300	[2]
10	TdRI(DOs)	$\overline{RD}$, $\overline{IORQ}$ ↑ to Data Out Float Delay		160		110		70	
11	TsDI(C)	Data In to Clock ↑ Setup Time	50		50		40		CL = 50 pF
12	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)	340		160		120		[3]
13	TsM1(Cr)	$\overline{M1}$ ↓ to Clock ↑ Setup Time	210		90		70		
14	TsM1(Cf)	$\overline{M1}$ ↑ to Clock ↓ Setup Time ($\overline{M1}$ Cycle)	0		0		0		[8]
15	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt Immediately Preceding $\overline{M1}$ ↓)		300		190		100	[5, 7]
16	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		140		100		[7]
17	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		190		130		120	[5] CL = 50 pF
18	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED Decode)		210		160		160	[5]
19	TcIO(C)	$\overline{IORQ}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)	220		200		170		
20	TdC(RDYr)	Clock ↓ to READY ↑ Delay	200		190		170		[5] CL = 50 pF
21	TdC(RDYf)	Clock ↓ to READY ↑ Delay	150		140		120		[5]
22	TwSTB	$\overline{STROBE}$ Pulse Width	150		150		120		[4]
23	TsSTB(C)	$\overline{STROBE}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)	220		220		150		[5]
24	TdIO(PD)	$\overline{IORQ}$ ↑ to PORT DATA Stable Delay (Mode 0)		200		180		160	[5]
25	TsPD(STB)	PORT DATA to $\overline{STROBE}$ ↑ Setup Time (Mode 1)	260		230		190		
26	TdSTB(PD)	$\overline{STROBE}$ ↓ to PORT DATA Stable (Mode 2)		230		210		180	[5]
27	TdSTB(PDr)	$\overline{STROBE}$ ↑ to PORT DATA Float Delay (Mode 2)		200		180		160	CL = 50 pF
28	TdPD(INT)	PORT DATA Match to $\overline{INT}$ ↓ Delay (Mode 3)		540		490		430	
29	TdSTB(INT)	$\overline{STROBE}$ ↑ to $\overline{INT}$ ↓ Delay		490		440		350	

NOTES:
[1] TcC = TwCh + TwCl + TrC + TfC.
[2] Increase TdRI(DO) by 10 ns for each 50 pF increase in load up to 200 pF max.
[3] Increase TdIO(DOI) by 10 ns for each 50 pF, increase in loading up to 200 pF max.
[4] For Mode 2: TwSTB > TsPD(STB).
[5] Increase these values by 2 ns for each 10 pF increase in loading up to 100 pF max.
[6] TsCS(RI) may be reduced. However, the time subtracted from TsCS(RI) will be added to TdRI(DO).
[7] 2.5 TcC > (N-2)TdIEI(IEOf) + TdM1(IEO) + TsIEI(IO) + TTL Buffer Delay, if any.
[8] $\overline{M1}$ must be active for a minimum of two clock cycles to reset the PIO.
[9] Z80B PIO numbers are preliminary and subject to change.

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.

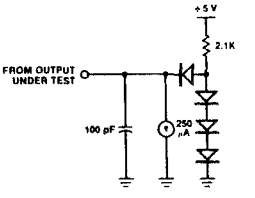
Test Conditions

The characteristics below apply for the following 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:

- 0° to +70°C, $+4.75\ V \le V_{CC} \le +5.25\ V$
- -40°C to +85°C, $+4.75\ V \le V_{CC} \le +5.25\ V$
- -55° to +125°C, $+4.75\ V \le V_{CC} \le +5.5\ V$

The product number for each operating temperature range may be found in the Ordering Information section.

All ac parameters assume a load capacitance of 100 pF max. Timing references between two output signals assume a load difference of 50 pF max.



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_{LI}	Input Leakage Current	-10.0	+10.0	µA	$0 < V_{IN} < V_{CC}$
I_Z	3-State Output/Data Bus Input Leakage Current	-10.0	+10.0	µA	$0 < V_{IN} < V_{CC}$
I_{CC}	Power Supply Current		100.0	mA	V_{OH} = 1.5V
I_{OHD}	Darlington Drive Current	-1.5	3.8	mA	R_{EXT} = 390 Ω

Over specified temperature and voltage range.

Capacitance

Symbol	Parameter	Min	Max	Unit	Test Condition
C	Clock Capacitance		10	pF	Unmeasured pins returned to ground
C_{IN}	Input Capacitance		5	pF	
C_{OUT}	Output Capacitance		10	pF	

Over specified temperature range; f = 1MHz

APPENDIX C

Z80A/Z80B COUNTER/TIMER CIRCUIT (CTC)

Z8430
Z80® CTC Counter/ Timer Circuit

Zilog

Product Specification

March 1981

Features

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

General Description

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

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

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

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

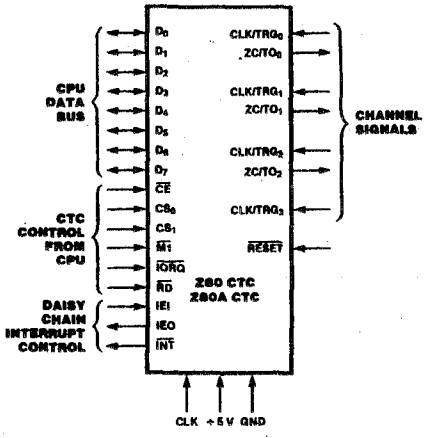


Figure 1. Pin Functions

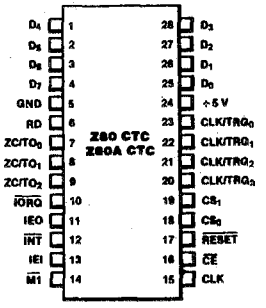


Figure 2. Pin Assignments

Functional Description

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

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

The timer mode determines time intervals as small as 4 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 ($\overline{INT}$), 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

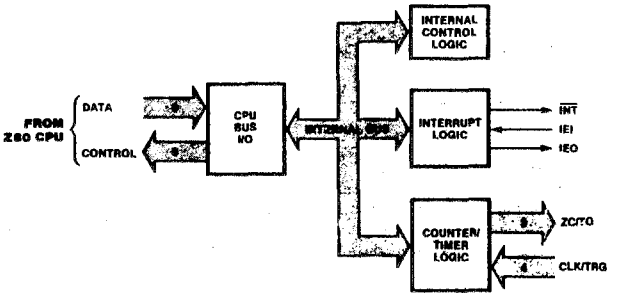


Figure 3. Functional Block Diagram

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 $\overline{INT}$ signal to the Z-80 CPU. When the Z-80 CPU responds with interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$), then the interrupt logic arbitrates the CTC internal priorities, and the interrupt control logic places a unique interrupt vector on the data bus.

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

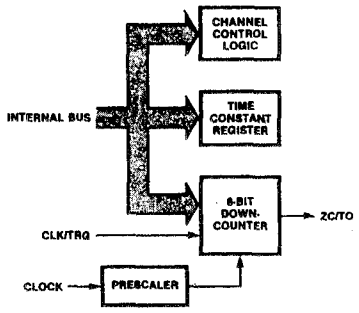


Figure 4. Counter/Timer Block Diagram

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

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

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

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

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

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

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

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

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 0 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.

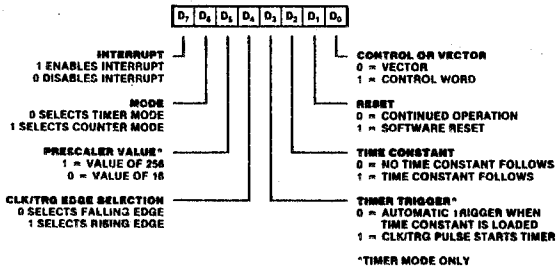


Figure 5. Channel Control Word

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.

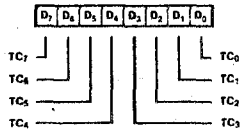


Figure 6. Time Constant Word

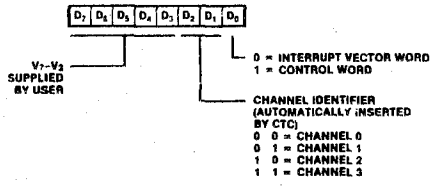


Figure 7. Interrupt Vector Word

Pin Description

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

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

CLK/TRG_0-CLK/TRG_3. *External Clock/Timer Trigger* (input, user-selectable active High or Low). Four pins corresponding to the four Z-80 CTC channels. In counter mode, every active edge on this pin decrements the down-counter. In timer mode, an active edge starts the timer.

CS_0-CS_1. *Channel Select* (inputs active High). Two-bit binary address code selects one of the four CTC channels for an I/O write or read (usually connected to A_0 and A_1).

D_0-D_7. *System Data Bus* (bidirectional, 3-state). Transfers all data and commands between the Z-80 CPU and the Z-80 CTC.

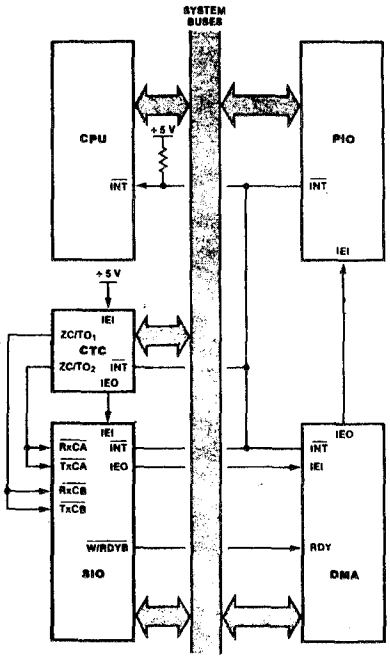


Figure 8. A Typical Z-80 Environment

$\overline{IEI}$. *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 an Z-80 CTC channel. IEO blocks lower priority devices from interrupting while a higher priority interrupting device is being serviced.

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

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

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

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

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

ZC/TO_0-ZC/TO_2. *Zero Count/Timeout* (output, active High). Three ZC/TO pins corresponding to Z-80 CTC channels 2 through 0 (Channel 3 has no ZC/TO pin). In both counter and timer modes the output is an active High pulse when the down-counter decrements to zero.

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.*

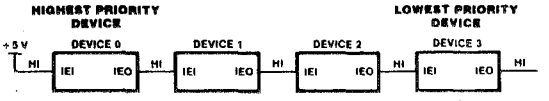


Figure 13. Daisy-Chain Interrupt Priorities

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.

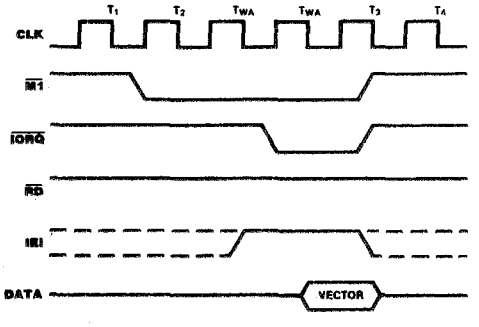


Figure 14. Interrupt Acknowledge Timing

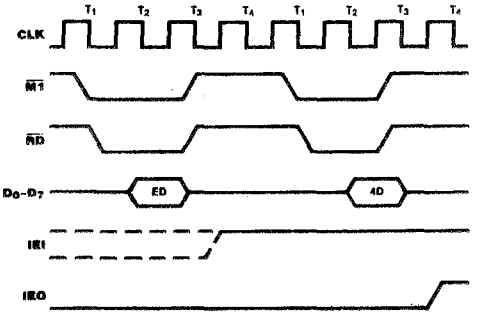


Figure 15. Return From Interrupt Timing

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.

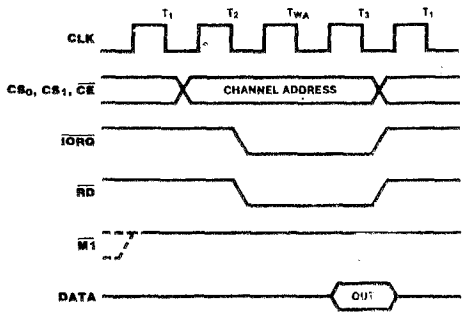


Figure 9. Read Cycle Timing

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_{WA}. No additional wait states are allowed.

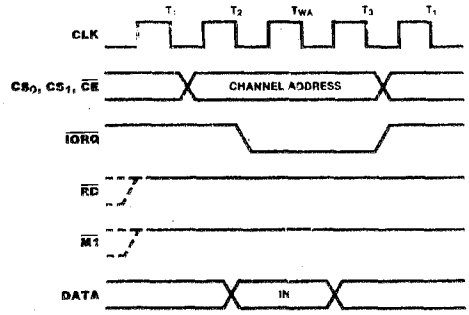


Figure 10. Write Cycle Timing

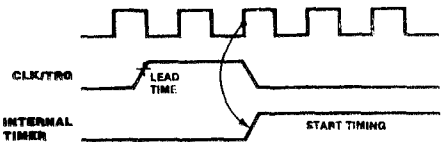


Figure 11. Timer Mode Timing

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.

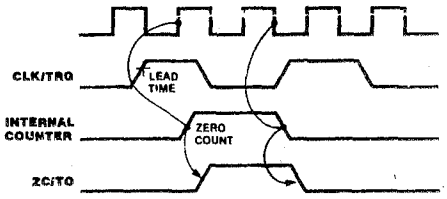


Figure 12. Counter Mode Timing

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.

APPENDIX D

Z80A/Z80B CENTRAL PROCESSING UNIT (CPU)

Z8400
Z80® CPU Central Processing Unit

Zilog

Product Specification

March 1981

Features

- The instruction set contains 158 instructions. The 78 instructions of the 8080A are included as a subset; 8080A software compatibility is maintained.
- Six MHz, 4 MHz and 2.5 MHz clocks for the 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 compatible, non-Z80 peripheral device, and Z80 Family peripheral with or without daisy chain.
- On-chip dynamic memory refresh counter.

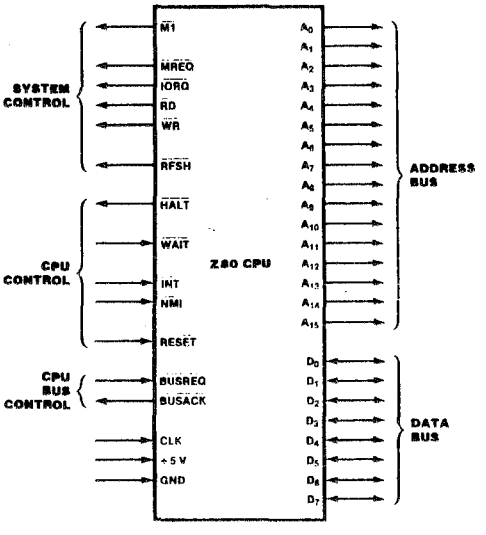


Figure 1. Pin Functions

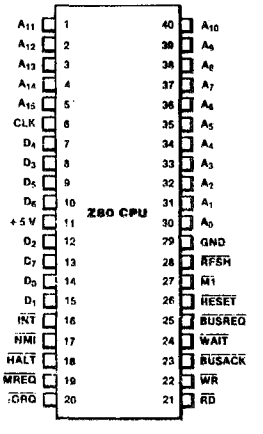


Figure 2. Pin Assignments

General Description

The Z80, Z80A, and Z80B 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 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.

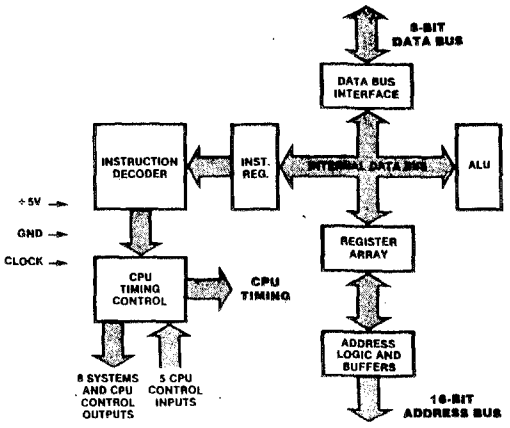


Figure 3. Z80 CPU Block Diagram

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-Synch 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 interfupt mode at any particular time. Table 1 provides further information on these registers.

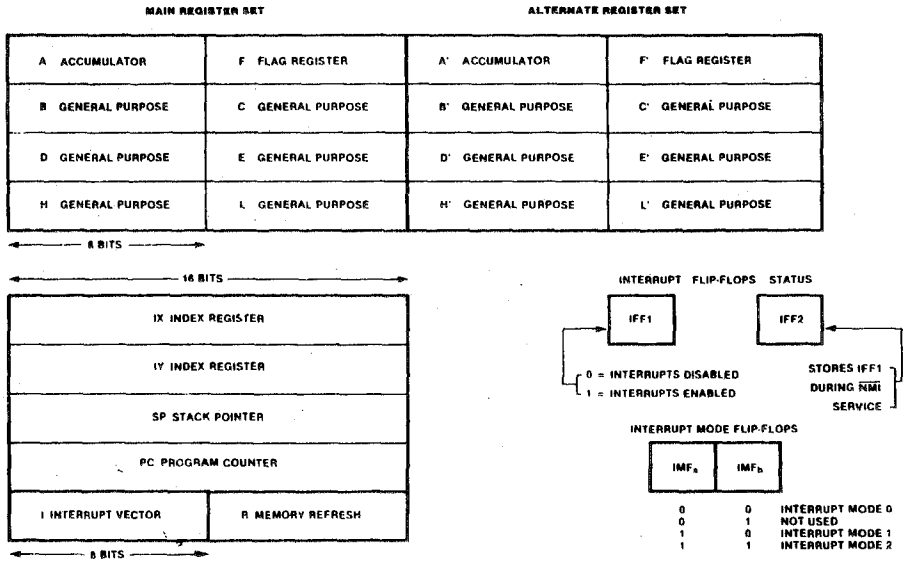


Figure 4. CPU Registers

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. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Same as IX, above.
SP	Stack Pointer	16	Stores addresses or data temporarily. 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 since it requires that interrupts be enabled in software in order to operate. Either $\overline{NMI}$ or $\overline{INT}$ 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 — compatible with 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 at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

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 a 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 and to place the interrupt vector on the bus.

Mode 0 Interrupt Operation. This mode is compatible with the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus, which is then acted on six times by the CPU. This is normally a Restart Instruction, which will initiate an unconditional jump to the selected one of eight restart locations in page zero of memory.

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 vector address 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 address vector on the data bus during the interrupt acknowledge cycle. The high-order byte of the interrupt service routine address is supplied by the I (Interrupt) register. 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 hardwared to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

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

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

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

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

Table 2. State of Flip-Flops

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	•	•	•	01 r r'		1	1	4	r, r' Reg.
LD r, n	r ← n	•	•	X	•	•	•	00 r 110		2	2	7	000 B
								— n —					001 C
LD r, (HL)	r ← (HL)	•	•	X	•	•	•	01 r 110		1	2	7	010 D
LD r, (IX+d)	r ← (IX+d)	•	•	X	•	•	•	11 011 101	DD	3	5	19	011 E
								01 r 101					100 H
								— d —					101 L
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	•	•	11 111 101	FD	3	5	19	111 A
								01 r 110					
								— d —					
LD (HL), r	(HL) ← r	•	•	X	•	•	•	01 110 r		1	2	7	
LD (IX+d), r	(IX+d) ← r	•	•	X	•	•	•	11 011 101	DD	3	5	19	
								01 110 r					
								— d —					
LD (IY+d), r	(IY+d) ← r	•	•	X	•	•	•	11 111 101	FD	3	5	19	
								01 110 r					
								— d —					
LD (HL), n	(HL) ← n	•	•	X	•	•	•	00 110 110	36	2	3	10	
								— n —					
LD (IX+d), n	(IX+d) ← n	•	•	X	•	•	•	11 011 101	DD	4	5	19	
								00 110 110	36				
								— d —					
								— n —					
LD (IY+d), n	(IY+d) ← n	•	•	X	•	•	•	11 111 101	FD	4	5	19	
								00 110 110	36				
								— d —					
								— n —					
LD A, (BC)	A ← (BC)	•	•	X	•	•	•	00 001 010	0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	•	•	•	00 011 010	1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	•	•	•	00 111 010	3A	3	4	13	
								— n —					
								— n —					
LD (BC), A	(BC) ← A	•	•	X	•	•	•	00 000 010	02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	•	•	•	00 010 010	12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	•	•	•	00 110 010	32	3	4	13	
								— n —					
								— n —					
LD A, I	A ← I	↕	↕	X	0	X	IFF	0 • 11 101 101	ED	2	2	9	
								01 010 111	57				
LD A, R	A ← R	↕	↕	X	0	X	IFF	0 • 11 101 101	ED	2	2	9	
								01 011 111	5F				
LD I, A	I ← A	•	•	X	•	X	•	• • 11 101 101	ED	2	2	9	
								01 000 111	47				
LD R, A	R ← A	•	•	X	•	X	•	• • 11 101 101	ED	2	2	9	
								01 001 111	4F				

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

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

Exchange, Block Transfer, Block Search Groups

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
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 DD 11 100 011 E3	2	6	23	
EX (SP), IY	IY_H ← (SP + 1) IY_L ← (SP)	•	•	X	X	•	•	11 111 101 FD 11 100 011 E3	2	6	23	
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC – 1	•	•	X	①	0	•	11 101 101 ED 10 100 000 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	0	•	11 101 101 ED 10 110 000 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
					①							
LDD	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1	*	*	X 0	X ↕	0	*	11 101 101 ED 10 101 000 A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1 Repeat until BC = 0	*	*	X 0	X 0	0	*	11 101 101 ED 10 111 000 B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
			②		①							
CPI	A − (HL) HL ← HL + 1 BC ← BC − 1	↕	↕	X ↕	X ↕	1	*	11 101 101 ED 10 100 001 A1	2	4	16	
			②		①							
CPIR	A − (HL) HL ← HL + 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0	↕	↕	X ↕	X ↕	1	*	11 101 101 ED 10 110 001 B1	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
			②		①							
CPD	A − (HL) HL ← HL − 1 BC ← BC − 1	↕	↕	X ↕	X ↕	1	*	11 101 101 ED 10 101 001 A9	2	4	16	
			②		①							
CPDR	A − (HL) HL ← HL − 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0	↕	↕	X ↕	X ↕	1	*	11 101 101 ED 10 111 001 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 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	↕	10 000 r	1	1	4	r Reg.
ADD A, n	A ← A + n	↕	↕	X ↕	X V	0	↕	11 000 110 ← n →	2	2	7	000 B 001 C 010 D
ADD A, (HL)	A ← A + (HL)	↕	↕	X ↕	X V	0	↕	10 000 110	1	2	7	011 E
ADD A, (IX + d)	A ← A + (IX + d)	↕	↕	X ↕	X V	0	↕	11 011 101 DD 10 000 110 ← d →	3	5	19	100 H 101 L 111 A
ADD A, (IY + d)	A ← A + (IY + d)	↕	↕	X ↕	X V	0	↕	11 111 101 FD 10 000 110 ← d →	3	5	19	
ADC A, s	A ← A + s + CY	↕	↕	X ↕	X V	0	↕	001				s is any of r, n, (HL), (IX + d), (IY + d) as shown for ADD instruction. The indicated bits replace the 000 in the ADD set above.
SUB s	A ← A − s	↕	↕	X ↕	X V	1	↕	010				
SBC A, s	A ← A − s − CY	↕	↕	X ↕	X V	1	↕	011				
AND s	A ← A ∧ s	↕	↕	X 1	X P	0	0	100				
OR s	A ← A ∨ s	↕	↕	X 0	X P	0	0	110				
XOR s	A ← A ⊕ s	↕	↕	X 0	X P	0	0	101				
CP s	A − s	↕	↕	X ↕	X V	1	↕	111				
INC r	r ← r + 1	↕	↕	X ↕	X V	0	*	00 r 100	1	1	4	
INC (HL)	(HL) ← (HL) + 1	↕	↕	X ↕	X V	0	*	00 110 100	1	3	11	
INC (IX + d)	(IX + d) ← (IX + d) + 1	↕	↕	X ↕	X V	0	*	11 011 101 DD 00 110 100 ← d →	3	6	23	
INC (IY + d)	(IY + d) ← (IY + d) + 1	↕	↕	X ↕	X V	0	*	11 111 101 FD 00 110 100 ← d →	3	6	23	
DEC m	m ← m − 1	↕	↕	X ↕	X V	1	*	101				m is any of r, (HL), (IX + d), (IY + d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode.

General-Purpose Arithmetic and CPU Control Groups

Mnemonic	Symbolic Operation	S	Z	X	H	X	P/V	N	C	Opcode 7 6 5 4 3 2 1 0 Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands.	↕	↕	X	↕	X	P	↕	↕	00 100 111 27	1	1	4	Decimal adjust accumulator.
CPL	$A \leftarrow \overline{A}$	•	•	X	1	X	•	1	•	00 101 111 2F	1	1	4	Complement accumulator (one's complement).
NEG	$A \leftarrow 0 - A$	↕	↕	X	↕	X	V	1	↕	11 101 101 ED 01 000 100 44	2	2	8	Negate acc. (two's complement).
CCF	$CY \leftarrow \overline{CY}$	•	•	X	X	X	•	0	↕	00 111 111 3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	X	0	X	•	0	1	00 110 111 37	1	1	4	Set carry flag.
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000 00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01 110 110 76	1	1	4	
DI ☆	IFF ← 0	•	•	X	•	X	•	•	•	11 110 011 F3	1	1	4	
EI ☆	IFF ← 1	•	•	X	•	X	•	•	•	11 111 011 FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 ED 01 000 110 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 ED 01 010 110 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 ED 01 011 110 5E	2	2	8	

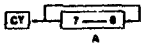
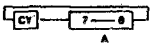
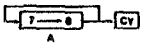
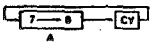
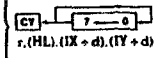
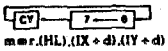
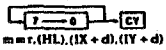
NOTES: IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.
☆ indicates interrupts are not sampled at the end of EI or DI.

16-Bit Arithmetic Group

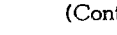
ADD HL, ss	HL ← HL + ss	•	•	X	X	X	•	0	↕	00 ss1 001	1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	HL ← HL + ss + CY	↕	↕	X	X	X	V	0	↕	11 101 101 ED 01 ss1 010	2	4	15	
SBC HL, ss	HL ← HL − ss − CY	↕	↕	X	X	X	V	1	↕	11 101 101 ED 01 ss0 010	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	↕	11 011 101 DD 01 pp1 001	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 FD 00 rr1 001	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 DD 00 100 011 23	2	2	10	
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 111 101 FD 00 100 011 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 DD 00 101 011 2B	2	2	10	
DEC IY	IY ← IY − 1	•	•	X	•	X	•	•	•	11 111 101 FD 00 101 011 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

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 CB 00 000 r	2	2	8	Rotate left circular register r.
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 001 011 CB 00 000 110	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 DD 11 001 011 CB — d — 00 000 110	4	6	23	
RLC (IY + d)		↕	↕	X	0	X	P	0	↕	11 111 101 FD 11 001 011 CB — d — 00 000 110	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 of RLC's with shown code.
RRC m	 m = r,(HL),(IX + d),(IY + d)	↕	↕	X	0	X	P	0	↕	001				

Rotate and Shift Group (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RR m	 m = r,(HL),(IX + d),(IY + d)	↕	↕	X	0	X	P	0	↕	011					
SLA m	m = r,(HL),(IX + d),(IY + d)	↕	↕	X	0	X	P	0	↕	100					
SRA m	m = r,(HL),(IX + d),(IY + d)	↕	↕	X	0	X	P	0	↕	101					
SRL m	m = r,(HL),(IX + d),(IY + d)	↕	↕	X	0	X	P	0	↕	111					
RLD		↕	↕	X	0	X	P	0	•	11 101 101 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

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

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

Jump Group

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11 000 011 — 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: 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	•	•	X	•	X	•	•	•	00 111 000 — e−2 —	38	2	2	7	If condition not met.
	If C = 1, PC ← PC + e											2	3	12	If condition is met.
JR NC, e	If C = 1, continue	•	•	X	•	X	•	•	•	00 110 000 — e−2 —	30	2	2	7	If condition not met.
	If C = 0, PC ← PC + e											2	3	12	If condition is met.
JP Z, e	If Z = 0 continue	•	•	X	•	X	•	•	•	00 101 000 — e−2 —	28	2	2	7	If condition not met.
	If Z = 1, PC ← PC + e											2	3	12	If condition is met.
JR NZ, e	If Z = 1, continue	•	•	X	•	X	•	•	•	00 100 000 — e−2 —	20	2	2	7	If condition not met.
	If Z = 0, PC ← PC + e											2	3	12	If condition is met.
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11 101 001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11 011 101 11 101 001	DD E9	2	2	8	

Jump Group (Continued)

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

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

Call and Return Group

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode	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 CD — n — — n —	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.
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 101 101 ED 01 001 101 4D	2	4	14	
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11 101 101 ED 01 000 101 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	

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

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

NOTE: [1]RETN loads IFF_2 → IFF_1

Input and Output Group

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Opcode	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011 DB — n —	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 ED 01 r 000	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	‡	X	X	X	X	1	•	11 101 101 ED 10 100 010 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	1	X	X	X	X	1	•	11 101 101 ED 10 110 010 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	‡	X	X	X	X	1	•	11 101 101 ED 10 101 010 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	1	X	X	X	X	1	•	11 101 101 ED 10 111 010 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 D3 — n —	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 ED 01 r 001	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	‡	X	X	X	X	1	•	11 101 101 ED 10 100 011 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	1	X	X	X	X	1	•	11 101 101 ED 10 110 011 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	‡	X	X	X	X	1	•	11 101 101 ED 10 101 011 AB	2	4	16	C to A_0 – A_7 B to A_8 – A_{15}

NOTE: ‡ If the result of B − 1 is zero the Z flag is set, otherwise it is reset.

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

Summary of Flag Operation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

CPU Timing

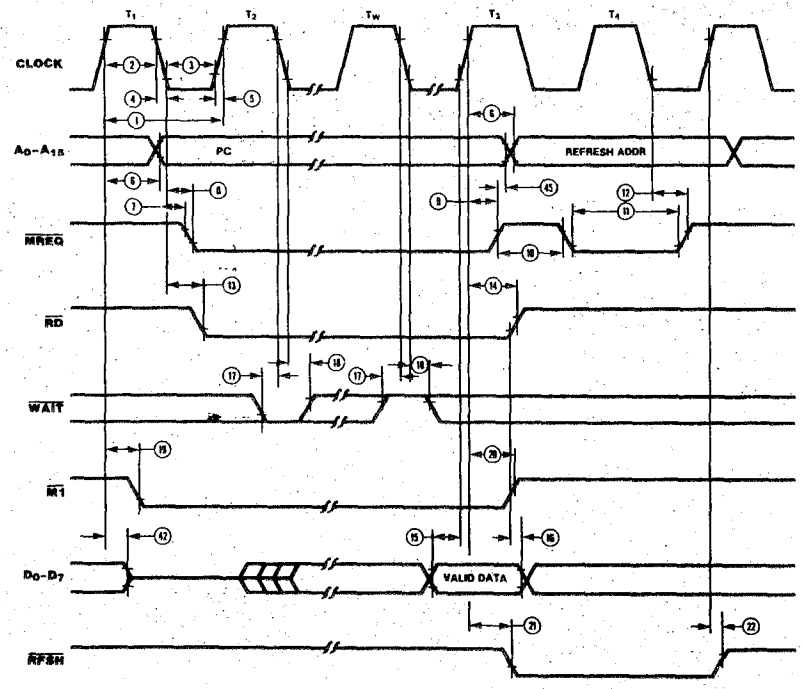
The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

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

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

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

The CPU samples the $\overline{WAIT}$ input with the rising edge of clock state T3. During clock states T3 and T4 of an M1 cycle dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.



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

Figure 5. Instruction Opcode Fetch

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, so that it can be used directly as a Chip Enable for dynamic memories. 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.

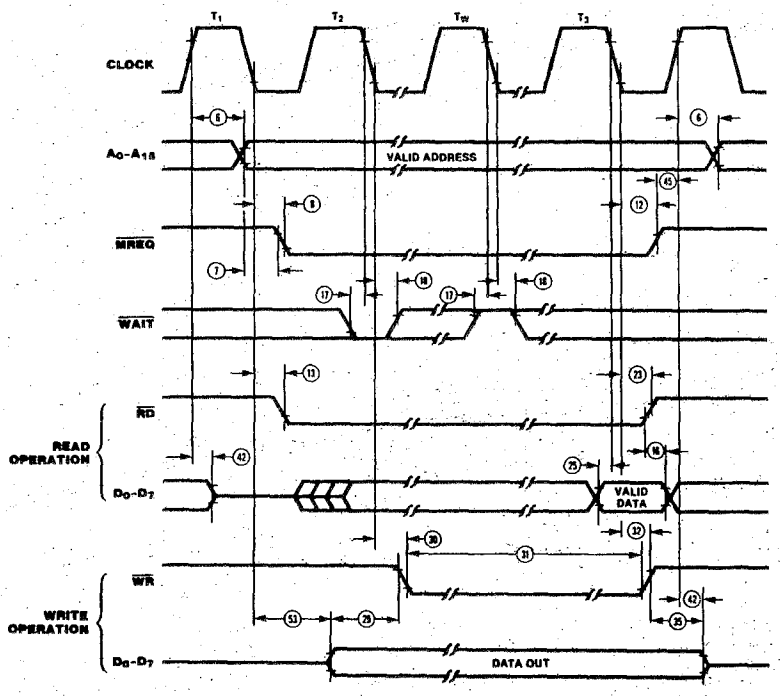
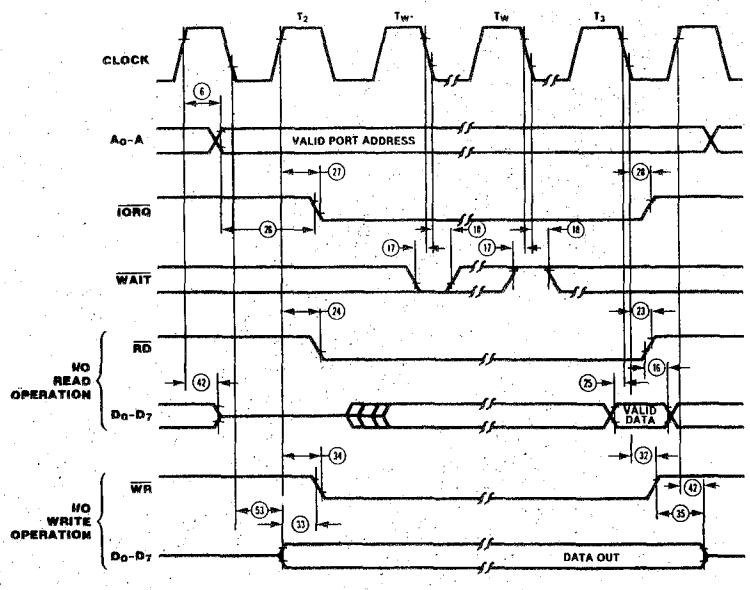


Figure 6. Memory Read or Write Cycles

CPU Timing (Continued)

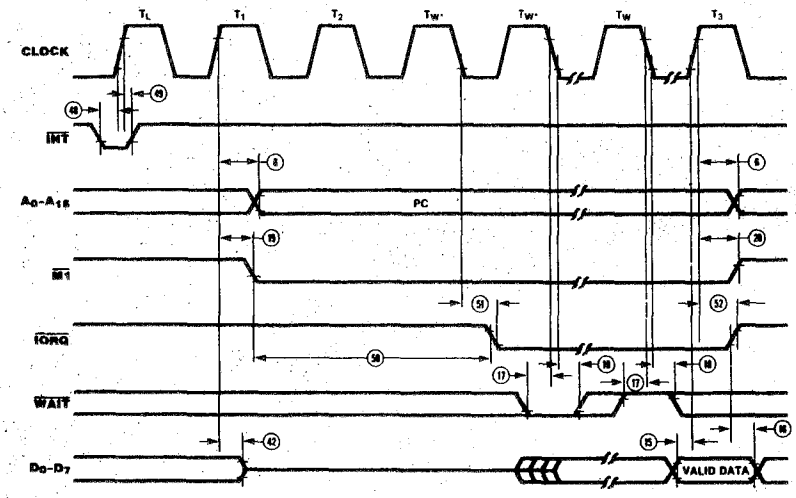
Input or Output Cycles. Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state (T_W). This extra Wait state allows sufficient time for an I/O port to decode the address and the port address lines.



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.

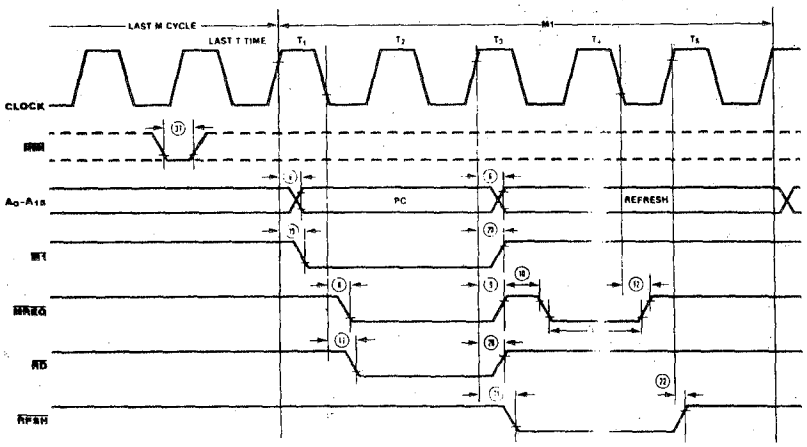


NOTE: 1) T_L = Last state of previous instruction. 2) Two Wait cycles automatically inserted by CPU(*).

Figure 8. Interrupt Request/Acknowledge Cycle

CPU Timing (Continued)

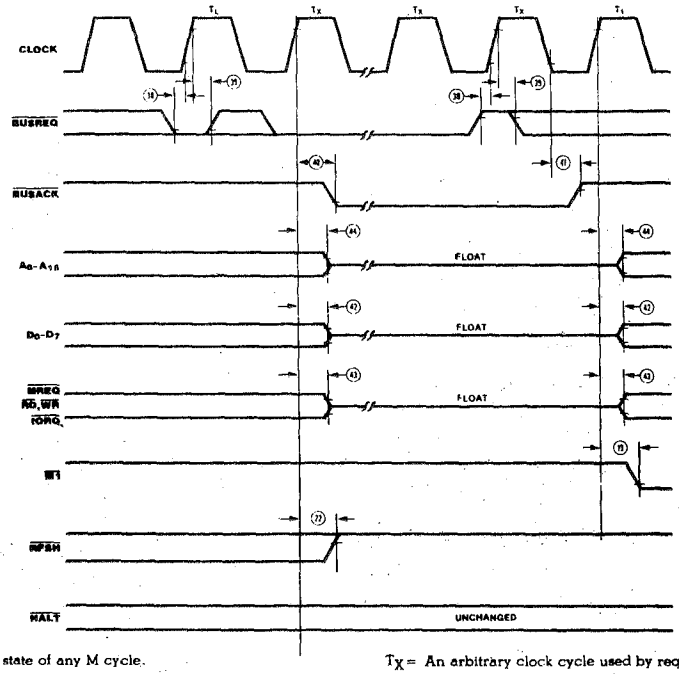
Non-Maskable Interrupt Request Cycle. $\overline{NMI}$ is sampled at the same time as the maskable interrupt input $\overline{INT}$ but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal memory read operation except that data put on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



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

Figure 9. Non-Maskable Interrupt Request Operation

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

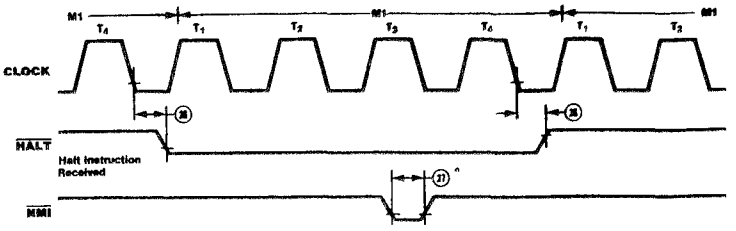


NOTE: T_L = Last state of any M cycle. T_X = An arbitrary clock cycle used by requesting device.

Figure 10. Bus Request/Acknowledge Cycle

CPU Timing (Continued)

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



NOTE: 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, two 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).

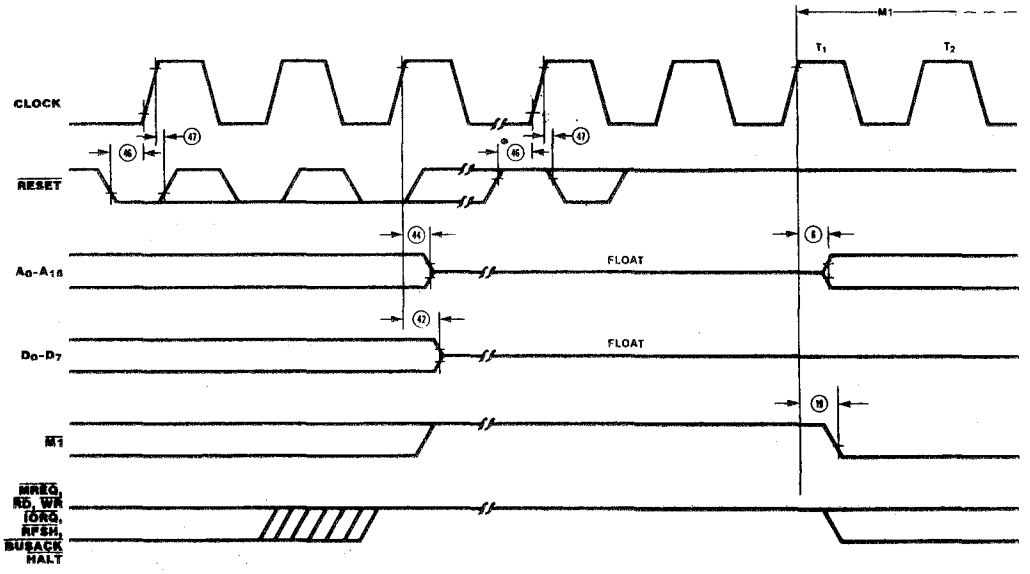


Figure 12. Reset Cycle

AC Characteristics

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

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

AC Characteristics (Continued)

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

*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.

Footnotes to AC Characteristics

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

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

APPENDIX E

Z80 DIRECT MEMORY ADDRESS (DMA)

Z8410
Z80® DMA Direct Memory Access Controller

Zilog

Product Specification

June 1982

Features

- Transfers, searches and search/transfers in Byte-at-a-Time, Burst or Continuous modes. Cycle length and edge timing can be programmed to match the speed of any port.
- Dual port addresses (source and destination) generated for memory-to-I/O, memory-to-memory, or I/O-to-I/O operations. Addresses may be fixed or automatically incremented/decremented.
- Next-operation loading without disturbing current operations via buffered starting-address registers. An entire previous sequence can be repeated automatically.
- Extensive programmability of functions. CPU can read complete channel status.
- Standard Z-80 Family bus-request and prioritized interrupt-request daisy chains implemented without external logic. Sophisticated, internally modifiable interrupt vectoring.
- Direct interfacing to system buses without external logic.

General Description

The Z-80 DMA (Direct Memory Access) is a powerful and versatile device for controlling and processing transfers of data. Its basic function of managing CPU-independent transfers between two ports is augmented by an array of features that optimize transfer speed and control with little or no external logic in systems using an 8- or 16-bit data bus and a 16-bit address bus.

Transfers can be done between any two ports (source and destination), including memory-to-I/O, memory-to-memory, and I/O-to-I/O. Dual port addresses are automatically generated for each transaction and may be either fixed or incrementing/decrementing. In addition, bit-maskable byte searches can be performed either concurrently with transfers or as an operation in itself.

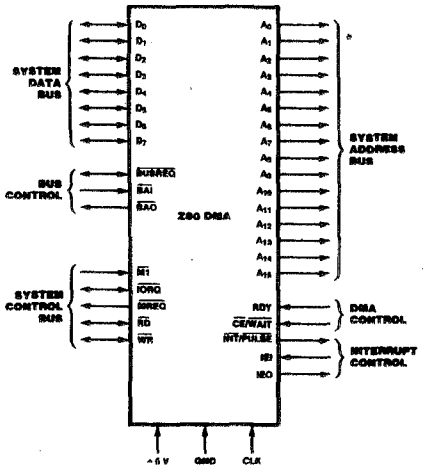


Figure 1. Pin Functions

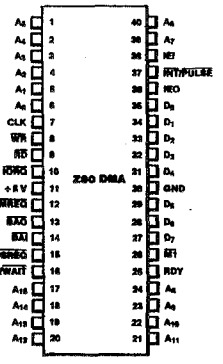


Figure 2. Pin Assignments

General Description (Continued)

The Z-80 DMA contains direct interfacing to and independent control of system buses, as well as sophisticated bus and interrupt controls. Many programmable features, including variable cycle timing and auto-restart, minimize CPU software overhead. They are especially useful in adapting this special-purpose transfer processor to a broad variety of memory, I/O and CPU environments.

The Z-80 DMA is an n-channel silicon-gate depletion-load device packaged in a 40-pin plastic or ceramic DIP. It uses a single +5 V power supply and the standard Z-80 Family single-phase clock.

Functional Description

Classes of Operation. The Z-80 DMA has three basic classes of operation:

- Transfers of data between two ports (memory or I/O peripheral)
- Searches for a particular 8-bit maskable byte at a single port in memory or an I/O peripheral
- Combined transfers with simultaneous search between two ports

Figure 4 illustrates the basic functions served by these classes of operation.

During a transfer, the DMA assumes control of the system address and data buses. Data is read from one addressable port and written to the other addressable port, byte by byte. The ports may be programmed to be either system main memory or peripheral I/O devices. Thus, a block of data may be written from one peripheral to another, from one area of main memory to another, or from a peripheral to main memory and vice versa.

During a search-only operation, data is read from the source port and compared byte by byte with a DMA-internal register containing a programmable match byte. This match byte may optionally be masked so that only certain bits within the match byte are compared. Search rates up to 1.25M bytes per second can be obtained with the 2.5 MHz Z-80 DMA or 2M bytes per second with the 4 MHz Z-80A DMA.

In combined searches and transfers, data is transferred between two ports while simultaneously searching for a bit-maskable byte match.

Data transfers or searches can be programmed to stop or interrupt under various conditions. In addition, CPU-readable status bits can be programmed to reflect the condition.

Modes of Operation. The Z-80 DMA can be programmed to operate in one of three transfer and/or search modes:

- *Byte-at-a-Time:* data operations are performed one byte at a time. Between each byte operation the system buses are released to the CPU. The buses are requested again for each succeeding byte operation.
- *Burst:* data operations continue until a port's Ready line to the DMA goes inactive. The DMA then stops and releases the system buses after completing its current byte operation.
- *Continuous:* data operations continue until the end of the programmed block of data is reached before the system buses are released. If a port's Ready line goes inactive before this occurs, the DMA simply pauses until the Ready line comes active again.

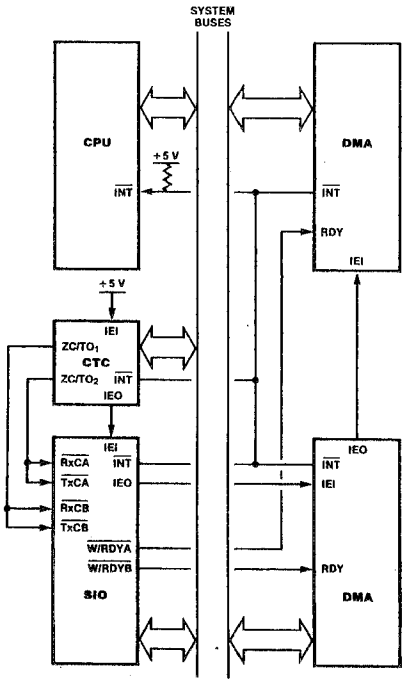
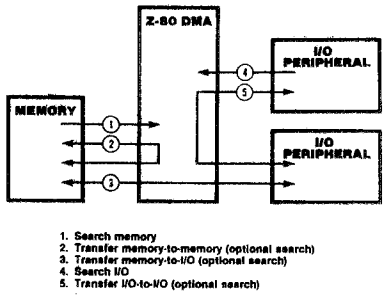


Figure 3. Typical Z-80 Environment



1. Search memory
2. Transfer memory-to-memory (optional search)
3. Transfer memory-to-I/O (optional search)
4. Search I/O
5. Transfer I/O-to-I/O (optional search)

Figure 4. Basic Functions of the Z-80 DMA

Functional Description (Continued)

In all modes, once a byte of data is read into the DMA, the operation on the byte will be completed in an orderly fashion, regardless of the state of other signals (including a port's Ready line).

Due to the DMA's high-speed buffered method of reading data, operations on one byte are not completed until the next byte is read in. This means that total transfer or search block lengths must be two or more bytes, and that block lengths programmed into the DMA must be one byte less than the desired block length (count is N-1 where N is the block length).

Commands and Status. The Z-80 DMA has several writable control registers and readable status registers available to the CPU. Control bytes can be written to the DMA whenever the DMA is not controlling the system buses, but the act of writing a control byte to the DMA disables the DMA until it is again enabled by a specific command. Status bytes can also be read at any such time, but writing the Read Status Byte command or the Initiate Read Sequence command disables the DMA.

Control bytes to the DMA include those which effect immediate command actions such as enable, disable, reset, load starting-address buffers, continue, clear counters, clear status bits and the like. In addition, many mode-setting control bytes can be written, including mode and class of operation, port configuration, starting addresses, block length, address counting rule, match and match-mask byte, interrupt conditions, interrupt vector, status-affects-vector condition, pulse counting, auto restart, Ready-line and Wait-line rules, and read mask.

Readable status registers include a general status byte reflecting Ready-line, end-of-block, byte-match and interrupt conditions, as well as 2-byte registers for the current byte count, Port A address and Port B address.

Variable Cycle. The Z-80 DMA has the unique feature of programmable operation-cycle length. This is valuable in tailoring the DMA to the particular requirements of other system components (fast or slow) and maximizes the data-transfer rate. It also eliminates external logic for signal conditioning.

There are two aspects to the variable cycle feature. First, the entire read and write cycles (periods) associated with the source and destination ports can be independently programmed as 2, 3 or 4 T-cycles long (more if Wait cycles are used), thereby increasing or decreasing the speed with which all DMA signals change (Figure 5).

Second, the four signals in each port specifically associated with transfers of data (I/O Request, Memory Request, Read, and Write) can each have its active trailing edge terminated one-half T-cycle early. This adds a further dimension of flexibility and speed, allowing such things as shorter-than-normal Read or Write signals that go inactive before data starts to change.

Address Generation. Two 16-bit addresses are generated by the Z-80 DMA for every transfer operation, one address for the source port and another for the destination port. Each address can be either variable or fixed. Variable addresses can increment or decrement from the programmed starting address. The fixed-address capability eliminates the need for separate enabling wires to I/O ports.

Port addresses are multiplexed onto the system address bus, depending on whether the DMA is reading the source port or writing to the destination port. Two readable address counters (2 bytes each) keep the current address of each port.

Auto Restart. The starting addresses of either port can be reloaded automatically at the end of a block. This option is selected by the Auto Restart control bit. The byte counter is cleared when the addresses are reloaded.

The Auto Restart feature relieves the CPU of software overhead for repetitive operations such as CRT refresh and many others. Moreover, when the CPU has access to the buses during byte-at-a-time or burst transfers, different starting addresses can be written into buffer registers during transfers, causing the Auto Restart to begin at a new location.

Interrupts. The Z-80 DMA can be programmed to interrupt the CPU on three conditions:

- Interrupt on Ready (before requesting bus)
- Interrupt on Match
- Interrupt on End of Block

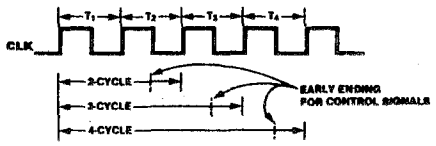


Figure 5. Variable Cycle Length

Functional Description (Continued)

Any of these interrupts cause an interrupt-pending status bit to be set, and each of them can optionally alter the DMA's interrupt vector. Due to the buffered constraint mentioned under "Modes of Operation," interrupts on Match at End of Block are caused by matches to the byte just prior to the last byte in the block.

The DMA shares the Z-80 Family's elaborate interrupt scheme, which provides fast interrupt service in real-time applications. In a Z-80 CPU environment, the DMA passes its internally modifiable 8-bit interrupt vector to the CPU, which adds an additional eight bits to form the memory address of the interrupt-routine table. This table contains the address of the beginning of the interrupt routine itself. In this process, CPU control is transferred directly to the interrupt routine, so that the next instruction executed after an interrupt acknowledge is the first instruction of the interrupt routine itself.

Pulse Generation. External devices can keep track of how many bytes have been transferred by using the DMA's pulse output, which provides a signal at 256-byte intervals. The interval sequence may be offset at the beginning by 1 to 255 bytes.

The Interrupt line outputs the pulse signal in a manner that prevents misinterpretation by the CPU as an interrupt request, since it only appears when the Bus Request and Bus Acknowledge lines are both active.

Pin Description

A_0-A_{15}. *System Address Bus* (output, 3-state). Addresses generated by the DMA are sent to both source and destination ports (main memory or I/O peripherals) on these lines.

$\overline{BAI}$. *Bus Acknowledge In* (input, active Low). Signals that the system buses have been released for DMA control. In multiple-DMA configurations, the $\overline{BAI}$ pin of the highest priority DMA is normally connected to the Bus Acknowledge pin of the CPU. Lower-priority DMAs have their $\overline{BAI}$ connected to the $\overline{BAO}$ of a higher-priority DMA.

$\overline{BAO}$. *Bus Acknowledge Out* (output, active Low). In a multiple-DMA configuration, this pin signals that no other higher-priority DMA has requested the system buses. $\overline{BAI}$ and $\overline{BAO}$ form a daisy chain for multiple-DMA priority resolution over bus control.

$\overline{BUSREQ}$. *Bus Request* (bidirectional, active Low, open drain). As an output, it sends requests for control of the system address bus, data bus and control bus to the CPU. As an input, when multiple DMAs are strung together in a priority daisy chain via $\overline{BAI}$ and $\overline{BAO}$, it senses when another DMA has requested the buses and causes this DMA to refrain from bus requesting until the other DMA is finished. Because it is a bidirectional pin, there cannot be any buffers between this DMA and any other DMA. It can, however, have a buffer between it and the CPU because it is unidirectional into the CPU. A pull-up resistor is connected to this pin.

$\overline{CE/WAIT}$. *Chip Enable and Wait* (input, active Low). Normally this functions only as a $\overline{CE}$ line, but it can also be programmed to serve a $\overline{WAIT}$ function. As a $\overline{CE}$ line from the CPU, it becomes active when $\overline{WR}$ and $\overline{IORQ}$ are active and the I/O port address on the system address bus is the DMA's address, thereby allowing a transfer of control or command bytes from the CPU to the DMA. As a $\overline{WAIT}$ line from memory or I/O devices, after the DMA has received a bus-request acknowledge from the CPU, it causes wait states to be inserted in the DMA's operation cycles thereby slowing the DMA to a speed that matches the memory or I/O device.

CLK. *System Clock* (input). Standard Z-80 single-phase clock at 2.5 MHz (Z-80 DMA) or 4.0 MHz (Z-80A DMA). For slower system clocks, a TTL gate with a pullup resistor may be adequate to meet the timing and voltage level specification. For higher-speed systems, use a clock driver with an active pullup to meet the V_{IH} specification and risetime requirements. In all cases there should be a resistive pullup to the power supply of 10K ohms (max) to ensure proper power when the DMA is reset.

D_0-D_7. *System Data Bus* (bidirectional, 3-state). Commands from the CPU, DMA status, and data from memory or I/O peripherals are transferred on these lines.

IEI. *Interrupt Enable In* (input, active High). This 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 DMA. Thus, this signal blocks lower-priority devices from interrupting while a higher-priority device is being serviced by its CPU interrupt service routine.

Pin Description (Continued)

$\overline{\text{INT}}$/PULSE. *Interrupt Request* (output, active Low, open drain). This requests a CPU interrupt. The CPU acknowledges the interrupt by pulling its $\overline{\text{IORQ}}$ output Low during an $\overline{\text{M1}}$ cycle. It is typically connected to the $\overline{\text{INT}}$ pin of the CPU with a pullup resistor and tied to all other $\overline{\text{INT}}$ pins in the system. This pin can also be used to generate periodic pulses to an external device. It can be used this way only when the DMA is bus master (i.e., the CPU's $\overline{\text{BUSREQ}}$ and $\overline{\text{BUSACK}}$ lines are both Low and the CPU cannot see interrupts).

$\overline{\text{IORQ}}$. *Input/Output Request* (bidirectional, active Low, 3-state). As an input, this indicates that the lower half of the address bus holds a valid I/O port address for transfer of control or status bytes from or to the CPU, respectively; this DMA is the addressed port if its $\overline{\text{CE}}$ pin and its $\overline{\text{WR}}$ or $\overline{\text{RD}}$ pins are simultaneously active. As an output, after the DMA has taken control of the system buses, it indicates that the 8-bit or 16-bit address bus holds a valid port address for another I/O device involved in a DMA transfer of data. When $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are both active simultaneously, an interrupt acknowledge is indicated.

$\overline{\text{M1}}$. *Machine Cycle One* (input, active Low). Indicates that the current CPU machine cycle is an instruction fetch. It is used by the DMA to decode the return-from-interrupt instruction (RETI) (ED-4D) sent by the CPU. During two-byte instruction fetches, $\overline{\text{M1}}$ is active as each opcode byte is fetched. An interrupt acknowledge is indicated when both $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ are active.

MREQ. *Memory Request* (output, active Low, 3-state). This indicates that the address bus holds a valid address for a memory read or write operation. After the DMA has taken control of the system buses, it indicates a DMA transfer request from or to memory.

$\overline{\text{RD}}$. *Read* (bidirectional, active Low, 3-state). As an input, this indicates that the CPU wants to read status bytes from the DMA's read registers. As an output, after the DMA has taken control of the system buses, it indicates a DMA-controlled read from a memory or I/O port address.

RDY. *Ready* (input, programmable active Low or High). This is monitored by the DMA to determine when a peripheral device associated with a DMA port is ready for a read or write operation. Depending on the mode of DMA operation (Byte, Burst or Continuous), the RDY line indirectly controls DMA activity by causing the $\overline{\text{BUSREQ}}$ line to go Low or High.

$\overline{\text{WR}}$. *Write* (bidirectional, active Low, 3-state). As an input, this indicates that the CPU wants to write control or command bytes to the DMA write registers. As an output, after the DMA has taken control of the system buses, it indicates a DMA-controlled write to a memory or I/O port address.

Internal Structure

The internal structure of the Z-80 DMA includes driver and receiver circuitry for interfacing with an 8-bit system data bus, a 16-bit system address bus, and system control lines (Figure 6). In a Z-80 CPU environment, the DMA can be tied directly to the analogous pins on the CPU (Figure 7) with no additional buffering, except for the $\overline{\text{CE}}$/$\overline{\text{WAIT}}$ line.

The DMA's internal data bus interfaces with the system data bus and services all internal logic and registers. Addresses generated from this logic for Ports A and B (source and destination) of the DMA's single transfer channel are multiplexed onto the system address bus.

Specialized logic circuits in the DMA are dedicated to the various functions of external bus interfacing, internal bus control, byte matching, byte counting, periodic pulse generation, CPU interrupts, bus requests and address generation. A set of twenty-one writable control registers and seven readable status registers provides the means by which the CPU governs and monitors the activities of these logic circuits. All registers are eight bits wide, with double-byte information stored in adjacent registers. The two address counters (two bytes each) for Ports A and B are buffered by the two starting addresses.

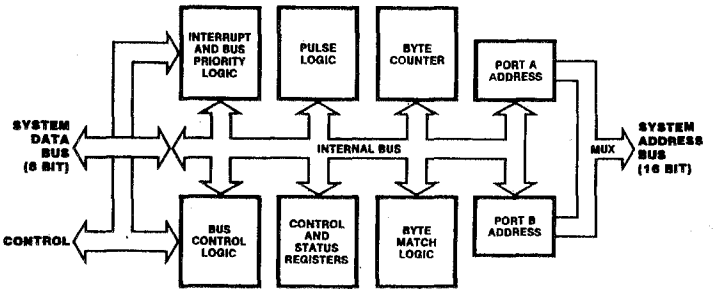


Figure 6. Block Diagram

Internal Structure (Continued)

The 21 writable control registers are organized into seven base-register groups, most of which have multiple registers. The base registers in each writable group contain both control/command bits and pointer bits that can be set to address other registers within the group. The seven readable status registers have no analogous second-level registers.

The registers are designated as follows, according to their base-register groups:

WR0-WR6 — Write Register groups 0 through 6 (7 base registers plus 14 associated registers)
RR0-RR6 — Read Registers 0 through 6

Writing to a register within a write-register group involves first writing to the base register, with the appropriate pointer bits set, then writing to one or more of the other registers within the group. All seven of the readable status registers are accessed sequentially according to a programmable mask contained in one of the writable registers. The section entitled "Programming" explains this in more detail.

A pipelining scheme is used for reading data in. The programmed block length is the number of bytes compared to the byte counter, which increments at the end of each cycle. In searches, data byte comparisons with the match byte are made during the read cycle of the next byte. Matches are, therefore, discovered only after the next byte is read in.

In multiple-DMA configurations, interrupt-request daisy chains are prioritized by the order in which their IEI and IEO lines are connected (Zilog Application Note 03-0041-01, *The Z-80 Family Program Interrupt Structure*). The system bus, however, may not be pre-empted. Any DMA that gains access to the system bus keeps the bus until it is finished.

Write Registers	
WR0	Base register byte Port A starting address (low byte) Port A starting address (high byte) Block length (low byte) Block length (high byte)
WR1	Base register byte Port A variable-timing byte
WR2	Base register byte Port B variable-timing byte
WR3	Base register byte Mask byte Match byte
WR4	Base register byte Port B starting address (low byte) Port B starting address (high byte) Interrupt control byte Pulse control byte Interrupt vector
WR5	Base register byte
WR6	Base register byte Read mask

Read Registers	
RR0	Status byte
RR1	Byte counter (low byte)
RR2	Byte counter (high byte)
RR3	Port A address counter (low byte)
RR4	Port A address counter (high byte)
RR5	Port B address counter (low byte)
RR6	Port B address counter (high byte)

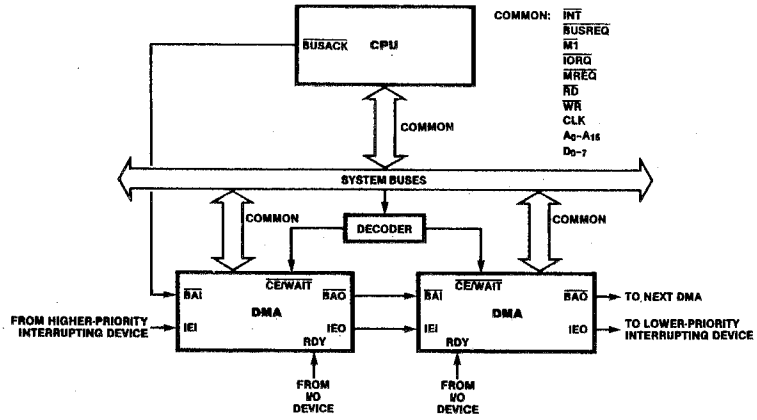


Figure 7. Multiple-DMA Interconnection to the Z-80 CPU

Programming

The Z-80 DMA has two programmable fundamental states: (1) an enabled state, in which it can gain control of the system buses and direct the transfer of data between ports, and (2) a disabled state, in which it can initiate neither bus requests nor data transfers. When the DMA is powered up or reset by any means, it is automatically placed into the disabled state. Program commands can be written to it by the CPU in either state, but this automatically puts the DMA in the disabled state, which is maintained until an enable command is issued by the CPU. The CPU must program the DMA in advance of any data search or transfer by addressing it as an I/O port and sending a sequence of control bytes using an Output instruction (such as OTIR for the Z-80 CPU).

Writing. Control or command bytes are written into one or more of the Write Register groups (WR0-WR6) by first writing to the base register byte in that group. All groups have base registers and most groups have additional associated registers. The associated registers in a group are sequentially accessed by first writing a byte to the base register containing register-group identification and pointer bits (1's) to one or more of that base register's associated registers.

This is illustrated in Figure 8b. In this figure, the sequence in which associated registers within a group can be written to is shown by the vertical position of the associated registers. For example, if a byte written to the DMA contains the bits that identify WR0 (bits D0, D1 and D7), and also contains 1's in the bit positions that point to the associated "Port A Starting Address (low byte)" and "Port A Starting Address (high byte)," then the next two bytes written to the DMA will be stored in these two registers, in that order.

Reading. The Read Registers (RR0-RR6) are read by the CPU by addressing the DMA as an I/O port using an Input instruction (such as INIR for the Z-80 CPU). The readable bytes contain DMA status, byte-counter values, and port addresses since the last DMA reset. The registers are always read in a fixed sequence beginning with RR0 and ending with RR6. However, the register read in this sequence is determined by programming the Read Mask in WR6. The sequence of reading is initialized by writing an Initiate Read Sequence or Set Read Status command to WR6. After a Reset DMA, the sequence must be initialized with the Initiate Read Sequence command or a Read Status command. The sequence of reading all registers that are not excluded by the Read Mask register must be completed before a new Initiate Read Sequence or Read Status command.

Fixed-Address Programming. A special circumstance arises when programming a destination port to have a fixed address. The load command in WR6 only loads a fixed address to a port selected as the source, not to a port selected as the destination. Therefore, a fixed destination address must be loaded by temporarily declaring it a fixed-source address and subsequently declaring the true source as such, thereby implicitly making the other a destination.

The following example illustrates the steps in this procedure, assuming that transfers are to occur from a variable-address source (Port A) to a fixed-address destination (Port B):

1. Temporarily declare Port B as source in WR0.
2. Load Port B address in WR6.
3. Declare Port A as source in WR0.
4. Load Port A address in WR6.
5. Enable DMA in WR6.

Figure 9 illustrates a program to transfer data from memory (Port A) to a peripheral device (Port B). In this example, the Port A memory starting address is 1050_H and the Port B peripheral fixed address is 05_H. Note that the data flow is 1001_H bytes—one more than specified by the block length. The table of DMA commands may be stored in consecutive memory locations and transferred to the DMA with an output instruction such as the Z-80 CPU's OTIR instruction.

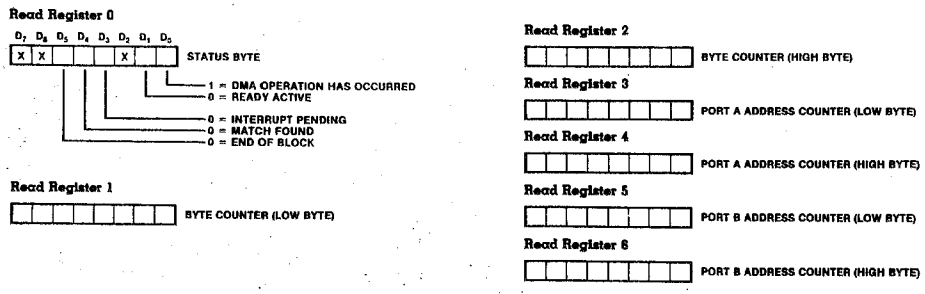


Figure 8a. Read Registers

Programming
(Continued)

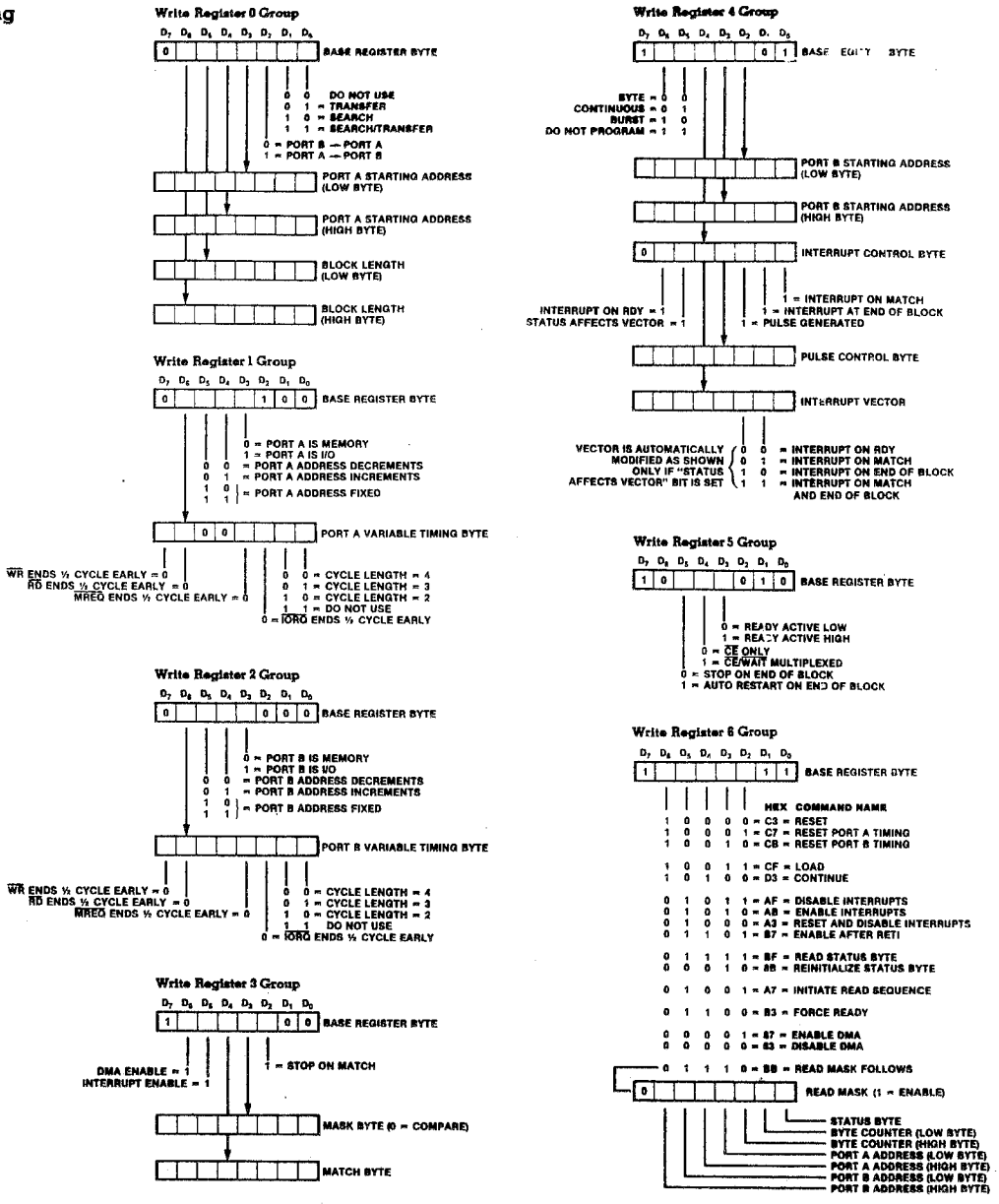


Figure 8b. Write Registers

Comments	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX
WR0 sets DMA to receive block length, Port A starting address and temporarily sets Port B as source.	0	1 Block Length Upper Follows	1 Block Length Lower Follows	1 Port A Upper Address Follows	1 Port A Lower Address Follows	0 B ⟶ A Temporary for Loading B Address*	0	1 Transfer, No Search	79
Port A address (lower)	0	1	0	1	0	0	0	0	50
Port A address (upper)	0	0	0	1	0	0	0	0	10
Block length (lower)	0	0	0	0	0	0	0	0	00
Block length (upper)	0	0	0	1	0	0	0	0	10
WR1 defines Port A as memory with fixed incrementing address.	0	0 No Timing Follows	0 Address Changes	1 Address Increments	0 Port is Memory	1	0	0	14
WR2 defines Port B as peripheral with fixed address.	0	0 No Timing Follows	1 Fixed Address	0	1 Port is I/O	0	1	0	28
WR4 sets mode to Burst, sets DMA to expect Port B address.	1	1 Burst Mode	0	0 No Interrupt Control Byte Follows	0 No Upper Address	1 Port B Lower Address Follows	0	1	C5
Port B address (lower)	0	0	0	0	0	1	0	1	05
WR5 sets Ready active High.	1	0	0 No Auto Restart	0 No Wait States	1 RDY Active High	0	1	0	8A
WR6 loads Port B address and resets block counter.*	1	1	0	0	1	1	1	1	CF
WR0 sets Port A as source.*	0	0	0 No Address or Block Length Bytes	0	0	1 A ⟶ B	0	1 Transfer, No Search	05
WR6 loads Port A address and resets block counter.	1	1	0	0	1	1	1	1	CF
WR6 enables DMA to start operation.	1	0	0	0	0	1	1	1	87

NOTE: The actual number of bytes transferred is one more than specified by the block length.
*These entries are necessary only in the case of a fixed destination address.

Figure 9. Sample DMA Program

2032-0133

Inactive State Timing (DMA as CPU Peripheral)

In its disabled or inactive state, the DMA is addressed by the CPU as an I/O peripheral for write and read (control and status) operations. Write timing is illustrated in Figure 10.

Reading of the DMA's status byte, byte counter or port address counters is illustrated in Figure 11. These operations require less than three T-cycles. The $\overline{CE}$, $\overline{IORQ}$ and $\overline{RD}$ lines are made active over two rising edges of CLK, and data appears on the bus approximately one T-cycle after they become active.

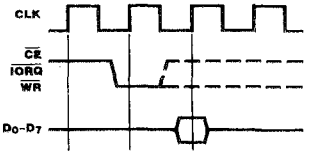


Figure 10. CPU-to-DMA Write Cycle

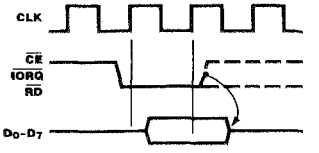


Figure 11. CPU-to-DMA Read Cycle

Active State Timing (DMA as Bus Controller)

Default Read and Write Cycles. By default, and after reset, the DMA's timing of read and write operations is exactly the same as the Z-80® CPU's timing of read and write cycles for memory and I/O peripherals, with one exception: during a read cycle, data is latched on the falling edge of T_3 and held on the data bus across the boundary between read and write cycles, through the end of the following write cycle.

Figure 12 illustrates the timing for memory-to-I/O port transfers and Figure 13 illustrates I/O-to-memory transfers. Memory-to-memory and I/O-to-I/O transfer timings are simply permutations of these diagrams.

The default timing uses three T-cycles for memory transactions and four T-cycles for I/O transactions, which include one automatically inserted wait cycle between T_2 and T_3. If the $\overline{CE}/\overline{WAIT}$ line is programmed to act as a $\overline{WAIT}$ line during the DMA's active state, it is sampled on the falling edge of T_2 for memory transactions and the falling edge of T_W for I/O transactions. If $\overline{CE}/\overline{WAIT}$ is Low during this time another T-cycle is added, during which the $\overline{CE}/\overline{WAIT}$ line will again be sampled. The duration of transactions can thus be indefinitely extended.

Variable Cycle and Edge Timing. The Z-80 DMA's default operation-cycle length for the source (read) port and destination (write) port can be independently programmed. This variable-cycle feature allows read or write cycles consisting of two, three or four T-cycles (more if Wait cycles are inserted), thereby increasing or decreasing the speed of all signals generated by the DMA. In addition,

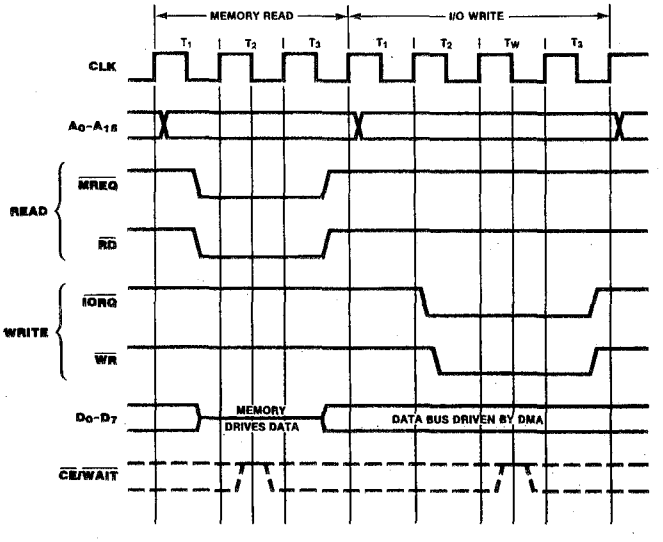


Figure 12. Memory-to-I/O Transfer

Active State Timing (DMA as Bus Controller) (Continued)

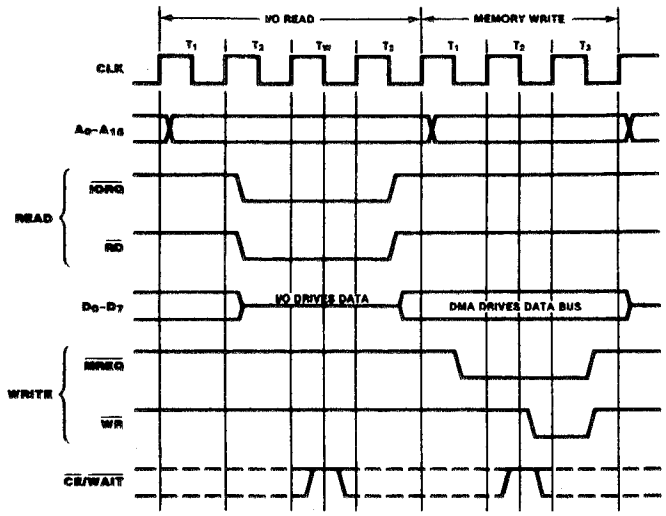


Figure 13. I/O-to-Memory Transfer

the trailing edges of the $\overline{IORQ}$, $\overline{MREQ}$, $\overline{RD}$ and $\overline{WR}$ signals can be independently terminated one-half cycle early. Figure 14 illustrates this.

In the variable-cycle mode, unlike default timing, $\overline{IORQ}$ comes active one-half cycle before $\overline{MREQ}$, $\overline{RD}$ and $\overline{WR}$. $\overline{CE}/\overline{WAIT}$ can be used to extend only the 3 or 4 T-cycle variable memory cycles and only the 4-cycle variable I/O cycle. The $\overline{CE}/\overline{WAIT}$ line is sampled at the falling edge of T_2 for 3- or 4-cycle memory cycles, and at the falling edge of T_3 for 4-cycle I/O cycles.

During transfers, data is latched on the clock edge causing the rising edge of $\overline{RD}$ and held through the end of the write cycle.

Bus Requests. Figure 15 illustrates the bus request and acceptance timing. The RDY line, which may be programmed active High or Low, is sampled on every rising edge of CLK. If it is found to be active, and if the bus is not in use by any other device, the following rising edge of CLK drives $\overline{BUSREQ}$ low. After receiving $\overline{BUSREQ}$ the CPU acknowledges on the $\overline{BAI}$ input either directly or through a multiple-DMA daisy chain. When a Low is detected on $\overline{BAI}$ for two consecutive rising edges of CLK, the DMA will begin transferring data on the next rising edge of CLK.

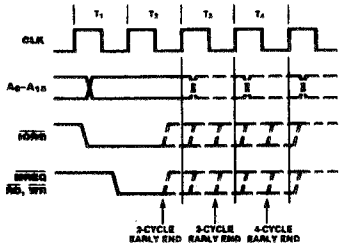


Figure 14. Variable-Cycle and Edge Timing

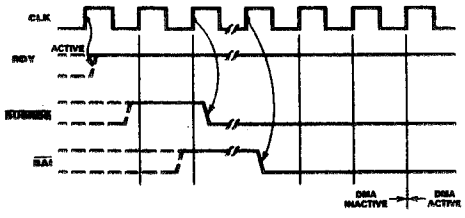


Figure 15. Bus Request and Acceptance

Active State Timing (DMA as Bus Controller) (Continued)

Bus Release Byte-at-a-Time. In Byte-at-a-Time mode, $\overline{BUSREQ}$ is brought High on the rising edge of CLK prior to the end of each read cycle (search-only) or write cycle (transfer and transfer/search) as illustrated in Figure 16. This is done regardless of the state of RDY. There is no possibility of confusion when a Z-80 CPU is used since the CPU cannot begin an operation until the following T-cycle. Most other CPUs are not bothered by this either, although note should be taken of it. The next bus request for the next byte will come after both $\overline{BUSREQ}$ and $\overline{BAI}$ have returned High.

Bus Release at End of Block. In Burst and Continuous modes, an end of block causes $\overline{BUSREQ}$ to go High usually on the same rising edge of CLK in which the DMA completes the transfer of the data block (Figure 17). The last byte in the block is transferred even if RDY goes inactive before completion of the last byte transfer.

Bus Release on Not Ready. In Burst mode, when RDY goes inactive it causes $\overline{BUSREQ}$ to go High on the next rising edge of CLK after the completion of its current byte operation (Figure 18). The action on $\overline{BUSREQ}$ is thus somewhat delayed from action on the RDY line. The DMA always completes its current byte operation in an orderly fashion before releasing the bus.

By contrast, $\overline{BUSREQ}$ is not released in Continuous mode when RDY goes inactive. Instead, the DMA idles after completing the current byte operation, awaiting an active RDY again.

Bus Release on Match. If the DMA is programmed to stop on match in Burst or Continuous modes, a match causes $\overline{BUSREQ}$ to go inactive on the next DMA operation, i.e., at the end of the next read in a search or at the end of the following write in a transfer (Figure 19). Due to the pipelining scheme, matches are determined while the next DMA read or write is being performed.

The RDY line can go inactive after the matching operation begins without affecting this bus-release timing.

Interrupts. Timings for interrupt acknowledge and return from interrupt are the same as timings for these in other Z-80 peripherals. Refer to Zilog Application Note 03-0041-01 *(The Z-80 Family Program Interrupt Structure).*

Interrupt on RDY (interrupt before requesting bus) does not directly affect the $\overline{BUSREQ}$ line. Instead, the interrupt service routine must handle this by issuing the following commands to WR6:

1. Enable after Return From Interrupt (RETI) Command — Hex B7
2. Enable DMA — Hex 87
3. An RETI instruction that resets the Interrupt Under Service latch in the Z-80 DMA.

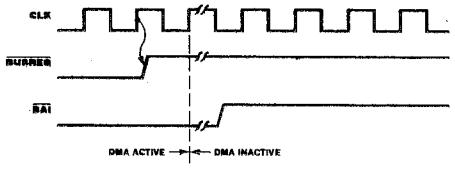


Figure 16. Bus Release (Byte-at-a-Time Mode)

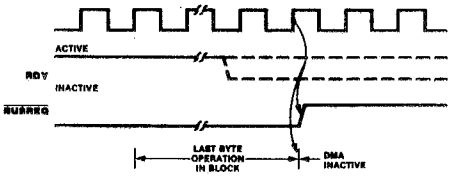


Figure 17. Bus Release at End of Block (Burst and Continuous Modes)

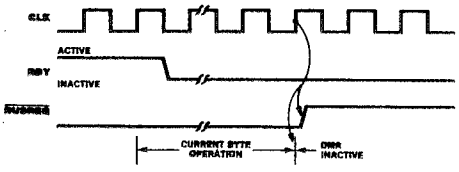


Figure 18. Bus Release When Not Ready (Burst Mode)

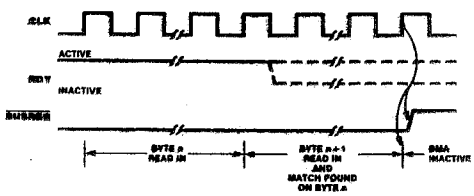


Figure 19. Bus Release on Match (Burst and Continuous Modes)

APPENDIX F

FLOPPY DISK CONTROLLER (FDC)

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

WD279X-02 Floppy Disk Formatter/Controller Family

FEATURES

- ON-CHIP PLL DATA SEPARATOR
- ON-CHIP WRITE PRECOMPENSATION LOGIC
- SINGLE +5V SUPPLY
- ACCOMMODATES SINGLE AND DOUBLE DENSITY FORMATS
  - IBM 3740 (FM)
  - IBM 34 (MFM)
- AUTOMATIC SEEK WITH VERIFY
- MULTIPLE SECTOR READ/WRITE
- TTL COMPATIBLE
- PROGRAMMABLE CONTROL
  - SELECTABLE TRACK-TO-TRACK ACCESS
  - HEAD LOAD TIMING
- SOFTWARE COMPATIBLE WITH THE FD179X SERIES
- SOFT SECTOR FORMAT COMPATIBILITY

APPLICATIONS

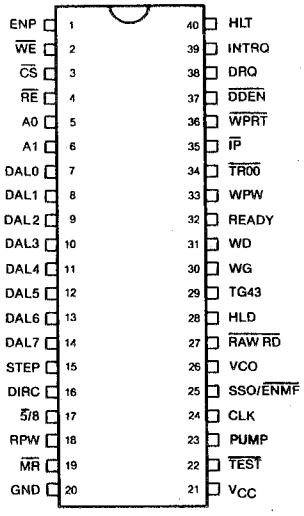
8" FLOPPY AND 5¼" MINI FLOPPY CONTROLLER
SINGLE OR DOUBLE DENSITY
CONTROLLER/FORMATTER

The WD279X Family are MOS/LSI devices which perform the functions of a Floppy Disk Controller/Formatter. Software compatible with its predecessor, the FD179X, the device also contains a high performance Phase-Lock-Loop Data Separator as well as Write Precompensation Logic.

When operating in Double Density mode, Write Precompensation is automatically engaged to a value programmed via an external potentiometer. An on-chip VCO and phase comparator allows adjustable frequency range for 5¼" or 8" Floppy Disk interfacing.

The WD279X is fabricated in NMOS silicon gate technology and available in a 40 pin dual-in-line ceramic or plastic package.

FEATURES	2791	2793	2795	2797
Single Density (FM)	X	X	X	X
Double Density (MFM)	X	X	X	X
True Data Bus		X		X
Inverted Data Bus	X		X	
Side Select Out			X	X
Internal CLK Divide	X	X		



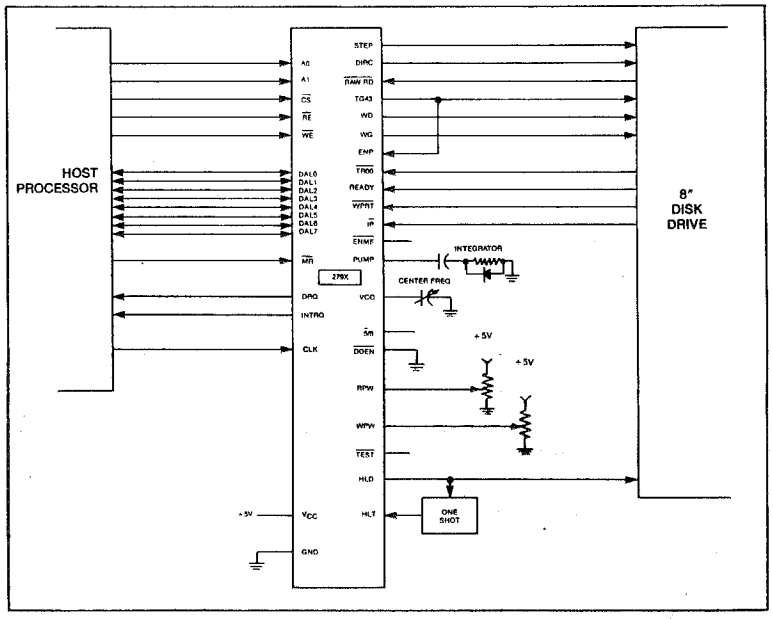


Figure 1.

PIN OUTS

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	ENABLE PRECOMP	ENP	A Logic high on this input enables write precompensation to be performed on the Write Data output.
19	MASTER RESET	$\overline{MR}$	A logic low (50 microseconds min.) on this input resets the device and loads HEX 03 into the command register. The Not Ready (Status Bit 7) is reset during MR ACTIVE. When $\overline{MR}$ is brought to a logic high a RESTORE Command is executed, regardless of the state of the Ready signal from the drive. Also, HEX 01 is loaded into sector register.
20	POWER SUPPLIES	V_{SS}	Ground
21		V_{CC}	+5V ±5%
COMPUTER INTERFACE:			
2	WRITE ENABLE	$\overline{WE}$	A logic low on this input gates data on the DAL into the selected register when $\overline{CS}$ is low.
3	CHIP SELECT	$\overline{CS}$	A logic low on this input selects the chip and enables computer communication with the device.
4	READ ENABLE	$\overline{RE}$	A logic low on this input controls the placement of data from a selected register on the DAL when $\overline{CS}$ is low.
5, 6	REGISTER SELECT LINES	A0, A1	These inputs select the register to receive/transfer data on the DAL lines under $\overline{RE}$ and $\overline{WE}$ control: $\overline{CS}$ A1 A0 $\overline{RE}$ $\overline{WE}$ 0 0 0 Status Reg Command Reg 0 0 1 Track Reg Track Reg 0 1 0 Sector Reg Sector Reg 0 1 1 Data Reg Data Reg
7-14	DATA ACCESS LINES	DAL0-DAL7	Eight bit bi-directional bus used for transfer of commands, status, and data. These lines are inverted (active low) on WD2791 and WD2795.
24	CLOCK	CLK	This input requires a free-running 50% duty cycle square wave clock for internal timing reference, 2 MHz ± 1% for 8" drives, 1 MHz ± 1% for mini-floppies.
38	DATA REQUEST	DRQ	This output indicates that the Data Register contains assembled data in Read operations, or the DR is empty in Write operations. This signal is reset when serviced by the computer through reading or loading the DR.
39	INTERRUPT REQUEST	INTRQ	This output is set at the completion of any command and is reset when the Status register is read or the Command register is written to.
FLOPPY DISK INTERFACE:			
15	STEP	STEP	The step output contains a pulse for each step.
16	DIRECTION	DIRC	Direction Output is active high when stepping in, active low when stepping out.
17	5¼," 8" SELECT	$\overline{5/8}$	This input selects the internal VCO frequency for use with 5¼" drives or 8" drives.
18	READ PULSE WIDTH	RPW	An external potentiometer tied to this input controls the phase comparator within the data separator.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	A logic low on this input allows adjustment of external resistors by enabling internal signals to appear on selected pins.
23	PUMP	PUMP	High-Impedance output signal which is forced high or low to increase/decrease the VCO frequency.
25	ENABLE MINI-FLOPPY (2791, 2793)	$\overline{\text{ENMF}}$	A logic low on this input enables an internal ÷2 of the Master Clock when $\overline{5/8}$ is also at a logic 0. This allows both 5¼" and 8" drive operation with a single 2 MHz clock. For a 1 MHz clock on Pin 24, this line must be left open or tied to a Logic 1.
25	SIDE SELECT OUTPUT (2795, 2797)	SSO	The logic level of the Side Select Output is directly controlled by the 'S' flag in Type II or III commands. When U = 1, SSO is set to a logic 1. When U = 0, SSO is set to a logic 0. The SSO is compared with the side information in the Sector I.D. Field. If they do not compare Status Bit 4 (RNF) is set. The Side Select Output is only updated at the beginning of a Type II or III command. It is forced to a logic 0 upon a MASTER RESET condition.
26	VOLTAGE-CONTROLLED OSCILLATOR	VCO	An external capacitor tied to this pin adjusts the VCO center frequency.
27	$\overline{\text{RAW READ}}$	$\overline{\text{RAW READ}}$	The data input signal directly from the drive. This input shall be a negative pulse for each recorded flux transition.
28	HEAD LOAD	HLD	The HLD output controls the loading of the Read-Write head against the media.
29	TRACK GREATER THAN 43	TG43	This output informs the drive that the Read/Write head is positioned between tracks 44-76. This output is valid only during Read and Write Commands.
30	WRITE GATE	WG	This output is made valid before writing is to be performed on the diskette.
31	WRITE DATA	WD	A 250ns (MFM) or 500 ns (FM) output pulse per flux transition. WD contains the unique Address marks as well as data and clock in both FM and MFM formats.
32	READY	READY	This input indicates disk readiness and is sampled for a logic high before Read or Write commands are performed. If Ready is low the Read or Write operation is not performed and an interrupt is generated. Type I operations are performed regardless of the state of Ready. The Ready input appears in inverted format as Status Register bit 7.
33	WRITE PRECOMP WIDTH	WPW	An external potentiometer tied to this input controls the amount of delay in Write precompensation mode.
34	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This input informs the WD279X that the Read/Write head is positioned over Track 00.
35	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	This input informs the WD279X when the index hole is encountered on the diskette.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
36	WRITE PROTECT	$\overline{WPRT}$	This input is sampled whenever a Write Command is received. A logic low terminates the command and sets the Write Protect Status bit.
37	DOUBLE DENSITY	$\overline{DDEN}$	This input pin selects either single or double density operation. When $\overline{DDEN}$ = 0, double density is selected. When $\overline{DDEN}$ = 1, single density is selected.
40	HEAD LOAD TIMING	HLT	When a logic high is found on the HLT input the head is assumed to be engaged. It is typically derived from a 1 shot triggered by HLD.

GENERAL DESCRIPTION

The WD279X are N-Channel Silicon Gate MOS LSI devices which perform the functions of a Floppy Disk Formatter/Controller in a single chip implementation. The WD279X, which can be considered the end result of both the FD1771 and FD179X designs, is IBM 3740 compatible in single density mode (FM) and System 34 compatible in Double Density Mode (MFM). The WD279X contains all the features of its predecessor the FD179X plus a high performance Phase-Lock-Loop Data Separator as well as Write Precompensation Logic. In Double Density mode, Write Precompensation is automatically engaged to a value programmed via an external potentiometer. In order to maintain compatibility, the FD1771, FD179X and WD279X designs were made as close as possible with the computer interface, instruction set, and I/O registers being identical. Also, head load control is identical. In each case, the actual pin assignments vary by only a few pins from any one to another.

The processor interface consists of an 8-bit bi-directional bus for data, status, and control word transfers. The WD279X is set up to operate on a multiplexed bus with other bus-oriented devices.

The WD279X is TTL compatible on all inputs and outputs. The outputs will drive one TTL load or three LS loads. The 2793 is identical to the 2791 except the DAL lines are TRUE for systems that utilize true data busses.

The 2795/7 has a side select output for controlling double sided drives.

ORGANIZATION

The Floppy Disk Formatter block diagram is illustrated on page 5. 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 ($\overline{RAW\ READ}$) 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 (towards track 76) 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: $G(x) = x^{16} + x^{12} + x^5 + 1$.

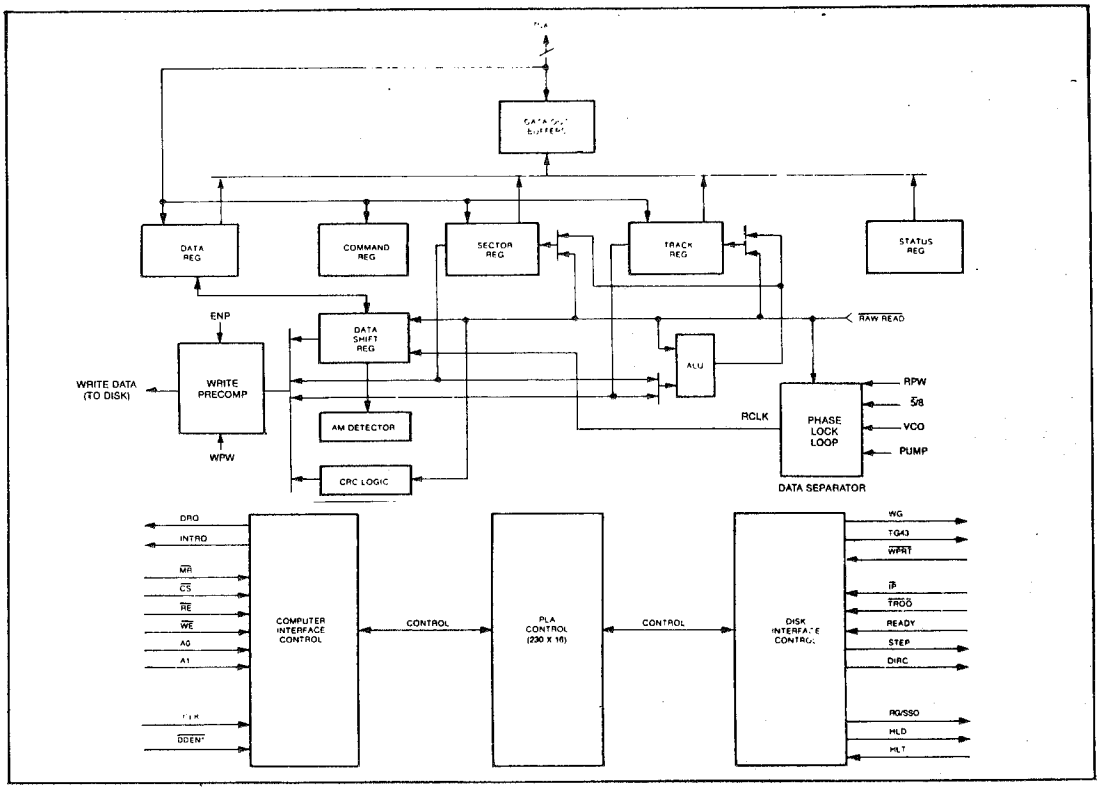
The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

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.

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.

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

Write Precompensation — enables write precompensation to be performed on the Write Data output.



WD279X BLOCK DIAGRAM

Data Separator — a high performance Phase-Lock-Loop Data Separator with on-chip VCO and phase comparator allows adjustable frequency range for 5¼" or 8" Floppy Disk interfacing.

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 WD279X. The DAL are three state buffers that are enabled as output drivers when Chip Select ($\overline{CS}$) and Read Enable ($\overline{RE}$) are active (low logic state) or act as input receivers when $\overline{CS}$ and Write Enable ($\overline{WE}$) 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 signals $\overline{RE}$ during a Read operation or $\overline{WE}$ during a Write operation are interpreted as selecting the following registers:

A1	A0	READ ($\overline{RE}$)	WRITE ($\overline{WE}$)
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 WD279X 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 operation 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 bit 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 279X has two modes of operation according to the state of $\overline{DDEN}$ (Pin 37). When $\overline{DDEN}$ = 1, Single Density (FM) is selected. When $\overline{DDEN}$ = 0, Double Density (MFM) is selected. In either case, the CLK input (Pin 24) is set at 2 MHz for 8" drives or 1 MHz for 5¼" drives.

On the 2791/2793, the $\overline{ENMF}$ input (Pin 25) can be used for controlling both 5¼" and 8" drives with a single 2 MHz clock. When $\overline{ENMF}$ = 0, an internal ÷ 2 of the CLK is performed. When $\overline{ENMF}$ = 1, no divide takes place. This allows the use of a 2 MHz clock for both 5¼" and 8" configurations.

The internal VCO frequency must also be set to the proper value. The $\overline{5/8}$ input (Pin 17) is used to select data separator operation by internally dividing the Read Clock. When $\overline{5/8}$ = 0, 5¼" data separation is selected; when 5/8 = 1, 8" drive data separation is selected.

CLOCK (24)	$\overline{ENMF}$ (25)	$\overline{5/8}$ (17)	DRIVE
2 MHz	1	1	8"
2 MHz	0	0	5¼"
1 MHz	1	0	5¼"

Note: All other conditions invalid.

FUNCTIONAL DESCRIPTION

The WD279X-02 is software compatible with the FD179X-02 series of Floppy Disk Controllers. Commands, status, and data transfers are performed in the same way. Software generated for the 179X can be transferred to a 279X system without modification.

In addition to the 179X, the 279X contains an internal Data Separator and Write precompensation circuit. The $\overline{TEST}$ (Pin 22) line is used to adjust both data separator and precompensation. When $\overline{TEST}$ = 0, the WD (Pin 31) line is internally connected to the output of the write precomp one-shot. Adjustment of the WPW (Pin 33) line can then be accomplished. A second one-shot tracks the precomp setting at approximately 3:1 to insure adequate Write Data pulse widths to meet drive specifications.

Similarly, Data separation is also adjusted with $\overline{TEST}$ = 0. The TG43 (Pin 29) line is internally connected to the output of the read data one-shot, which is adjusted via the RPW (Pin 18) line. The DIRC (Pin 16) line contains the Read Clock output (.5 MHz for 8" drives). The VCO Trimming capacitor (Pin 26) is adjusted for center frequency.

Internal timing signals are used to generate pulses during the adjustment mode so that these adjustments can be made while the device is in-circuit. The $\overline{TEST}$ line also contains a pull-up resistor, so adjustments can be performed simply by grounding the $\overline{TEST}$ pin, overriding the pull-up. The $\overline{TEST}$ pin cannot be used to disable stepping rates during operation as its function is quite different from the 179X.

Other pins on the device also include pull-up resistors and may be left open to satisfy a Logic 1 condition. These are: ENP, $\overline{5/8}$, $\overline{ENMF}$, $\overline{WPRT}$, and $\overline{DDEN}$.

GENERAL DISK READ OPERATIONS

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

Sector Length Table*	
Sector Length Field (hex)	Number of Bytes in Sector (decimal)
00	128
01	256
02	512
03	1024

* 2795/97 may vary — see command summary.

The number of sectors per track as far as the 279X is concerned can be from 1 to 255 sectors. The number of tracks as far as the 279X is concerned is from 0 to 255 tracks. For IBM 3740 compatibility, sector lengths are 128 bytes with 26 sectors per track. For System 34 compatibility (MFM), sector lengths are 256 bytes/sector with 26 sectors/track; or lengths of 1024 bytes/sector with 8 sectors/track.

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 279X before the Write Gate signal can be activated.

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

For write operations, the 279X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of pulses set to a width approximately three times greater than the precomp adjustment. Write Data provides the unique address marks in both formats.

READY

Whenever a Read or Write command (Type II or III) is received the 297X samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. All Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated. TG43 may be tied to ENP to enable write precompensation on tracks 44-76.

COMMAND DESCRIPTION

The WD279X 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.

TABLE 1. COMMAND SUMMARY

A. Commands for Models: 2791, 2793

Type	Command	Bits 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	T	h	V	r_1	r_0
I	Step-in	0	1	0	T	h	V	r_1	r_0
I	Step-out	0	1	1	T	h	V	r_1	r_0
II	Read Sector	1	0	0	m	S	E	C	0
II	Write Sector	1	0	1	m	S	E	C	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

B. Commands for Models: 2795, 2797

Type	Command	Bits 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	T	h	V	r_1	r_0
I	Step-in	0	1	0	T	h	V	r_1	r_0
I	Step-out	0	1	1	T	h	V	r_1	r_0
II	Read Sector	1	0	0	m	L	E	U	0
II	Write Sector	1	0	1	m	L	E	U	a_0
III	Read Address	1	1	0	0	0	E	U	0
III	Read Track	1	1	1	0	0	E	U	0
III	Write Track	1	1	1	1	0	E	U	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

FLAG SUMMARY

TABLE 2. FLAG SUMMARY

Command Type	Bit No(s)		Description
I	0, 1	r_1 r_0 = Stepping Motor Rate See Table 3 for Rate Summary	
I	2	V = Track Number Verify Flag	V = 0, No verify V = 1, Verify on destination track
I	3	h = Head Load Flag	h = 0, Unload head at beginning h = 1, Load head at beginning
I	4	T = Track Update Flag	T = 0, No update T = 1, Update track register
II & III	0	a_0 = Data Address Mark	a_0 = 0, FB (DAM) a_0 = 1, F8 (deleted DAM)
II	1	C = Side Compare Flag	C = 0, Disable side compare C = 1, Enable side compare
II & III	1	U = Update SSO	U = 0, Update SSO to 0 U = 1, Update SSO to 1
II & III	2	E = 15 MS Delay	E = 0, No. 15 MS delay E = 1, 15 MS delay (30 MS for 1 MHz)
II	3	S = Side Compare Flag	S = 0, Compare for side 0 S = 1, Compare for side 1
II	3	L = Sector Length Flag	LSB's Sector Length in ID Field: 00 / 01 / 10 / 11 L = 0: 256 / 512 / 1024 / 128 L = 1: 128 / 256 / 512 / 1024
II	4	m = Multiple Record Flag	m = 0, Single record m = 1, Multiple records
IV	0-3	I_x = Interrupt Condition Flags I_0 = 1 Not Ready To Ready Transition I_1 = 1 Ready To Not Ready Transition I_2 = 1 Index Pulse I_3 = 1 Immediate Interrupt, Requires A Reset* I_3-I_0 = 0 Terminate With No Interrupt (INTRQ)	

*NOTE: See Type IV Command Description for further information.

Write Precompensation

When operating in Double Density mode ($\overline{DDEN}$ = 0), the 279X has the capability of providing a user-defined precompensation value for Write Data. An external potentiometer (10K) tied to the WPW signal (Pin 33) allows a setting of 100 to 300 ns from nominal.

Setting the Write precomp value is accomplished by forcing the $\overline{TEST}$ line (Pin 22) to a Logic 0. A stream of pulses can then be seen on the Write Data (Pin 31) line. Adjust the WPW Potentiometer for the desired pulse width. This adjustment may be performed in-circuit since Write Gate (Pin 30) is inactive while $\overline{TEST}$ = 0.

Data Separation

The 279X can operate with either an external data separator or its own internal recovery circuits. The condition of the $\overline{TEST}$ line (Pin 22) in conjunction with $\overline{MR}$ (Pin 19) will select internal or external mode.

To program the 279X for external VCO, a $\overline{MR}$ pulse must be applied while $\overline{TEST}$ = 0. A clock equivalent to eight times the data rate (e.g., 4.0 MHz for 8" Double Density) is applied to the VCO input (Pin 26). The feedback reference voltage is available on the Pump output (Pin 23) for external integration to control the VCO. $\overline{TEST}$ is returned to a Logic 1 for normal operation. Note: To maintain this mode, $\overline{TEST}$ must be held low whenever $\overline{MR}$ is applied.

For internal VCO operation, the $\overline{TEST}$ line must be high during the $\overline{MR}$ pulse, then set to a Logic 0 for the adjustment procedure.

A 50K Potentiometer tied to the RPW input (Pin 18) is used to set the internal Read Data pulse for proper phasing. With a scope on Pin 29 (TG43), adjust the RPW pulse for 1/8 of the data rate (250 ns for 8" Double Density). An external variable capacitor of 5-60 pf is tied to the VCO input (Pin 26) for adjusting center frequency. With a frequency counter on Pin 16 (DIRC) adjust the trimmer cap to yield the appropriate Data Rate (500 KHz for 8" Double Density). The $\overline{DDEN}$ line must be low while the 5/8 line is held high or the adjustment times above will be doubled.

After adjustments have been made, the $\overline{TEST}$ pin is returned to a Logic 1 and the device is ready for operation. Adjustments may be made in-circuit since the DIRC and TG43 lines may toggle without affecting the drive.

The PUMP output (Pin 23) consists of positive and negative pulses, which their duration is equivalent to the phase difference of incoming Data vs. VCO frequency. This signal is internally connected to the VCO input, but a Filter is needed to connect these pulses to a slow moving DC voltage.

The internal phase-detector is unsymmetrical for a random distribution of data pulses by a factor of two, in favor of a PUMP UP condition. Therefore, it is desirable to have a PUMP DOWN twice as responsive to prevent run-away during a lock attempt.

A first order lag-lead filter can be used at the PUMP output (Pin 23). This filter controls the instantaneous response of the VCO to bit-shifted data (jitter) as well as the response to normal frequency shift, i.e., the lock-up time. A balance must be accomplished between the two conditions to inhibit over-responsiveness to jitter and to prevent an extremely wide lock-up response, leading to PUMP run-away. The filter affects these two reactions in mutually opposite directions.

The Source Impedance for a PUMP UP/DOWN condition is 600/120 ohms, respectively, therefore the change in bias voltage for each pump can be approximated:

$$dV = \frac{dt \, \Delta V}{RC}$$

dt = 250 ns. (set by RPW)
C = 0.1μf
R = R_S + R
ΔV = 2.6 V for PUMP UP
0.9V for PUMP DOWN

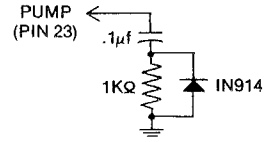
Look up response (T_L) is the transient time for the Loop to lock from center frequency (F_O) to maximum lock range:

$$T_L = 10\% \, F_L \times K_O \times \Delta P$$

Where:
K_O = VCO Conversion Gain = 3.7KHz/mV
F_L = Lock Range = 4.00 MHz
ΔP = Change in Bias for each Pump = 4 mV/PUMP

400 KHz x 3.7 KHz x 4 mV = 27 pumps
27 pumps = 54 μsec = 3.4 Byte times (8" Double Density)

The following Filter Circuit is recommended for 8" FM/MFM:



Since 5¼" Drives operate at exactly one-half the data rate (250 Kb/sec) the above capacitor should be doubled to .2 or .22μf.

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 as defined in Table 3.

A 2μs (MFM) or 4 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 12 μs before the first stepping pulse is generated.

The rates (shown in Table 3) can be applied to a Step-Direction Motor through the device interface.

TABLE 3. STEPPING RATES

CLK		2 MHz	1 MHz
R1	R0	$\overline{TEST}$ = 1	$\overline{TEST}$ = 1
0	0	3 ms	6 ms
0	1	6 ms	12 ms
1	0	10 ms	20 ms
1	1	15 ms	30 ms

After the last directional step an additional 15 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. Note that this time doubles to 30 ms for

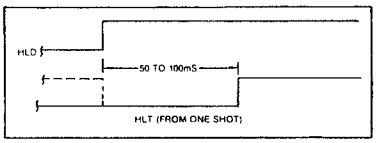
a 1 MHz clock. There is also a 15 ms head settling time if the E flag is set in any Type II or III command.

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 15 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. If there is a match but not a valid CRC, the CRC error status bit is set (Status bit 3), and the next encountered ID field is read from the disk for the verification operation.

The WD279X 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.

The Head Load (HLD) output controls the movement of the read/write head against the media. HLD is activated at the beginning of a Type I command if the h flag is set (h = 1), at the end of the Type I command if the verify flag (V = 1), or upon receipt of any Type II or III command. Once HLD is active it remains active until either a Type I command is received with (h = 0 and V = 0); or if the 279X is in an idle state (non-busy) and 15 index pulses have occurred.

Head Load timing (HLT) is an input to the 279X which is used for the head engage time. When HLT = 1, the 279X assumes the head is completely engaged. The head engage time is typically 30 to 100 ms depending on drive. The low to high transition on HLD is typically used to fire a one shot. The output of the one shot is then used for HLT and supplied as an input to the 279X.



HEAD LOAD TIMING

When both HLD and HLT are true, the 279X will then read from or write to the media. The "and" of HLD and HLT appears as status Bit 5 in Type I status.

In summary for the Type I commands: if h = 0 and V = 0, HLD is reset. If h = 1 and V = 0, HLD is set at the beginning of the command and HLT is not sampled nor is there an internal 15 ms delay. If h = 0 and V = 1, HLD is set near the end of the command, an internal 15 ms occurs, and the 279X waits for HLT to be true. If h = 1 and V = 1, HLD is set at the beginning of the command. Near the end of the command, after all the steps have been issued, an internal 15 ms delay occurs and the 279X then waits for HLT to occur.

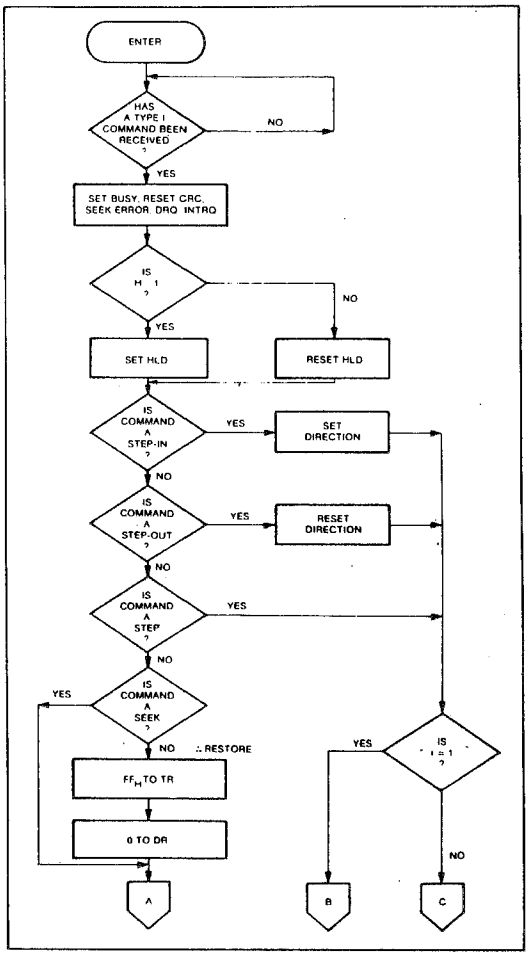
For Type II and III commands with E flag off, HLD is made active and HLT is sampled until true. With E flag on, HLD is made active, an internal 15 ms delay occurs and then HLT is sampled until true.

RESTORE (SEEK TRACK 0)

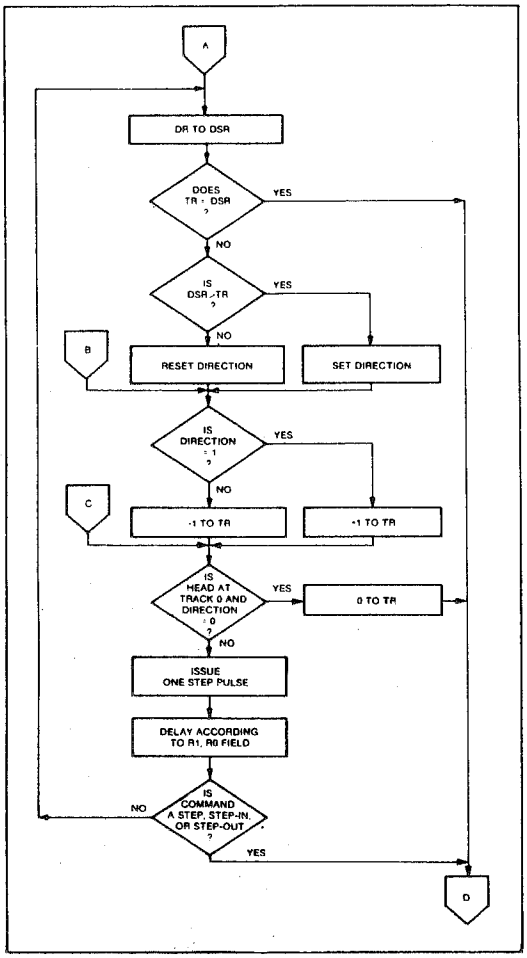
Upon receipt of this command the Track 00 ($\overline{TR00}$) input is sampled. If $\overline{TR00}$ is active low indicating the Read-Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TR00}$ is not active low, stepping pulses (pins 15 to 16) at a rate specified by the r1r0 field are issued until the $\overline{TR00}$ input is activated. 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 279X terminates operation, interrupts, and sets the Seek error status bit. A verification operation takes place if the V flag is set. The h bit allows the head to be loaded at the start of command. Note that the Restore command is executed when $\overline{MR}$ goes from an active to an inactive state.

SEEK

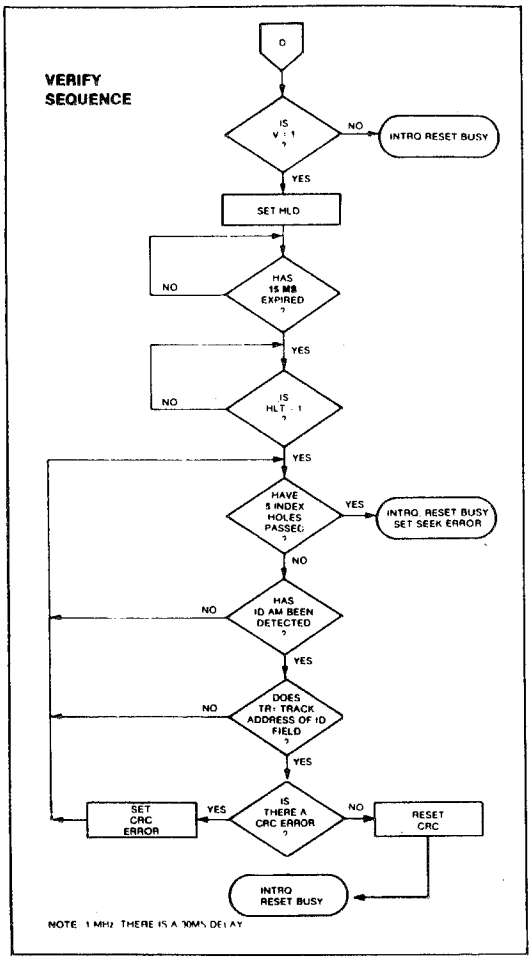
This command assumes that the Track Register contains the track number of the current position of the Read-Write head and the Data Register contains the desired track number. The WD279X will update the Track register and issue stepping pulses in the appropriate direction until the



TYPE I COMMAND FLOW



TYPE I COMMAND FLOW



TYPE I COMMAND FLOW

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 head to be loaded 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.

STEP

Upon receipt of this command, the 279X 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 r1r0 field, a verification takes place if the V flag is on. If the T flag is on, the Track Register is updated. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the 279X issues one stepping pulse in the direction towards track 76. If the T flag is on, the Track Register is incremented by one. After a delay determined by the r1r0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-OUT

Upon receipt of this command, the 279X issues one stepping pulse in the direction towards track 0. If the T flag is on, the Track Register is decremented by one. After a delay determined by the r1r0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

EXCEPTIONS

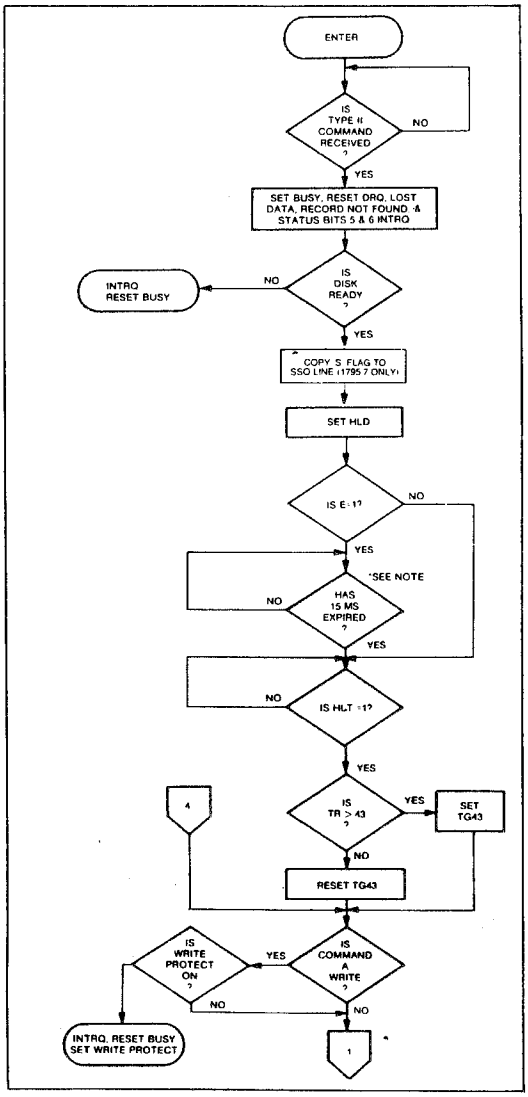
On the 2795/7 devices, the SSO output is not affected during Type I commands, and an internal side compare does not take place when the (V) Verify Flag is on.

TYPE II COMMANDS

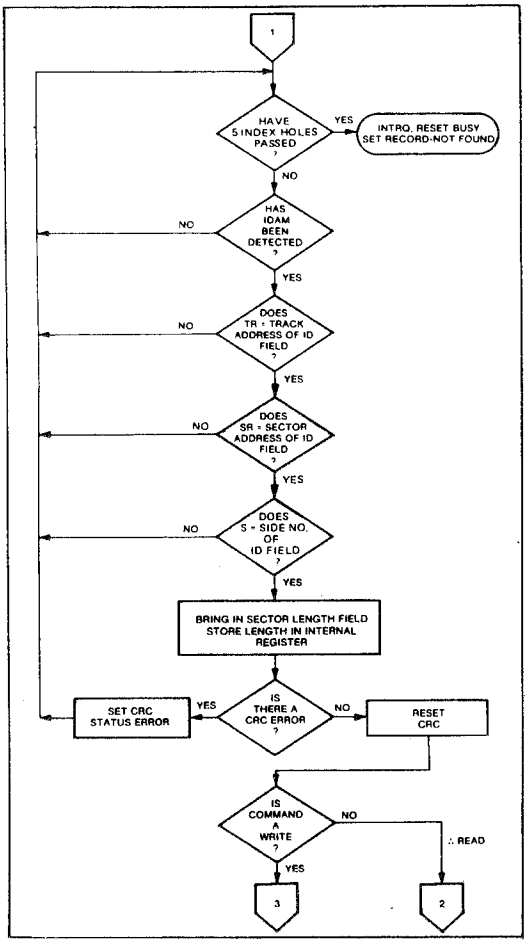
The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the

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 (this is the normal case) HLD is made active and HLT is sampled after a 15 msec delay. If the E flag is 0, the head is loaded and HLT sampled with no 15 msec delay.

When an ID field is located on the disk, the 279X 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



TYPE II COMMAND

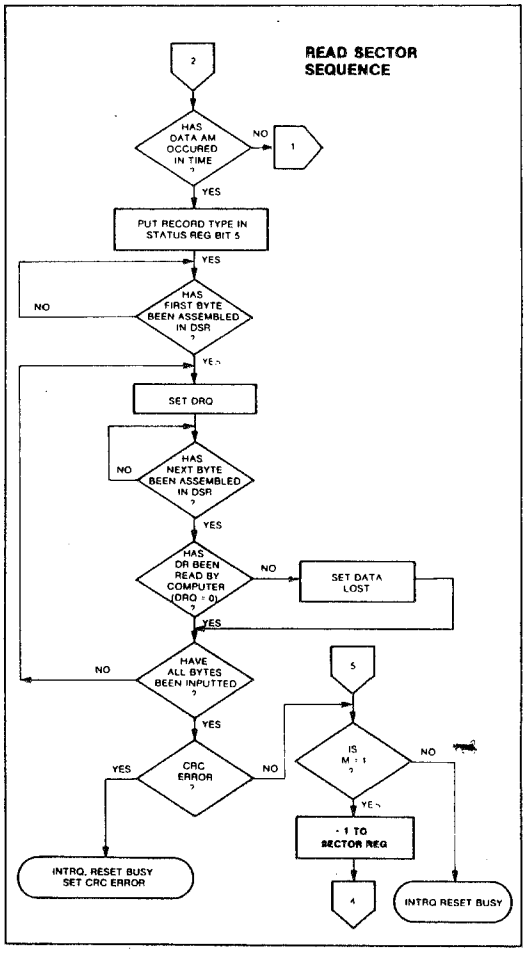


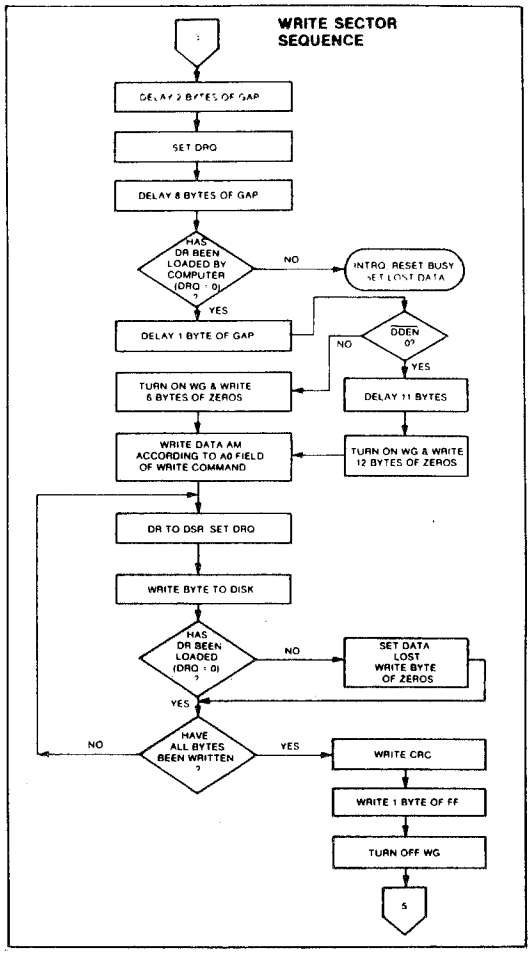
TYPE II COMMAND

depending upon the command. The 279X must find an ID field with a Track number, Sector number, side number, and CRC within 5 revolutions of the disk; otherwise, the Record not found status bit is set (Status bit 4) and the command is terminated with an interrupt.

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 279X 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 279X is instructed to read sector 27 and there are only 26 on the track, the sector register exceeds





TYPE II COMMAND

the number available. The 279X will search for 5 disk revolutions, interrupt out, reset busy, and set the record not found status bit.

The Type II commands for 2791-93 also contain side select compare flags. When C = 0 (Bit 1) no side comparison is made. When C = 1, the LSB of the side number is read off the ID Field of the disk and compared with the contents of the (S) flag (Bit 3). If the S flag compares with the side number recorded in the ID field, the 279X continues with the ID search. If a comparison is not made within 5 index pulses, the interrupt line is made active and the Record-Not-Found status bit is set.

The Type II and III commands for the 2795-97 contain a side select flag (Bit 1). When U = 0, SSO is updated to 0. Similarly, U = 1 updates SSO to 1. The chip compares the SSO to the ID field. If they do not compare within 5 revolutions the interrupt line is made active and the RNF status bit is set.

The 2795/7 READ SECTOR and WRITE SECTOR com-

TYPE II COMMAND

mands include a 'L' flag. The 'L' flag, in conjunction with the sector length byte of the ID Field, allows different byte lengths to be implemented in each sector. For IBM compatibility, the 'L' flag should be set to a one.

READ SECTOR

Upon receipt of the Read Sector command, the head is loaded, the Busy status bit set, and when an ID field is encountered that has the correct track number, correct sector number, correct side 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 search is repeated.

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 sector 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 head is loaded (HLD active) and the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, a DRQ is generated. The 279X 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 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 279X 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 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 bye of FE in FM or in MFM. The WG output is then deactivated. For a 2 MHz clock the INTRQ will set 8 to 12 μsec after the last CRC byte is written. For partial sector writing, the proper method is to write the data and fill the balance with zeroes. By letting the chip fill the zeroes, errors may be masked by the lost data status and improper CRC Bytes.

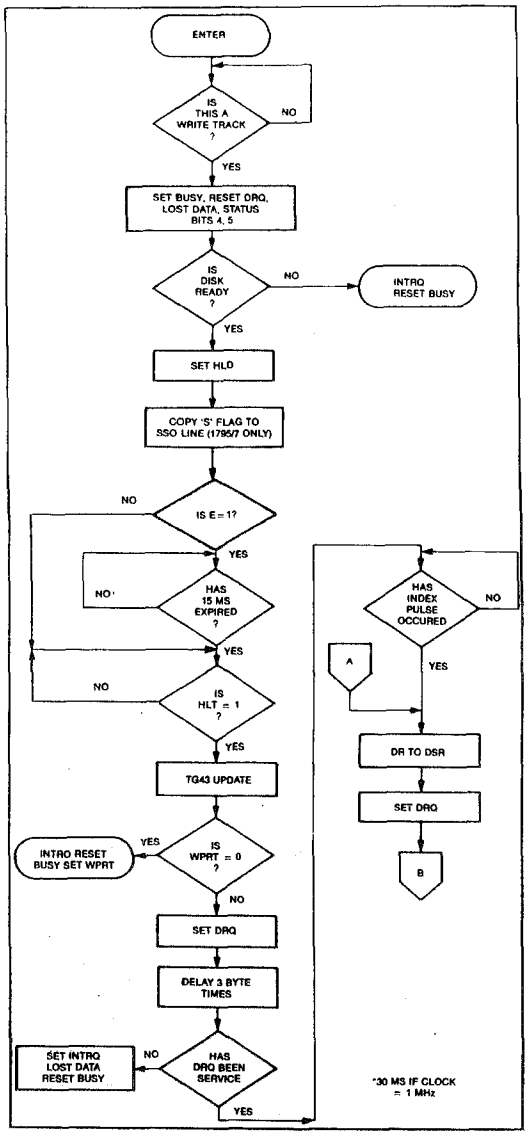
TYPES III COMMANDS

READ ADDRESS

Upon receipt of the Read Address command, the head is loaded and the Busy Status Bit is set. The next encountered ID field is then read in from the disk, and the 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

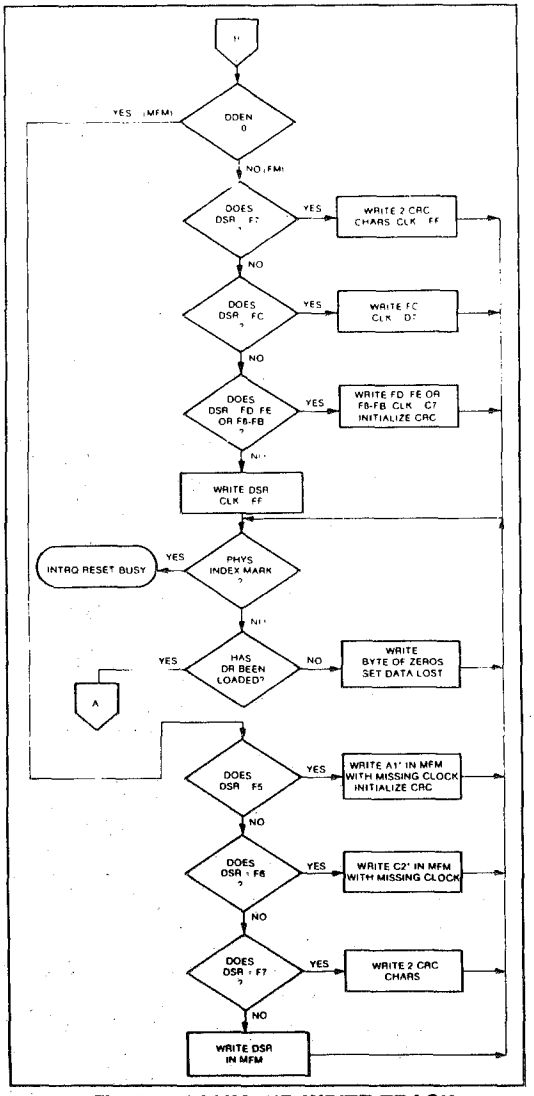
Although the CRC characters are transferred to the computer, the 279X checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register so that a comparison can be made by the host. At the end of the operation an interrupt is generated and the Busy Status is reset.



TYPE III COMMAND WRITE TRACK

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 ac-



TYPE III COMMAND WRITE TRACK

cumulation 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; the internal side compare is not performed; and the address mark detector is on for the duration of the command. Because the A.M. detector is always on, write splices or noise may cause the chip to look for an A.M. If an address mark does not appear on schedule with the Lost Data status flag being set.

The ID A.M., 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 head is loaded and 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 by the time the index pulse is encountered 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 279X detects a data pattern of F5 thru FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation.

The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR

CONTROL BYTES FOR INITIALIZATION

DATA PATTERN IN DR (HEX)	WD279X INTERPRETATION IN FM ($\overline{DDEN}$ = 1)	WD279X INTERPRETATION IN MFM ($\overline{DDEN}$ = 0)
00 thru F4	Write 00 thru F4 with CLK = FF	Write 00 thru F4, in MFM
F5	Not Allowed	Write A1* in MFM, Preset 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

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 thru 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.

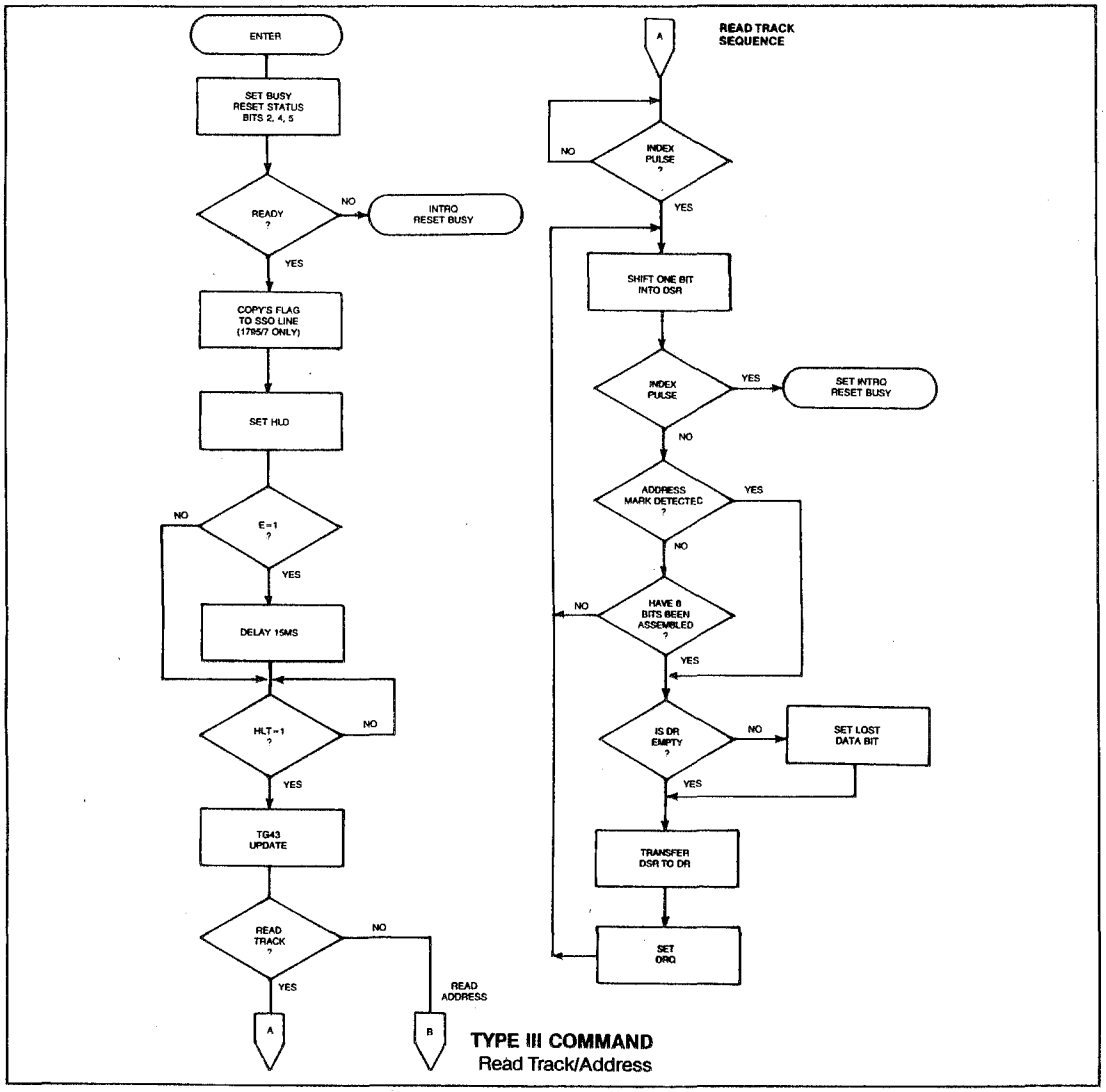
TYPE IV COMMANDS

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 = Not-Ready to Ready Transition
I_1 = Ready to Not-Ready Transition
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



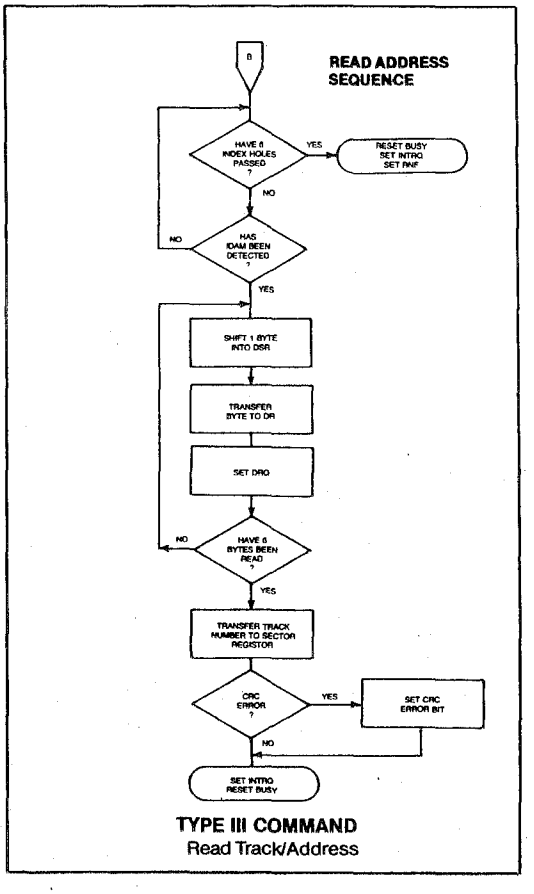
TYPE III COMMAND
Read Track/Address

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 8 micro sec (double density) or 16 micro sec (single density) before issuing a new command after issuing a forced interrupt (times double when clock = 1 MHz). 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.)

More than one condition may be set at a time. If for example, the READY TO NOT-READY condition (I_1 = 1) and the Every Index Pulse (I_2 = 1) are both set, the resultant command would be HEX "DA." The "OR" function is performed so that either a READY TO NOT-READY or the next Index Pulse will cause an interrupt condition.



TYPE III COMMAND
Read Track/Address

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

Status varies according to the type of command executed as shown in Table 4.

Because of internal sync cycles, certain time delays must be observed when operating under programmed I/O. They are: (times double when clock = 1 MHz)

Operation	Next Operation	Delay Req'd. FM	MFM
Write to Command Reg.	Read Busy Bit (Status Bit 0)	12μs	6μs
Write to Command Reg.	Read Status Bits 1-7	28μs	14μs
Write Any Register	Read From Diff. Register	0	0

IBM 3740 FORMAT — 128 BYTES/SECTOR

Shown below is the IBM 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)[3]
6	00
1	FC (Index Mark)
1 26	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)
27	FF (or 00)
247[2]	FF (or 00)

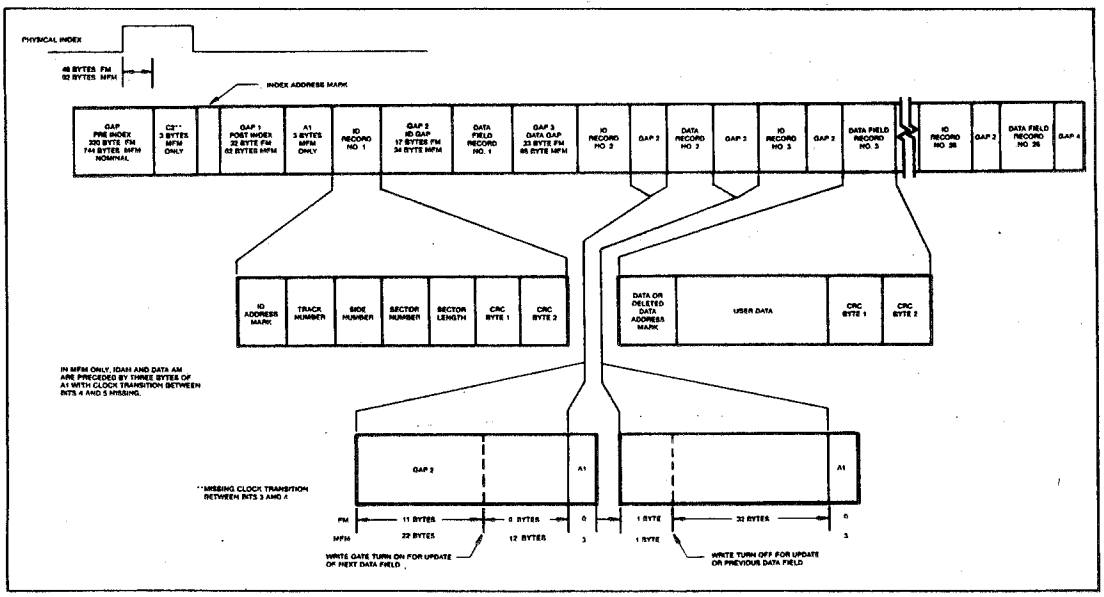
1. Write bracketed field 26 times
2. Continue writing until 279X interrupts out. Approx. 247 bytes.
3. A '00' option is allowed on 2795/7 only.

IBM SYSTEM 34 FORMAT-256 BYTES/SECTOR

Shown below is the IBM dual-density format with 256 bytes/sector. In order for 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
80	4E
12	00
3	F6 (Writes C2)
1	FC (Index Mark)
* 50	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 CRCs written)
22	4E
12	00
3	F5 (Writes A1)
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
598**	4E

* Write bracketed field 26 times
** Continue writing until 279X interrupts out. Approx. 598 bytes.



IBM TRACK FORMAT

1. NON-IBM FORMATS

Variations in the IBM 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 IBM format.
3) 3 bytes of A1 must be used in MFM.

In addition, the Index Address Mark is not required for operation by the 279X. Gap 1, 3, and 4 lengths can be as short as 2 bytes for 279X 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 IBM format be used for highest system reliability.

	FM	MFM
Gap I	16 bytes FF	32 bytes 4E
Gap II	11 bytes FF	22 bytes 4E
*	6 bytes 00	12 bytes 00
*		3 bytes A1
Gap III**	10 bytes FF	24 bytes 4E
	4 bytes 00	8 bytes 00
		3 bytes A1
Gap IV	16 bytes FF	16 bytes 4E

* Byte counts must be exact.
** Byte counts are minimum, except exactly 3 bytes of A1 must be written.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Voltage to any input with respect to V_{SS} = +15 to −0.3V

C_{IN} & C_{OUT} = 15 pF max with all pins grounded except one under test.
Operating temperature = 0°C to 70°C
Storage temperature = −55°C to +125°C

NOTE: Maximum limits indicate where permanent device damage occurs. Continuous operation at these limits is not intended and should be limited to those conditions specified in the DC Electrical characteristics.

OPERATING CHARACTERISTICS (DC)

T_A = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5M ± .25V

SYMBOL	CHARACTERISTIC	MIN.	TYP.	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 = −100μA
V_{OL}	Output Low Voltage			0.45	V	I_O = 1.6 mA
V_{OHP}	Output High PUMP	2.2			V	I_{OP} = −1.0 mA
V_{OLP}	Output Low PUMP			0.2	V	I_{OP} = +1.0 mA
P_D	Power Dissipation			.75	W	All Outputs Open
R_{PU}	Internal Pull-up*	100		1700	μA	V_{IN} = 0V
I_{CC}	Supply Current		70	150	mA	All Outputs Open

* Internal Pull-up resistors on PINS 1, 17, 22, 25, 37, and 40.

TIMING CHARACTERISTICS

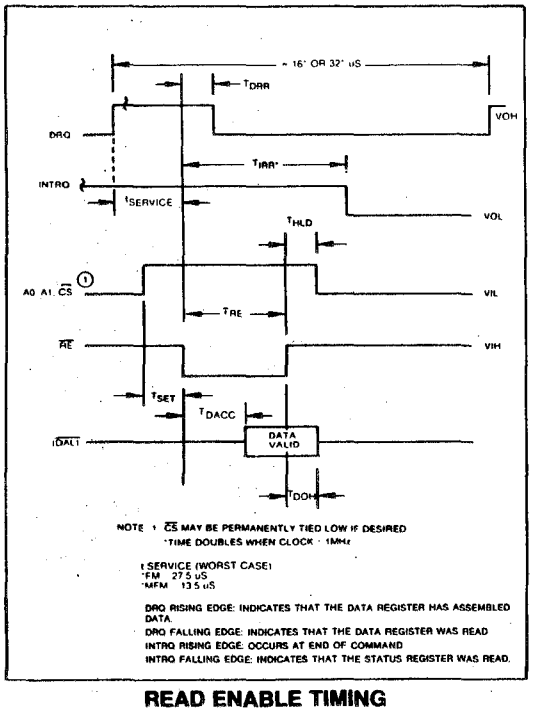
T_A = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5V ± .25V

READ ENABLE TIMING

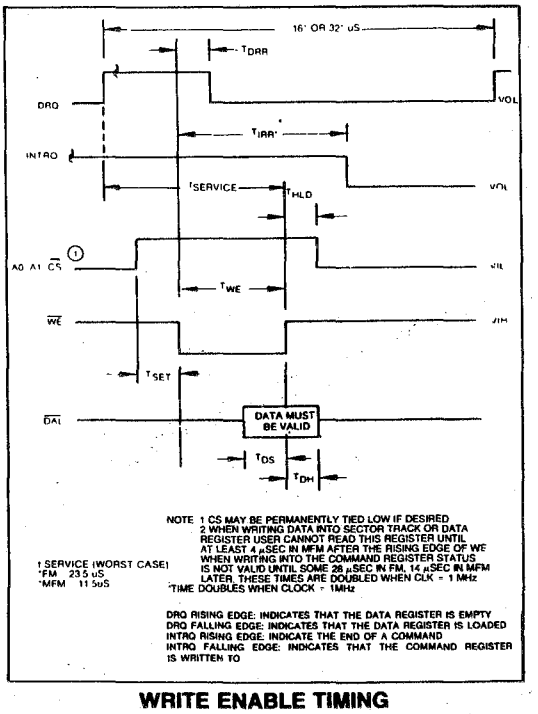
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{SET}	Setup ADDR & CS to $\overline{RE}$	50			nsec	
T_{HLD}	Hold ADDR & CS from $\overline{RE}$	10			nsec	
T_{RE}	$\overline{RE}$ Pulse Width	200			nsec	C_L = 50 pf
T_{DRR}	DRQ Reset from $\overline{RE}$		100	200	nsec	
T_{IRR}	INTRQ Reset from $\overline{RE}$		500	3000	nsec	See Note
T_{DACC}	Data Valid from $\overline{RE}$		100	200	nsec	C_L = 50 pf
T_{DOH}	Data Hold From $\overline{RE}$	20		150	nsec	C_L = 50 pf

WRITE ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{SET}	Setup ADDR & CS to $\overline{WE}$	50			nsec	
T_{HLD}	Hold ADDR & CS from $\overline{WE}$	10			nsec	
T_{WE}	$\overline{WE}$ Pulse Width	200			nsec	
T_{DRR}	DRQ Reset from $\overline{WE}$		100	200	nsec	
T_{IRR}	INTRQ Reset from $\overline{WE}$		500	3000	nsec	See Note
T_{DS}	Data Setup to $\overline{WE}$	150			nsec	
T_{DH}	Data Hold from $\overline{WE}$	50			nsec	



READ ENABLE TIMING



WRITE ENABLE TIMING

INPUT DATA TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{PW}	$\overline{\text{Raw Read}}$ Pulse Width	100	200		nsec	
T_{BC}	$\overline{\text{Raw Read}}$ Cycle Time	1500	2000		nsec	

WRITE DATA TIMING: (ALL TIMES DOUBLE WHEN CLK = 1 MHz)

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{WP}	Write Data Pulse Width	400	500	600	nsec	FM
		240		1000	nsec	MFM
T_{WG}	Write Gate to Write Data		2		µsec	FM
			1		µsec	MFM
T_{WF}	Write Gate off from WD		2		µsec	FM
			1		µsec	MFM

MISCELLANEOUS TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{CD_1}	Clock Duty (low)	230	250	20000	nsec	
T_{CD_2}	Clock Duty (high)	230	250	20000	nsec	
T_{STP}	Step Pulse Output	2 or 4			µsec	See Note
T_{DIR}	Dir Setup to Step		12		µsec	± CLK ERROR
T_{MR}	Master Reset Pulse Width	50			µsec	
T_{IP}	Index Pulse Width	10			µsec	See Note
RPW	Read Window Pulse Width					Input 0-5V
		120		700	nsec	MFM
		240		1400	nsec	FM ± 15%
WPW	Write Data Pulse Width					Input 0-5V
		300		1000	nsec	MFM
			500		nsec	FM
	Precomp Adjust.	100		250	nsec	MFM
RPW	Read Window Pulse Width					Input 0-5V
		120		700	nsec	MFM
		240		1400	nsec	FM ± 15%
WPW	Write Data Pulse Width					Input 0-5V
		300		1000	nsec	MFM
			500		nsec	FM
	Precomp Adjust.	100		250	nsec	MFM
VCO	Free Run Voltage Controlled	6.0			MHz	Ext. C = 0
	Oscillator. Adjustable by ext. capacitor on Pin 26		4.0		MHz	Ext. C = 35 pf
	Pump Up + 25%	5.0			MHz	PU = 2.2V Cext = 35 pf
VCO	Pump Down − 25%			3.0	MHz	$\overline{\text{PD}}$ = 0.2V Cext = 35 pf
VCO	5% Change V_{CC}	3.8		4.2	MHz	Cext = 35 pf
	T_A = 75°C	3.5			MHz	Cext = 35 pf
Cext	Necessary external capacitor	10	35	80	pf	VCO = 4.0MHz nom
RCLK	Derived read clock = VCO ÷ 8, 16, 32					VCO = 4.0MHz
			500		KHz	$\overline{\text{DDEN}}$ = 0 $\overline{5/8}$ = 1
			250		KHz	$\overline{\text{DDEN}}$ = 0 $\overline{5/8}$ = 0
			250		KHz	$\overline{\text{DDEN}}$ = 1 $\overline{5/8}$ = 1
			125		KHz	$\overline{\text{DDEN}}$ = 1 $\overline{5/8}$ = 0
PU/DON	PU/$\overline{\text{PD}}$ time on (pulse width)			250	ns	MFM
				500	ns	FM

APPENDIX G

FLOPPY DISK ERROR CODES

Table 4. STATUS REGISTER SUMMARY

BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ SECTOR	READ TRACK	WRITE SECTOR	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	0	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD LOADED	0	RECORD TYPE	0	0	0
S4	SEEK ERROR	RNF	RNF	0	RNF	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX PULSE	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

BIT NAME	MEANING
S7 NOT READY (40)	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED (30)	When set, indicates Write Protect is activated. This bit is an inverted copy of $\overline{WRPT}$ input.
S5 HEAD LOADED (20)	When set, it indicates the head is loaded and engaged. This bit is a logical "and" of HLD and HLT signals.
S4 SEEK ERROR (10)	When set, the desired track was not verified. This bit is reset to 0 when updated.
S3 CRC ERROR (4)	CRC encountered in ID field.
S2 TRACK 00 (3)	When set, indicates Read/Write head is positioned to Track 0. This bit is an inverted copy of the $\overline{TROO}$ input.
S1 INDEX (2)	When set, indicates index mark detected from drive. This bit is an inverted copy of the $\overline{IP}$ input.
S0 BUSY (1)	When set command is in progress. When reset no command is in progress.

1. In certain cases the SUPER SIX must be run at 4MHz. This is because timing on the S-100 bus is critical when interfacing with select other S-100 bus boards.

For the users of Super Six computer board the following changes apply to any Super Six up to Rev. "M":

1. Add a 1 K Resistor (pull up) to the BRQ output of U21 (D.M.A.).

2. Change the value of R11 and R16 from 1 K each to 2 K, for more V.C.O. range.

3. Connect Phantom to U33 pin 1, for all memory to be turned OFF when active.

And for Super Six users running at 6 MHZ with up to Rev. "M" board, using any other non Advanced Digital boards that has one or the two next characteristics.

1. An I/O board latches the incoming data on the falling edge of (pWR*) during an I/O transfer of data.

2. A memory transfer where the signal (pSTVAL*) is expected to fall inside of (PSYNC) active.

Modify your Super Six, through all the rework instructions, else do only first three steps.

Rework Instructions:

1. Locate U15 on the solder side, connect a 1 K resistor across from U15 pin 14 to U15 pin 9.

2. Remove R11 and R16 next to (L.E.D.1) both a 1 K and replace with a 2 K resistor for R11 and R16.

3. Locate U33 on the solder side cut trace between U33 pin 1 and 2 and jumper from U33 pin 1 to U33 pin 10.

4. Locate U71 on the solder side and between pin 1 and 20 cut trace to the feed through. Now install a 14 pin socket at U1 (spare ic) with pin 1 on 1 jumper pin 7 of the socket to hole number 8, for ground into the IC. Jumper U71 pin 8 to U1 pin 9. Locate U68 on solder side, jumper U68 pin 2 to U1 pins 12 and 13. Locate U6 and jumper pin 11 to U1 pin 11. Locate U6A and jumper pin 3 to U1 pin 10. Remove I.C. U6A (74LS32) and lift pin 1, bend it so won't sort out. Jumper pin 1 to pin 4 on component side over I.C., install back I.C. on solder side jumper U6A pin 2 to U25 Pin 4.

5. Locate U28 component side, lift I.C. and pull out pin 9 install back I.C. Locate U4 and U5 solder side. Jumper U4 pin 12 to U5 pin 13. Jumper U5 pin 12 to U1 pin 2 and 1. Jumper U4 pin 10 to U1 pin 5. Jumper U1 pin 3 to U2 pin 1. Locate U21 (D.M.A.) jumper U21 pin 12 to U1 pin 4.

6. Install a 74S74 in U1 socket.

APPENDIX I

WARRANTY LETTER

WARRANTY NOTICE

Advanced Digital Corporation now requires a Return Authorization Number for the return of any equipment for repair or credit. This number will be issued by the Customer Support Department. Any equipment received without the Return Authorization Number clearly marked on the outside of the package may be subject to significant delays in the repair process.

Return Authorization Numbers are active for 30 days after they are issued. If the equipment specified in the Return Authorization is not received by Advanced Digital within this 30 day period significant delays in handling the repair may be incurred.

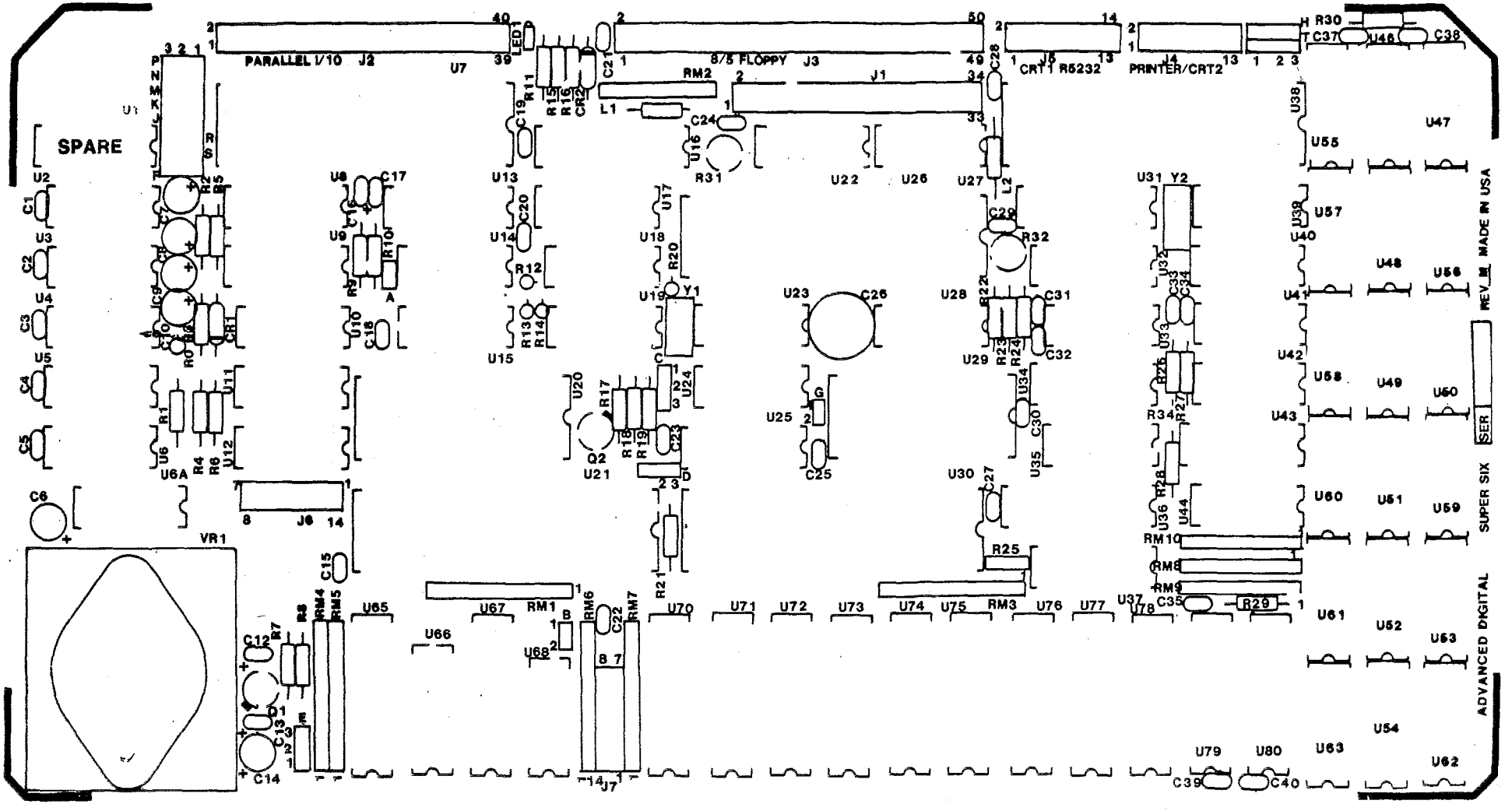
If the equipment must be returned a second time, a new Return Authorization Number must be issued. Reuse of Return Authorization Numbers may result in delays in processing returns.

Effective November 1, 1981, repair of all kit and nonwarranty boards will be $70.00. This fee is subject to change without notice.

Returns for credit will be subject to a 15% restocking charge. If material for credit was purchased through a dealer, Advanced Digital cannot issue a credit. Adjustment must be handled through the dealer. Other credit returns should have reference to the original invoice number.

If you have any questions regarding special handling, packaging of the equipment, or procedures for returning equipment, please contact the Advanced Digital Corporation.

The warranty on the super SIX is one year from the date of purchase.


SPARE
PARALLEL I/10 J2
8/5 FLOPPY J3
CRT1 R5232
PRINTER/CRT2
J1
J6
J7
VR1
LED1
L1
L2
Y1
Y2
Q2
D1
RM1
RM2
RM3
RM4
RM5
RM6
RM7
RM8
RM9
RM10
SUPER SIX
ADVANCED DIGITAL
MADE IN USA
REV. M
SER

SUPER SIX SINGLE BOARD COMPUTER PARTS LIST

ITEM	PART NO.	LOCATION	QTY.
1	74S 10	U2	1
2	74LS00	U3, U11, U44	3
3	74LS132	U4	1
4	74LS04	U5, U68	2
5	74 S74	U6, U28	2
6	74LS32	U6-A, U40	2
7	Z80B PIO	U7	1
8	CD4098	U8	1
9	74LS138	U9, U34	2
10	7438	U10	1
11	74LS13	U12, U33	2
12	74LS123	U13	1
13	74LS14	U14	1
14	7407	U15, U22	2
15	16L8	U16, U36	2
16	74LS174	U17	1
17	74LS11	U18	1
18	74S04	U19	1
19	Z80B CTC	U20	1
20	Z80 DMA	U21	1
21	74LS92	U23	1
22	74LS393	U24	1
23	8T97	U25	1
24	7406	U26	1
25	WD2793	U27	1
26	2726 EPROM	U29	1
27	Z80B	U30	1
28	BR1945	U31	1
29	74LS109	U32	1
30	74LS27	U35	1
31	74S287	U37	1
32	Z80B DART	U38	1
33	74LS280	U39	1
35	8T98	U43, U66	2
36	TTLDM100	U41	1
37	74LS02	U42	1
38	4164-15	U46-U54	18
39	74LS244	U65	1
40	74LS374	U67	1
41	74LS240	U70, U71, U77	3
42	74LS373	U73	1
43	74LS245	U74, U76	2
44	74LS273	U78	1
45	74S240	U80, U79	2

LIST OF DISCRETE COMPONENTS

ITEM	PART NO.	LOCATION	QTY.
46	40 PIN SOCKET	U7, U21, U27, U30, U38	5
47	28 PIN SOCKET	U20	1
48	24 PIN SOCKET	U29	1
49	20 PIN SOCKET		15
50	18 PIN SOCKET	U31	1
51	16 PIN SOCKET		28
52	14 PIN SOCKET		27
53	220/330 SIP	RM2	1
54	10K SIP	RM1, RM4, RM5, RM3	4
55	4.7K SIP	RM6, RM7	2
56	33 OHM SIP (8 PIN)	RM8, RM9, RM10	3

CAPACITOR LIST FOR SUPER SIX

ITEM	PART NO.	LOCATION	QTY.
57	.1 MF CAP		30
58	10 MR CAP	C14, C6	2
59	4.7 MF CAP(25V)	C9, C10	2
60	6.8 MF (10V)	C7, C8	2
61	33 PF	C23	1
62	47 PF	C35, C32, C31, C33	4
63	5-50 PF CAP	C26	1

RESISTOR LIST FOR SUPER SIX (ALL RESISTORS ARE 1/4 WATT)

ITEM	PART NO.	LOCATION	QTY.
64	1K RESISTOR	R7, 8, 11, 15, 16	8
65	220 OHM	R12, 18, 22, 23, 24, 29, 21, 20 26, 27, 28	8
66	4.7K	R1	1
67	10K	R6, R10	2
68	27K	R3	1
69	470K	R2	1
70	5.1 MEGA OHM	R5	1
71	10 OHM	R17	1
72	1.5K	R13	1
73	560 OHM	R14	1
74	33 OHM	R30, R31	2
75	10K TRIM POT (82PR10K)		1
76	50K TRIM POT (82PR50K)		1

ACCESSORIES

ITEM	PART NO.	LOCATION	QTY.
77	220MH CHOCK	L1, L2	2
78	6104 TERMALOY	VR1	1
79	78H05		1
81	40 PIN CONN	J2	1
82	50 PIN CONN	J3	1
83	34 PIN CONNEC	J1	1
84	14 PIN CONNEC	J4, J5, J6	3
85	16 PIN CONNEC	RS (8 PIN DOUBLE)	1
86	8 PIN SINGLE	RS	1
87	3 PIN SINGLE	E, C, D, R	4
88	2 PIN SINGLE	B, G	2
89	DIALIGHT 5552222 LED		1
90	2N3906 TRANS	Q2	1
91	2N2222 TRANS	Q1	1
92	24 MHZ XTAL	Y1	1
93	5.068 MHZ XTA	Y2	1
94	1N914 DIODE	CR1, CR@	2

APPENDIX K

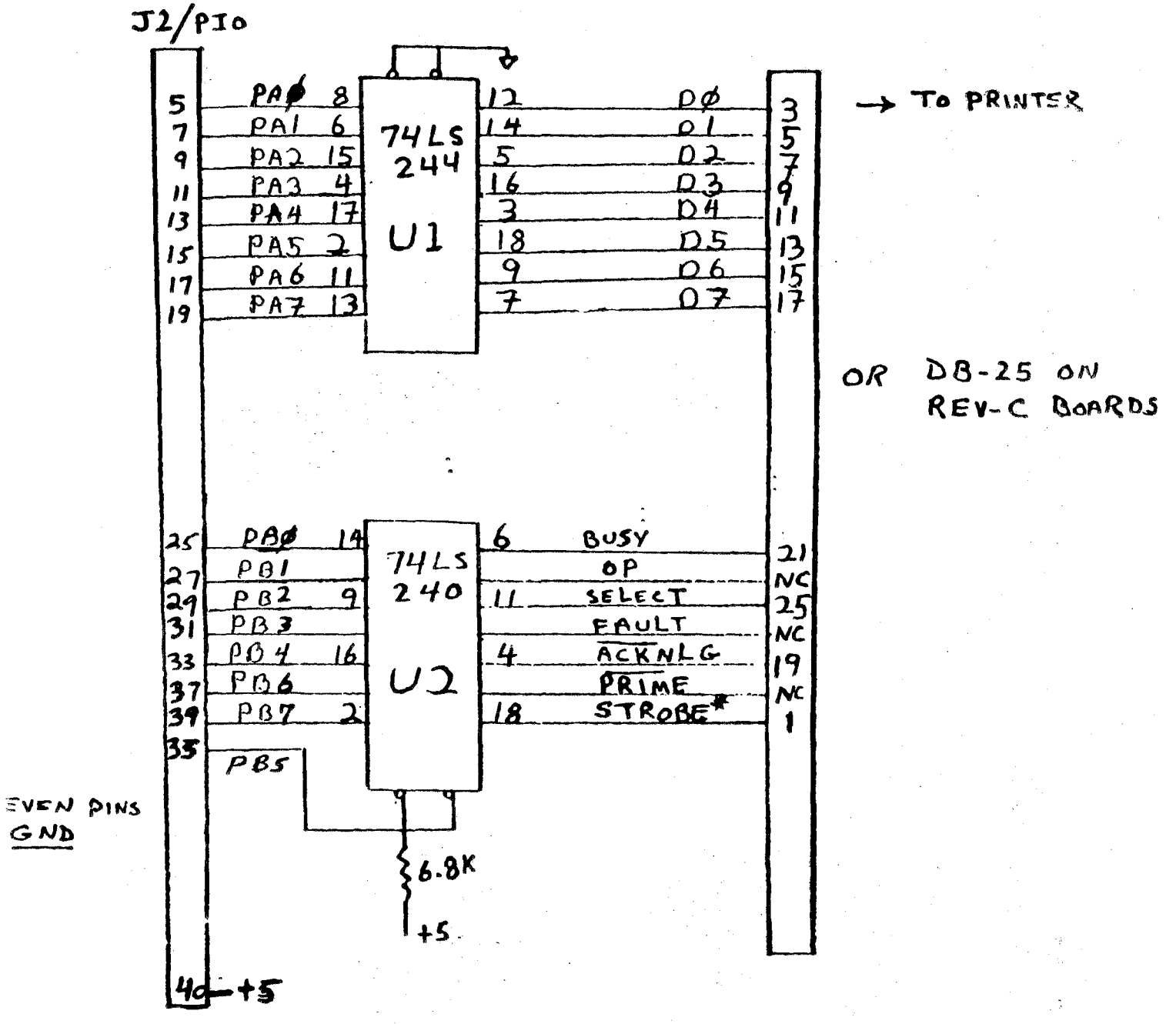
PLL REALIGNMENT

Following is the procedure for realigning the PLL on the SUPER SIX single board computer.

1. Turn POWER switch to ON (insure test jumper is removed).
2. Install Jumper G.
3. Adjust trimmer Cap for 250KHz on pin 16 of the FDC.
4. Adjust 50KHz trimmer pot for a positive going pulse of 250ns on pin 29 of the FDC.
5. Adjust 10KHz trimmer pot for a positive going pulse of about 125ns (100ns - 300ns, depending on disk drive manufacturer) on pin 31 of the FDC.
6. Remove jumper G.

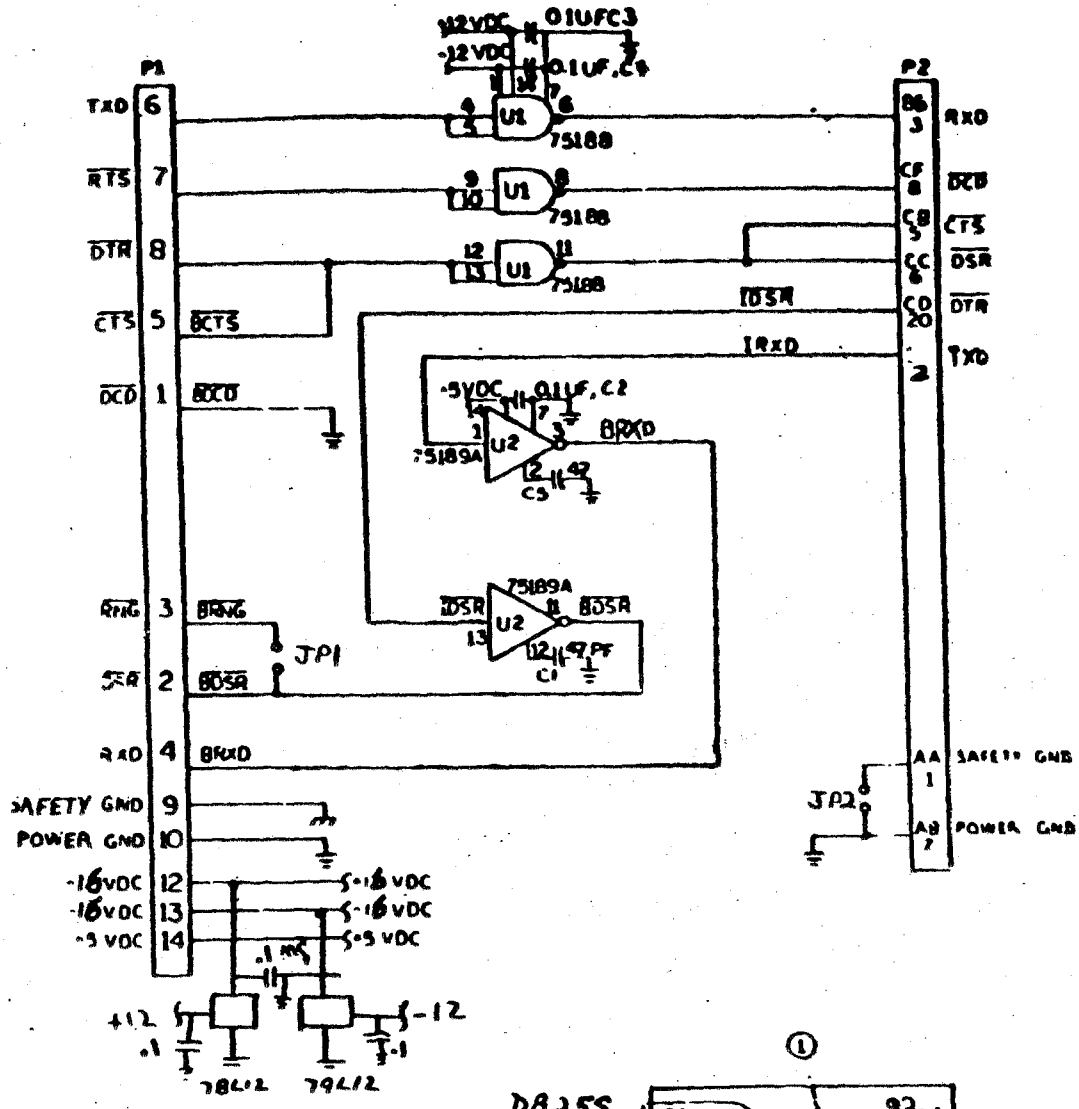
APPENDIX M

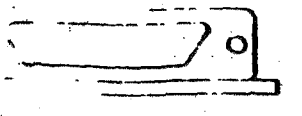
PSNET/PAR SCHEMATIC

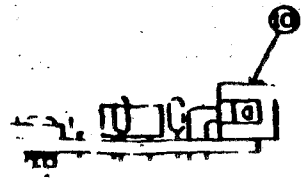


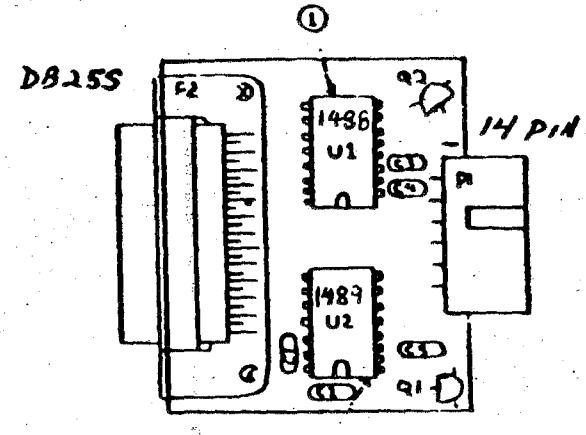
PS NET/PAR

3-25-92







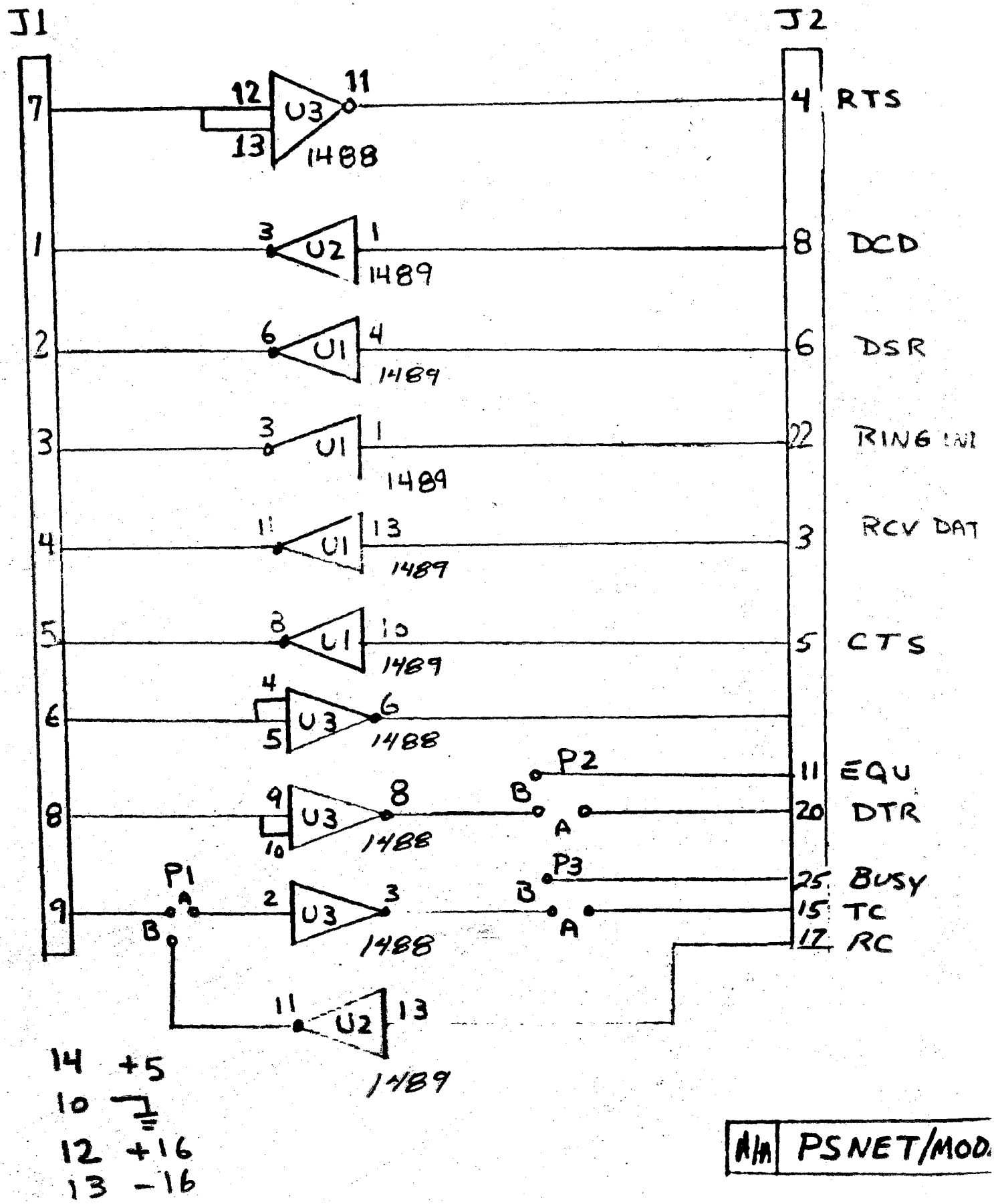


ENG. H/A

10-1-81

PS. NET/1

ADVANCED DIGITAL


J1
J2
7
12
13
U3
11
1488
4
RTS
1
3
U2
1
1489
8
DCD
2
6
U1
4
1489
6
DSR
3
3
U1
1
1489
22
RING IN
4
11
U1
13
1489
3
RCV DAT
5
8
U1
10
1489
5
CTS
6
4
5
U3
6
1488
8
9
U3
8
10
1488
P2
B
A
11
EQU
20
DTR
P3
B
A
25
BUSY
15
TC
17
RC
9
P1
A
B
2
U3
3
1488
11
U2
13
1489
14 +5
10
12 +16
13 -16
A/M
PSNET/MOD

APPENDIX N

SUPER SIX SCHEMATICS