

***THE WORLD'S FIRST . . . .***

**S-100 Single Board Computer**

**TECHNICAL MANUAL**

for

**SUPER QUAD**



12700-B Knott Avenue, Garden Grove, CA 92641  
TELE: (714) 891-4004 / TELEX: 678401 Tabirin

## INTRODUCTION

-----

ADVANCED DIGITAL CORPORATION is proud to introduce the SUPER QUAD. The SUPER QUAD is a Z80 based single board computer designed to be a bus master in an S100 bus system. The SUPER QUAD SBC has all the hardware needed to run a single user CP/M system or 2 user MP/M system with up to 4 external floppy disk drives and an external Centronics parallel interface printer all on one board.

The board also runs with TURBO-DOS tm.

The SUPER QUAD SBC contains:

- 1) Z-80A cpu (4 MHZ)
- 2) Floppy disk controller (up to 4 drives 8" or 5¼")
- 3) 64K of dynamic memory (16K bank selectable)
- 4) 2K or 4K of shadow eprom (2716 or 2732)
- 5) 2 serial ports (Z80A SIO opt. synchronous)
- 6) 2 12 bit parallel ports, (Z80A PIO)  
one of which can be used  
for S100 vectored interrupts
- 7) Real time interrupt clock (Z80A CTC)
- 8) S100 extended address A16-A23

ONE YEAR WARRANTY.

\*Note: Items 5 and 6 require external adaptation for RS-232 and Centronics. The adapter boards are 2 x 2" and are called PS NET. They hoop up to the back of the main frame with a DB-25 connector.

TABLE OF CONTENTS

---

	Page
Introduction .....	1
Table of Contents .....	2,3
1.0 Operation .....	
1.1 Floppy disk .....	4
1.2 64 K Dynamic Ram .....	4
1.3 Monitor Eprom .....	5
1.4 Serial Ports .....	5
1.5 Parallel Ports .....	5
1.6 Real Time Clock .....	6
1.7 S100 bus interface .....	6
2.0 EPROM and Monitor .....	7
2.1 Enable / Disable software.....	7
2.2 Monitor Sign-on.....	8
2.3 Monitor Commands .....	8
2.4 Cold boot loader program .....	9
3.0 Input / Output port assignments.....	10
4.0 Input / Output port description .....	11
4.1 SIO serial port channel A .....	11
4.2 SIO serial port channel B .....	11
4.3 PIO parallel port channel A .....	11
4.4 PIO parallel port channel B .....	11
4.5 Control timer Interrupt circuit .....	11
4.6 Floppy Disk controller .....	11
4.7 Floppy disk control port .....	12
4.8 Extended address port .....	13
4.9 On-board memory control port .....	14
5.0 Jumper definitions .....	15
6.0 Jumper descriptions .....	15
6.1 Jumper A - CPU clock rate .....	
6.2 Jumper B - SIO channel A clock .....	
6.3 Jumper C - SIO channel B clock .....	
6.4 Jumper D - Drive type selection .....	
6.5 Jumper E - V10 / PINT .....	16
6.6 Jumper P - V10 / paralalled bit D0 .....	16
6.7 Jumper N - V11 / paralalled bit D1 .....	17
6.8 Jumper M - V12 / paralalled bit D2 .....	18
6.9 Jumper K - V13 / paralalled bit D3 .....	18
6.10 Jumper J - V14 / paralalled bit D4 .....	18
6.11 Jumper H - V15 / paralalled bit D5 .....	18
6.12 Jumper G - V16 / paralalled bit D6 .....	19

6.13	Jumper F - V17 / paralalled bit D7 .....	19
6.14	Jumper R - 2716 / 2732 .....	19
6.15	Jumper S - Floppy connector .....	
6.16	Jumper T - Memory Write Signal .....	20
6.17	Jumper layout.....	AA
7.0	Baud Rate Switch .....	21
7.1	Switch Setting .....	21
8.0	External Connector pin definitions .....	22
8.1	J1 - S100 bus .....	22
8.2	J2 - Parallel Port .....	23
8.3	J3 - Floppy Disk controller .....	24
8.4	J4 - Serial port channel A .....	25
8.5	J5 - Serial port channel B .....	25
9.0	Block Diagram .....	26
10.0	Factory Installed Jumpers .....	27
10.1	8 inch drive configuration .....	
10.2	5.25 inch drive configuration .....	28
10.3	Shugart 800 drive .....	29
10.4	shugart 850.....	
10.5	MPI drive .....	
10.6	MFE model 700.....	
10.7	TANDON SLIM LINE .....	30
10.8	NEC model FD1160.....	
10.9	QUME Data Track 8.....	
10.9.1	TANDON 5 1/4".....	31
11.0	Appendix & Data sheets .....	32
11.1	Appendex A - Z80A SIO / DART' .....	
11.2	Appendex B - Z80A PIO .....	
11.3	Appendex C - Z80A CTC .....	
11.4	Appendex D - Floppy Disk controller ....	
11.4.1	WD 1691,BR1941.....	
11.5	Appendex E - Z80A CPU .....	
11.6	Appendix F- FLOPPY ERROR CODE.....	
11.8	Warranty.....	
11.9	Application notes: Turbo-Dos app. notes... Additional memory boards.....	
11.10	Recomended additional S-100 boards: Hard disk: morrow designs,konan,xcomp Memory: measurement systems,PCE Graphics: micro angelo Slave Processor: Advanced Digital, Musys corp.	
12.0	Parts list	
13.0	Schematic Diagram	

## 1.1 The Floppy Disk Controller

-----

The floppy disk controller can access up to four 8 inch or 5.25 inch floppy disk drives. It can read and write IBM 3740 single density format and double density 128,256,512,1024 sector size formats. Data transfer is done by programmed I/O with wait and interrupt synchronization.

Note : The controller cannot access both 8 inch and 5.25 inch drives simultaneously, The controller is switched from 8 inch to 5.25 inch drives by hardware jumper options.

## 1.2 The 64 k Dynamic ram

-----

The 64 k ram array can be switched on and off in 16 k increments (0-16K,16K-32K,32K-48K,48K-64K) under software control. This allows the CPU to access bank switchable external memory on the S100 bus. The memory has an access time of 200ns. Refresh is done during Z80 M1 cycles and during wait and reset states. The memory can be accessed by an external DMA device on the S100 bus.

Note : Any external DMA device that is using continuous mode DMA cycles must transfer data at an average rate of 15 us per byte or faster when holding the DMA request line for more than 1.5 ms This is not a problem because most designers are smart enough to use byte-at-a-time or burst transfer modes when dealing with slow DMA transfer rates. The ram row address is the low order address therefor the entire ram array is refreshed by the DMA device every 128 contiguous memory cycles.

### 1.3 System Monitor Eprom

-----

The system monitor eprom is switched on during reset. It can be disabled and enabled under software control. It resides when enabled at F000h to FFFFh. It has commands that allow the user to load the CP/M , MP/M or other boot loaders from floppy disk. In addition it can be used to load , examine goto and test memory. When the prom is disabled. it does not use any system address space.

### 1.4 Serial ports

-----

A Z80A DART is used for the two serial ports , but a Z80A SIO/0 chip can be used in it's place. This allows asynchronous and synchronous serial data communications plus a variety of interrupt modes. Modem control signals are available at each serial connector. There are two switch selectable baud rate generators for baud rates of 50 to 19.2 k baud.

Note : The serial ports are TTL and must be connected to external interface boards for RS232 communications.

(PS NET/I)

### 1.5 Parallel ports

-----

A Z80A PIO is used as the parallel port. The "A" channel of this chip is connected to the parallel port connector. This port has 8 bi-directional data lines and two hand shake lines. The "B" port can be split between the parallel port connector and the S100 bus vectored interrupt lines by jumper options. This allows the port to be used as an additional parallel port or interrupt controller or both.

In the output mode the parallel ports can drive one TTL load .

## 1.6 Real Time Interrupt clock

---

A Z80A CTC is used for providing a real time system clock for MP/M. Three channels of the CTC are available to the user for strapping via a jumper header for synchronous baud rates or long clock times.

## 1.7 S100 Bus Interface

---

The S100 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 and A16 - A23 are also supported via an I/O port. The PAMNTON line is also implemented for the dynamic ram array.

## 2.0 EPROM and Monitor operation

The onboard EPROM occupies address F000H-FFFFH. The EPROM is switched on automatically during reset or power on, the EPROM contains SIO and 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 so that the ram at address F000H-FFFFH can be accessed. The EPROM can be turned on and off at any time so that hardware dependent I/O routines can be called.

### 2.1 Eprom Enable / Disable

Switching EPROM on :

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

Switching EPROM off :

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

Jumper R configures the board to accept a 2716 or 2732 EPROM.

Note : The EPROM is always addressed at F800H and can not be moved. Since the 2716 EPROM is 2K long it appears twice , F800H-FC00H and FBFFH-FFFFH.

### 2.2 Monitor Signon

The EPROM contains a simple debugger. The monitor signs on with :

```
> ADVANCED          DIGITAL CORP.
  Monitor Version 1.1
  Mar - 1982
  Press "H" for help
>
```

### 2.3 Monitor Commands

The monitor commands are :

Control B = Load disk boot loader 5.25"  
Control C = Load disk boot loader 8 "  
D SSSS QQQQ = Dump memory in hex from SSSS to QQQQ  
F SSSS QQQQ BB = Fill memory from SSSS to QQQQ with BB  
G AAAA = Go to address AAAA  
L AAAA = Load memory starting at AAAA  
M SSSS QQQQ DD = Move from S to Q to ADDR DD  
P SSSS QQQQ = Print in ascii from SSSS to QQQQ  
T = Test Memory  
ESC will terminate any command

The cold boot 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 occurs an error code will be printed.

The error code must be translated using the table

in appendix F page F-4 fig 2.

e.g. FDC COLD BOOT ERROR 80= DISK NOT DETECTED

FDC COLD BOOT ERROR 10= WRONG FORMAT OR NO DATA ON DISK OR BAD PHASE LOC  
LOOP.

## 2.4 Cold Boot Program

```
                ; READ TRACK 0 SECTOR 1 INTO MEMORY  
BOOT 5:  MVI    A,08H  
          OUT    WAIT                ;set double for 5 inch  
BOOT:    MVI    A,0DH                ; RESET FDC  
          OUT    FDC                 ; ISSUE COMMAND  
F4B5 3E0D  
F4B7 D30C  
F4B9 00  
          NOP  
FDCW1:   IN     FDC                 ; CHECK BUSY  
          RRC  
F4BA DB0C  
F4BC 0F  
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

### 3.0 INPUT / OUTPUT PORT ASSIGNMENTS

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	Read/Write	PIO Channel B Data port
06	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/States Port
0D	Read/Write	FDC Track Register
0E	Read/Write	FDC Sector Register
0F	Read/Write	FDC Data Port
10		Unused
11		Unused
12		Unused
13		Unused
14	Read/Write	FDC Synchronization/Drive/Density
15	Write	S100 Buss Extended Address A16-A24
16	Write	On-Board Memory Control Port
17		Unused
18		Unused
19		Unused
1A		Unused
1B		Unused
1C		Unused
1D		Unused
1E		Unused
1F		Unused

All addresses are listed in Hexidecimal.

The unused input / output ports are internally decoded and should not be used by external S100 I/O boards.

#### 4.0 INPUT / OUTPUT PORT DISCRPTIONS

4.1 Serial Communications Port A --- See Appendix A  
00 Read/Write SIO Channel A Data port  
01 Read/Write SIO Channel A Status/Control Port

4.2 Serial Communications Port B --- See Appendix A  
02 Read/Write SIO Channel B Data port  
03 Read/Write SIO Channel B Status/Control Port

4.3 Paralled Interface Port A --- See Appendix B  
04 Read/Write PIO Channel A Data port  
06 Write PIO Channel A Control Port

4.4 Parallel Interface Port B --- See Appendix B

This port can be jumpered via jumpers E through P to the  
S100 Vectored Interrupt lines or to connector J2 (see sec 6.0 )

05 Read/Write PIO Channel B Data port  
07 Write PIO Channel B Control Port

4.5 Control Timmer Interrupt circuit --- See Appendix C  
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

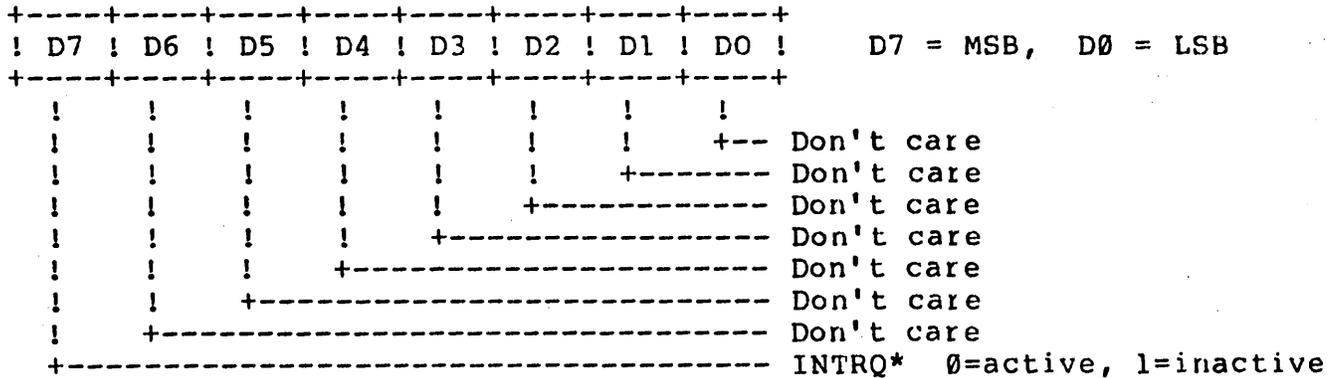
4.6 Floppy Disk Controller --- See Appendix D  
0C Read/Write FDC Command/States Port  
0D Read/Write FDC Track Register  
0E Read/Write FDC Sector Register  
0F Read/Write FDC Data Port

#### 4.7 Floppy Disk Control Port

14            Read/Write            FDC Synchronization/Drive/Density

##### Port Read : CC

When the cpu reads this port the cpu is placed into a wait state until a data byte can be transferred to or from the floppy disk controller or until the command complete/terminate status (INTRQ) is set by the floppy disk controller. The floppy disk controller INTRQ status bit is placed on the data bus as bit D7. This bit can be tested to determine if data is to be transferred or if the command is complete.

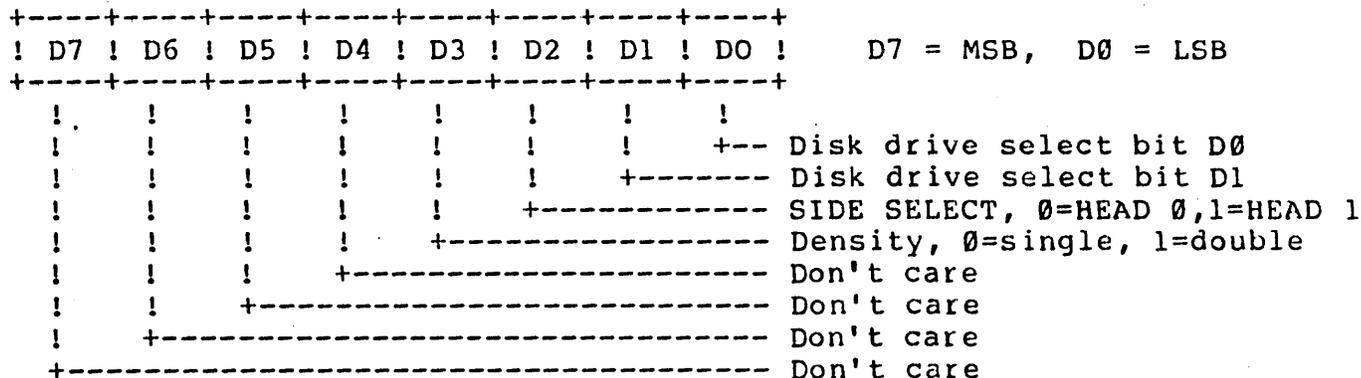


##### Port Write :

The low two bits D0 and D1 of this port control which drive is selected.

D1	D0	
0	0	Disk drive 0 selected
0	1	Disk drive 1 selected
1	0	Disk drive 2 selected
1	1	Disk drive 3 selected

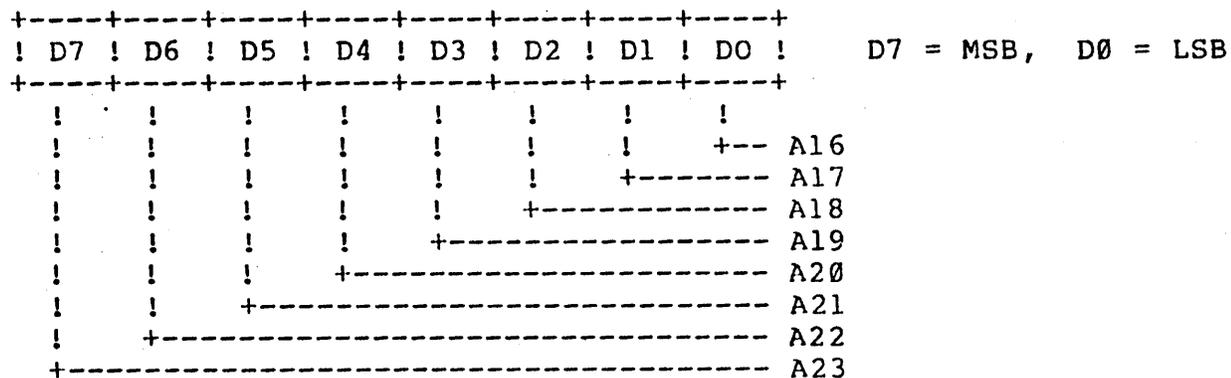
Bit D3 sets the density mode. When bit D3 = 0, single density is selected. When bit D3 = 1, double density is selected.



4.8 Extended address port --- See Section 8.1 (buss defination)  
15 Write S100 Buss Extended Address A16-A23

Port Write :

This port controls the S100 Extended address lines.



#### 4.9 On-Board Memory Control Port

16 Write On-Board Memory Control Port

This port controls the onboard memory management circuit, Prom enable and disable and power on jump reset circuits.

Port write :

The four low order bits D0, D1, D2 and D3 switch the on board memory in 16k banks corresponding to address 0000h-3FFFh, 4000H-7FFFH, 8000H-BFFFH and C000-FFFFH on and off. When a particular bank is switched off, external S100 memory can be accessed in that banks address range. This feature allows external memory to be added to the system for multi-user operating systems.

Bit D5 of this port switches the on-board EPROM on and off.

The onboard EPROM occupies address F000H-FFFFH. The EPROM is switched on automatically during reset or power on, the EPROM contains SIO and 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 so that the ram at address F000H-FFFFH can be accessed.

Bit D6 reset the power on jump circuit. Bit D6 must be set high after a reset or power on situation before ram can be accessed.

! D7	! D6	! D5	! D4	! D3	! D2	! D1	! D0	
								D7 = MSB, D0 = LSB
!	!	!	!	!	!	!	!	1=bank on, 0=bank off
!	!	!	!	!	!	!	!	--- Memory Bank 0000H-3FFFH
!	!	!	!	!	!	!	!	----- Memory Bank 4000H-7FFFH
!	!	!	!	!	!	!	!	----- Memory Bank 8000H-BFFFH
!	!	!	!	!	!	!	!	----- Memory Bank C000H-FFFFH
!	!	!	!	!	!	!	!	----- Don't care
!	!	!	!	!	!	!	!	----- PROM enable=0, Disable=1
!	!	!	!	!	!	!	!	----- Power on jump reset=1
!	!	!	!	!	!	!	!	----- Don't care

5.0

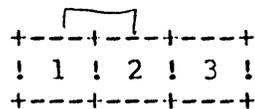
Jumper Definitions

Jumper	Function
A	CPU clock rate 2mhz/4mhz
B	External/Enternal Tx/Rx clock for SIO channel A
C	External/Enternal Tx/Rx clock for SIO channel B
D	Eight inch - five inch Drive selection
E	Select S100 interrupt vector line VI0 OR PINT.
P	Select S100 interrupt vector VIO/PINT or Parallel Port B bit D0 on J2-25.
N	Select S100 interrupt vector V11 or Parallel Port B bit D1 on J2-27.
M	Select S100 interrupt vector V12 or Parallel Port B bit D2 on J2-29.
K	Select S100 interrupt vector V13 or Parallel Port B bit D3 on J2-31.
J	Select S100 interrupt vector V14 or Parallel Port B bit D4 on J2-33.
H	Select S100 interrupt vector V15 or Parallel Port B bit D5 on J2-35.
G	Select S100 interrupt vector V16 or Parallel Port B bit D6 on J2-37.
F	Select S100 interrupt vector V17 or Parallel Port B bit D7 on J2-39.
R	Select 2716 or 2732 EPROM.
S	Define floppy disk connector for eight ad five inch drives.
T	Enable / Disable S100 bus memory write signal on J1 - 68

6.0

Jumper Descriptions

6.1 A CPU clock rate 2mhz/4mhz  
 This jumper determines the cpu clock rate.  
 The jumper is located below IC U7.



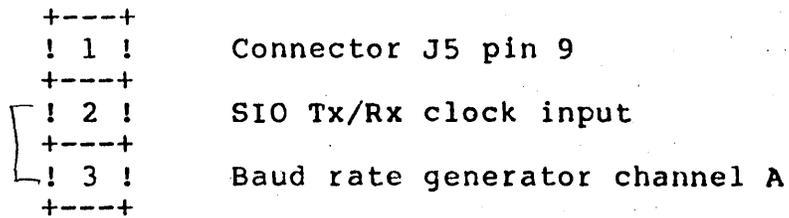
Install Plug between posts 1 & 2 for 4mhz operation.

Install Plug between posts 2 & 3 for 2mhz operation.

6.2 B External/internal Tx/Rx clock for SIO channel A

Jumper B connects the SIO channel A to either the internal baud rate generator or to the connector JA<sup>5</sup> pin 9 for use in synchronous applications.

Jumper B is located near J5.



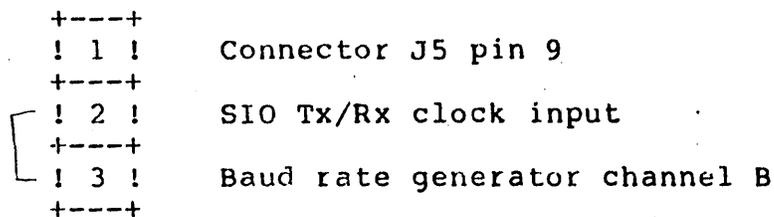
Install Plug between posts 1 & 2 for external SIO clock.

Install Plug between posts 2 & 3 for Baud rate generator.

### 6.3 C External/internal Tx/Rx clock for SIO channel B

Jumper C connects the SIO channel B to either the internal baud rate generator or to connector J5 pin 9 for use in synchronous applications.

Jumper C is located near J5.

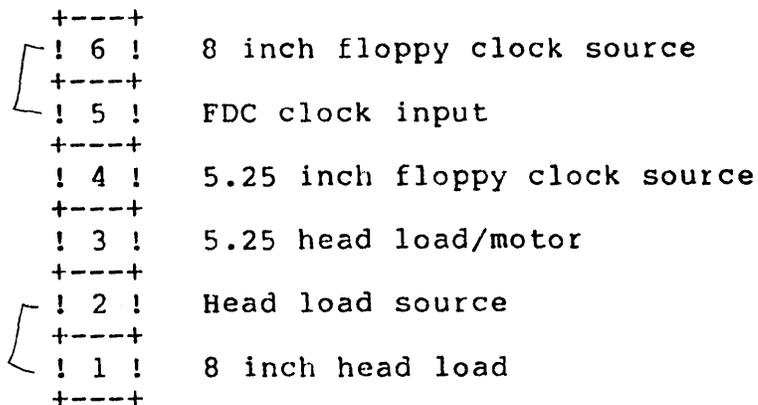


Install Plug between posts 1 & 2 for external SIO clock.

Install Plug between posts 2 & 3 for Baud rate generator.

### 6.4 D Eight inch - five inch Drive selection

Jumper D is located near IC U2.



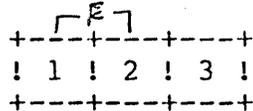
Install Plug between posts 1 & 2 and 5 & 6 for 8 drives.

Install Plug between posts 2 & 3 and 4 & 5 for 5.25 inch drives.

Note : There are other board modifications needed to interface the FDC to a 5.25 inch drive.

6.5 E Select S100 interrupt vector line VI0 OR PINT. Jumper E selects the interrupt line to be used when channel B bit D0 is programmed for interrupts.

Jumper E is located below IC U8.

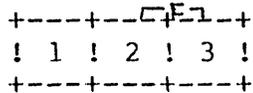


Install Plug between posts 1 & 2 for VI0 interrupt pin. (J1-4)

Install Plug between posts 2 & 3 for PINT interrupt pin. (J1-73)

6.6 F Select S100 interrupt vector VIO/PINT or Parallel Port B bit D0 on J2-25.

This jumper is located near connector J2.

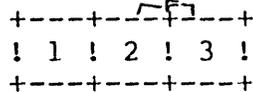


Install Plug between posts 1 & 2 to connect the PIO bit D0 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D0 to the jumper selector area E, VI0/PINT (when the PIO bit is programmed for interrupt mode).

6.7 G Select S100 interrupt vector V11 or Parallel Port B bit D1 on J2-27.

This jumper is located near connector J2.

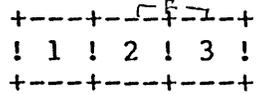


Install Plug between posts 1 & 2 to connect the PIO bit D1 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D1 to the vectored interrupt line V11 (when the PIO bit is programmed for interrupt mode).

6.8 *MH* Select S100 interrupt vector VI2 or Parallel Port B bit D2 on J2-29.

This jumper is located near connector J2.



Install Plug between posts 1 & 2 to connect the PIO bit D2 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D2 to the vectored interrupt line VI2 (when the PIO bit is programmed for interrupt mode).

6.9 *KJ* Select S100 interrupt vector VI3 or Parallel Port B bit D3 on J2-31.

This jumper is located near connector J2.

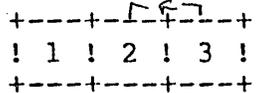


Install Plug between posts 1 & 2 to connect the PIO bit D3 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D3 to the vectored interrupt line VI3 (when the PIO bit is programmed for interrupt mode).

6.10 *JK* Select S100 interrupt vector VI4 or Parallel Port B bit D4 on J2-33.

This jumper is located near connector J2.

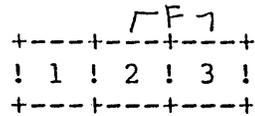


Install Plug between posts 1 & 2 to connect the PIO bit D4 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D4 to the vectored interrupt line VI4 (when the PIO bit is programmed for interrupt mode).

6.11 *AM* Select S100 interrupt vector VI5 or Parallel Port B bit D5 on J2-35.

This jumper is located near connector J2.

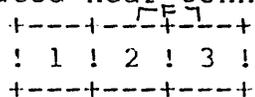


Install Plug between posts 1 & 2 to connect the PIO bit D5 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D5 to the vectored interrupt line VI5 (when the PIO bit is programmed for interrupt mode).

6.12  $\mathcal{N}$  Select S100 interrupt vector VI6 or Parallel Port B bit D6 on J2-37.

This jumper is located near connector J2.

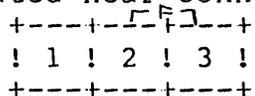


Install Plug between posts 1 & 2 to connect the PIO bit D6 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D6 to the vectored interrupt line VI6 (when the PIO bit is programmed for interrupt mode).

6.13  $\mathcal{P}$  Select S100 interrupt vector VI7 or Parallel Port B bit D7 on J2-39.

This jumper is located near connector J2.



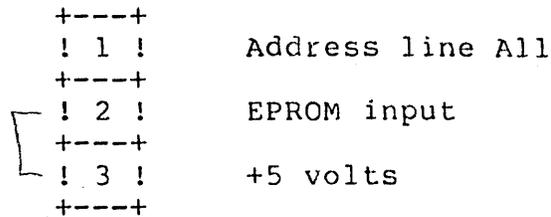
Install Plug between posts 1 & 2 to connect the PIO bit D7 to J2 pin 25 (when the PIO bit is programmed for input/output).

Install Plug between posts 2 & 3 to connect the PIO bit D7 to the vectored interrupt line VI7 (when the PIO bit is programmed for interrupt mode).

6.14 R Select 2716 or 2732 EPROM.

Jumper R configures the board to accept a 2716 or 2732 EPROM.

Jumper R is located near the Z80 chip.



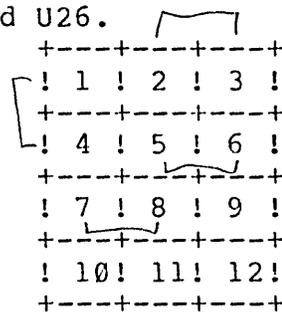
Install Plug between posts 1 & 2 for a 2732 EPROM.

Install Plug between posts 2 & 3 for a 2716 EPROM.

Note : The EPROM is always addressed at F800H and can not be moved. Since the 2716 EPROM is 2K long it appears twice , F800H-FC00H and FBFFH-FFFFH.

6.15 S Define floppy disk connector for eight , five inch drives and FDC chip type.

This jumper is located U26.

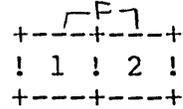


For 8" set-up plug the following jumpers:  
1to4,2to3,5to6,7to8.

For 5 1/4" plug the following jumpers:  
1to2,5to8,6to9,7to10,11to12.(see factory installed jumpers for additional jumpers or changes.)

6.16 T Enable / Disable S100 bus memory write signal on  
 J1 - 68

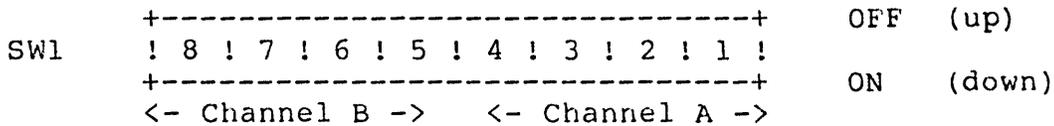
This jumper is located near U18.



Install Plug between posts 1 & 2 to connect the memory write signal (MEMWR) to the S100 bus pin 68.

7.0 Baud Rate Switch  
 -----

The baud rate of the two serial channels can be select separately by setting the baud rate switch. The baud rate switch is an 8 pole switch located near U54. It is split into two sections. Switches 1,2,3,4 set the baud rate for the SIO channel B and switches 5,6,7,8 set the baud rate for the SIO channel A.



7.1 Baud rate switch setting

Switch	8	7	6	5	Channel B baud rate
Switch	4	3	2	1	Channel A baud rate
on	on	on	on	50	
off	on	on	on	75	
on	off	on	on	110	
off	off	on	on	134.5	
on	on	off	on	150	
off	on	off	on	300	
on	off	off	on	600	
off	off	off	on	1200	
on	on	on	off	1800	
off	on	on	off	2000	
on	off	on	off	2400	
off	off	on	off	3600	
on	on	off	off	4800	
off	on	off	off	7200	
on	off	off	off	9600	
off	off	off	off	19,200	

For exact baud rate frequencies see Appendix D

## 8.0 External Connector Pin definitions

---

### 8.1 Connector J1 - S100 bus connector

PIN#	NAME	PIN #	NAME
1	+8V	51	+8V
2	+16V	52	-16V
3	XRDY	53	GND
4	VI0*	54	SLAVE CLR*
5	VI1*	55-57	DMA0*-DMA2*
6	VI2*	58	SXTRQ*
7	VI3*	59	A19
8	VI4*	60	SIXTN*
9	VI5*	61-64	A20-A23
10	VI6*	65,65	NDEF
11	VI7*	67	PHANTOM*
12	NMI*	68	MWRT
13	PWRFAIL*	69	RFU
14	DMA3*	70	GND
15	A18	71	RFU
16	A17	72	RDY
17	A16	73	INT*
18	SDSB*	74	HOLD*
19	CDSB*	75	RESET*
20	GND	76	PSYNCH
21	NDEF	77	PWR*
22	ADSB*	78	PDBIN
23	DODSB*	79-87	A0-A11
24	Ø	88-95	DO2-DI0
25	PSTVAL*	96	SINTA
26	PHLDA	97	SWO*
27,28	RFU	98	ERROR*
29-34	A5,A4,A3,A15,A12,A9		
35	DO1/DATA 1	99	POC*
36	DO0/DATA 0	100	GND
37	A10		
38	DO4		
39	DO5		
40-43	DO6,DI2,DI3,DI7		
44	SMI		
45	SOUT		
46	SINP		
47	SMEMR		
48	SHLTA		
49	CLOCK		
50	GND		

## 8.2 Connector J2 - Parallel port connector

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 D0	
6	PA0 RET	ground	
7	PA1	PIO Channel A data bit D1	
8	PA1 RET	ground	
9	PA2	PIO Channel A data bit D2	
10	PA2 RET	ground	
11	PA3	PIO Channel A data bit D3	
12	PA3 RET	ground	
13	PA4	PIO Channel A data bit D4	
14	PA4 RET	ground	
15	PA5	PIO Channel A data bit D5	
16	PA5 RET	ground	
17	PA6	PIO Channel A data bit D6	
18	PA6 RET	ground	
19	PA7	PIO Channel A data bit D7	
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 D0
	26	PB0 RET	ground
*	27	PB1	PIO Channel B data bit D1
	28	PB1 RET	ground
*	29	PB2	PIO Channel B data bit D2
	30	PB2 RET	ground
*	31	PB3	PIO Channel B data bit D3
	32	PB3 RET	ground
*	33	PB4	PIO Channel B data bit D4
	34	PB4 RET	ground
*	35	PB5	PIO Channel B data bit D5
	36	PB5 RET	ground
*	37	PB6	PIO Channel B data bit D6
	38	PB6 RET	ground
*	39	PB7	PIO Channel B data bit D7
	40	+ 5 VOLTS	

\* Note : These pins can be jumpered to the S100 bus vectored interrupt lines.

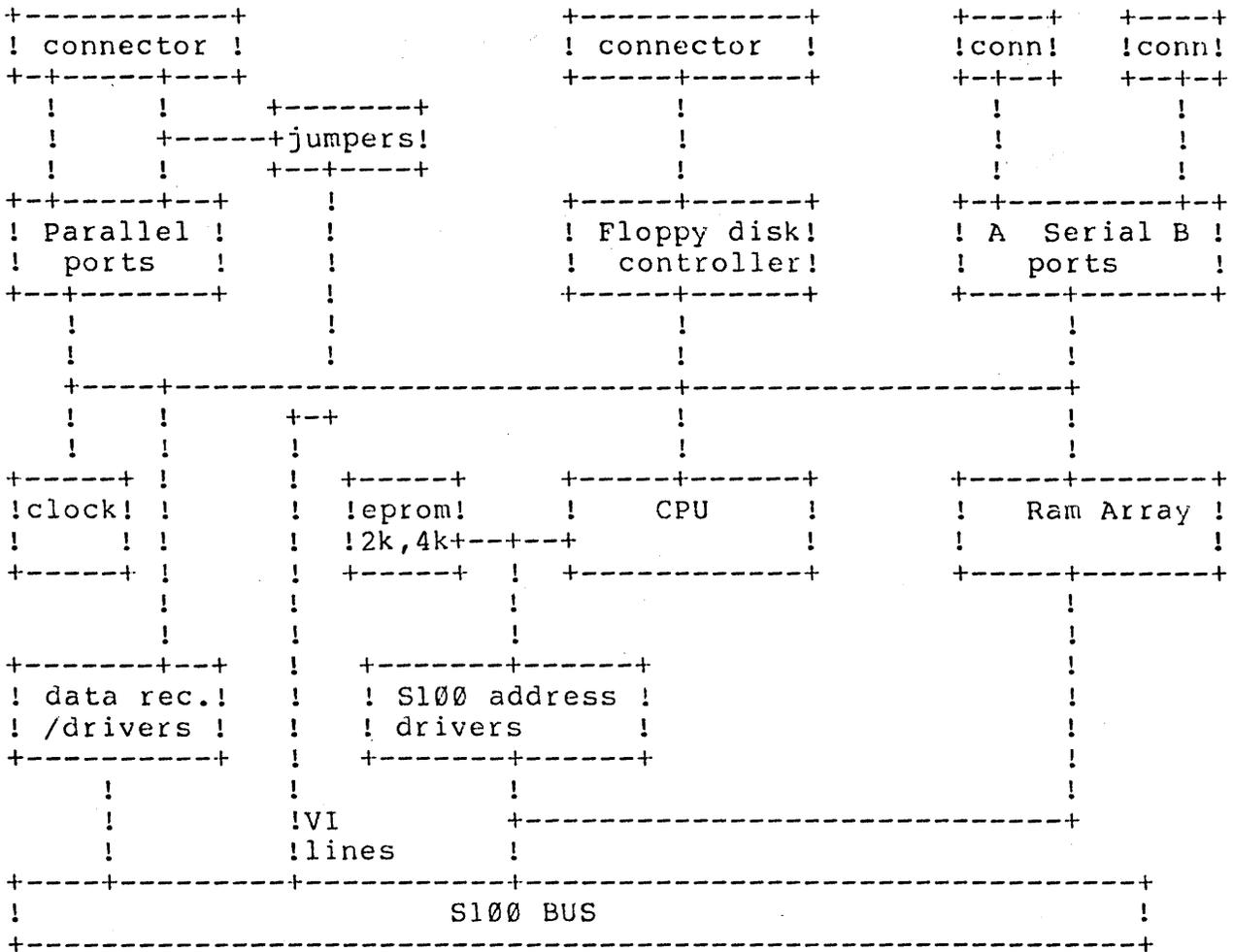
8.3 Connector J3 Floppy disk connector

8 inch 5.25 inch

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	2	Head load*
19	3	ground
20	4	Index*
21	5	ground
22	6	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	16	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	32	Motor on*
49	33	ground
50	34	N/C

4	Connector J4	-	Serial port Channel A
1	N/C		
2	DCDA*		Data Carrer 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*		Transmitt / receive clock
10	GND		
11	N/C		
12	+16 VOLTS		
13	-16 VOLTS		
14	+5 VOLTS		

8.5	Connector J5	-	Serial port Channel B
1	N/C		
2	DCDA*		Data Carrer 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*		Transmitt / receive clock
10	GND		
11	N/C		
12	+16 VOLTS		
13	-16 VOLTS		
14	+5 VOLTS		



10.0

Factory Installed Jumpers  
-----

10.1 Factory Installed jumpers for 8 inch floppy option

Jumper  
-----

A	1-2	CPU clock 4mhz
B	2-3	Tx/Rx clock for SIO A internal
C	2-3	Tx/Rx clock for SIO B internal
D	1-2,5-6	Eight inch Drive selection
E	1-2	Select vector line VI0
F	2-3	Parallel Port B bit D0 on J2-25.
G	2-3	Parallel Port B bit D1 on J2-27.
H	2-3	Parallel Port B bit D2 on J2-29.
J	2-3	Parallel Port B bit D3 on J2-31.
K	2-3	Parallel Port B bit D4 on J2-33.
M	2-3	Parallel Port B bit D5 on J2-35.
N	2-3	Parallel Port B bit D6 on J2-37.
P	2-3	Parallel Port B bit D7 on J2-39.
R	2-3	Select 2716
S	1,4;2,3;5,6;7,8	define floppy disk connector
T	1-2	Enable S100 bus memory write signal

## 10.2 Factory Installed jumpers for 5.25 inch floppy option

### Jumper

-----

A	2-3	CPU clock 4mhz
B	2-3	Tx/Rx clock for SIO A internal
C	2-3	Tx/Rx clock for SIO B internal
D	1-2,5-6	Eight inch Drive selection
E	1-2	Select vector line VI0
F	2-3	Parallel Port B bit D0 on J2-25.
G	2-3	Parallel Port B bit D1 on J2-27.
H	2-3	Parallel Port B bit D2 on J2-29.
J	2-3	Parallel Port B bit D3 on J2-31.
K	2-3	Parallel Port B bit D4 on J2-33.
M	2-3	Parallel Port B bit D5 on J2-35.
N	2-3	Parallel Port B bit D6 on J2-37.
P	2-3	Parallel Port B bit D7 on J2-39.
R	2-1	Select 2716
S	1,2;5,8;6,9;7,10;11,12	Define floppy disk connector
T	1-2	Enable S100 bus memory write signal

(to convert an 8" board to 5 1/4" the following is required in addition to the addendum)

Capacitor C22 is changed to a 100pf or parallel a 47pf with it.

Capacitor C6 is changed to a 10mf.

Resistor R4 is changed to 220k ohm

Re-adjust the phase lock loop by adjusting the R26 pot.

An easy way to adjust would be to put one trace of the scope on pin 7 of the u27(74s124) and the other channel on a 2 MHZ.

10.3 Shugart SA 800 Jumpers  
-----

Disk drive jumpers

Remove all jumpers on the disk drive. Install jumpers as follows:

Jumper Y  
Jumper C  
Jumper T2  
Jumper T1  
Jumper 800  
Jumper L  
Jumper A  
Jumper B  
Jumper DS

10.4 Shugart 850  
-----

Disk drive Jumpers

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  
DS1 for drive 1 and DS2 for drive 2  
install termination at the end of cable.

10.6 MFE Model 700

Disk drive jumpers

Install jumpers as follows:

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

10.7 TANDON SLIM LINE

Disk drive jumpers

Install jumpers as follows:

DS1 OR DS2  
INSTALL THE TERMINATION RES. AT THE END OF THE CABLE

10.8 NEC model FD1160

Disk drive Jumpers

Install jumpers as follows:

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

10.9 QUME data track 8

Disk drive jumpers

Install jumpers as follows:

DS1 OR DS2  
CUT X  
CUT Z  
CUT L  
INSTALL Y  
INSTALL C

10.9.1 TANDON 5 1/4 " DRIVE(48TP1)

Disk drive jumpers

Install jumpers as follows:

On the dip shunt header only pin2,7,8 are on.  
for DS1 and pins3,7,8 for DS2.

11.0 Appendix & Data sheets  
-----

11.1 Appendix A - Z80A SIO / DART .....

# Z8470 Z80<sup>®</sup> DART Dual Asynchronous Receiver/Transmitter



## 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 micro-computer 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.

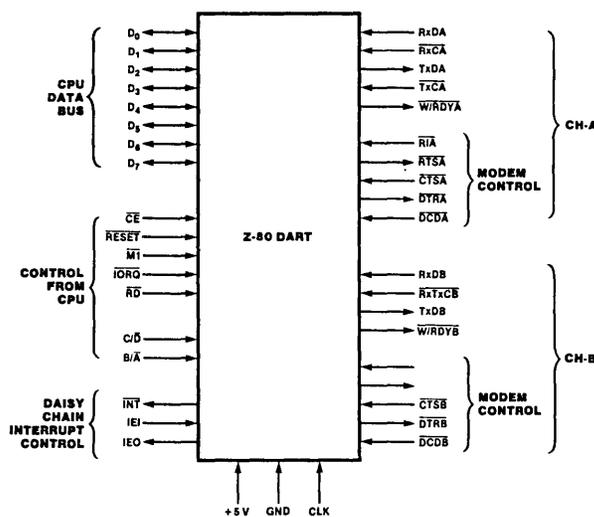


Figure 1. Z80 DART Pin Functions

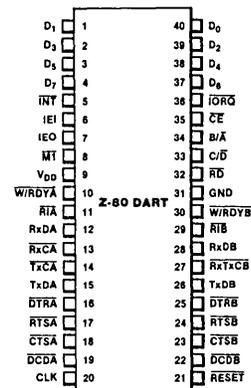


Figure 2. Pin Assignments

Pin	Description
<b>B/<math>\bar{A}</math>.</b>	<b>Channel A Or B Select</b> (input, High selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the Z-80 DART.
<b>C/<math>\bar{D}</math>.</b>	<b>Control Or Data Select</b> (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.
<b><math>\overline{CE}</math>.</b>	<b>Chip Enable</b> (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.
<b>CLK.</b>	<b>System Clock</b> (input). The Z-80 DART uses the standard Z-80 single-phase system clock to synchronize internal signals.
<b><math>\overline{CTSA}</math>, <math>\overline{CTSB}</math>.</b>	<b>Clear To Send</b> (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.
<b>D<sub>0</sub>-D<sub>7</sub>.</b>	<b>System Data Bus</b> (bidirectional, 3-state) transfers data and commands between the CPU and the Z-80 DART.
<b>DCDA, DCDB.</b>	<b>Data Carrier Detect</b> (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.
<b>DTRA, DTRB.</b>	<b>Data Terminal Ready</b> (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be programmed as general-purpose outputs.
<b>IEI.</b>	<b>Interrupt Enable In</b> (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.
<b>IEO.</b>	<b>Interrupt Enable Out</b> (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.
<b>INT.</b>	<b>Interrupt Request</b> (output, open drain, active Low). When the Z-80 DART is requesting an interrupt, it pulls INT Low.
<b>M1.</b>	<b>Machine Cycle One</b> (input from Z-80 CPU, active Low). When M1 and RD are both active, the Z-80 CPU is fetching an instruction from memory; when M1 is active while $\overline{IORQ}$ is active, the Z-80 DART accepts 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.
<b><math>\overline{IORQ}</math>.</b>	<b>Input/Output Request</b> (input from CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/ $\bar{A}$ , C/ $\bar{D}$ , $\overline{CE}$ and RD to transfer commands and data between the CPU and the Z-80 DART. When $\overline{CE}$ , RD and $\overline{IORQ}$ are all active, the channel selected by B/ $\bar{A}$ transfers data to the CPU (a read operation). When $\overline{CE}$ and $\overline{IORQ}$ are active, but RD is inactive, the channel selected by B/ $\bar{A}$ is written to by the CPU with either data or control information as specified by C/ $\bar{D}$ .
<b>RxCA, RxCB.</b>	<b>Receiver Clocks</b> (inputs). Receive data is sampled on the rising edge of RxC. The Receive Clocks may be 1, 16, 32 or 64 times the data rate.
<b>RD.</b>	<b>Read Cycle Status</b> (input from CPU, active Low). If RD is active, a memory or I/O read operation is in progress.
<b>RxDA, RxDB.</b>	<b>Receive Data</b> (inputs, active High).
<b>RESET.</b>	<b>Reset</b> (input, active Low). Disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High and disables all interrupts.
<b>RIA, RIB.</b>	<b>Ring Indicator</b> (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.
<b>RTSA, RTSB.</b>	<b>Request to Send</b> (outputs, active Low). When the RTS bit is set, the RTS output goes Low. When the RTS bit is reset, the output goes High after the transmitter empties.
<b>TxCA, TxCB.</b>	<b>Transmitter Clocks</b> (inputs). TxD changes on the falling edge of 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.
<b>TxDA, TxDB.</b>	<b>Transmit Data</b> (outputs, active High).
<b>W/RDYA, W/RDYB.</b>	<b>Wait/Ready</b> (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.

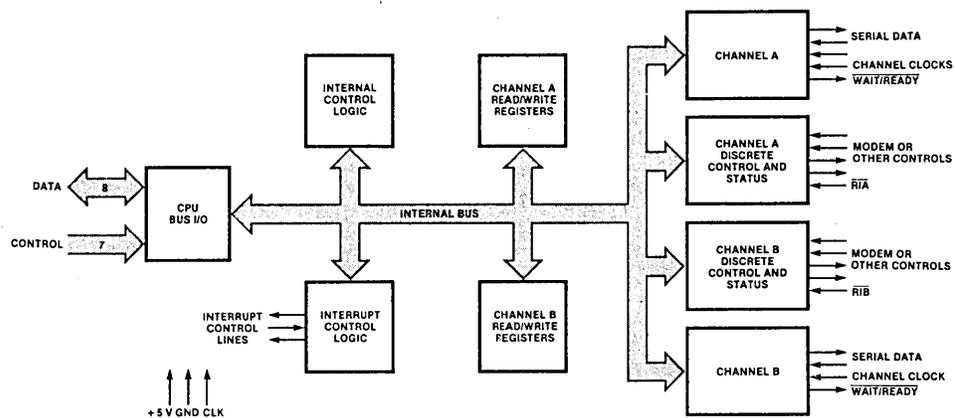


Figure 3. Block Diagram

---

**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 acknowledgment 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  $\overline{WRDY}$  output in conjunction with the Wait/Ready bits of Write Register 1. The  $\overline{WRDY}$  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 (CTS), Data Carrier Detect (DCD) and Ring Indicator (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.

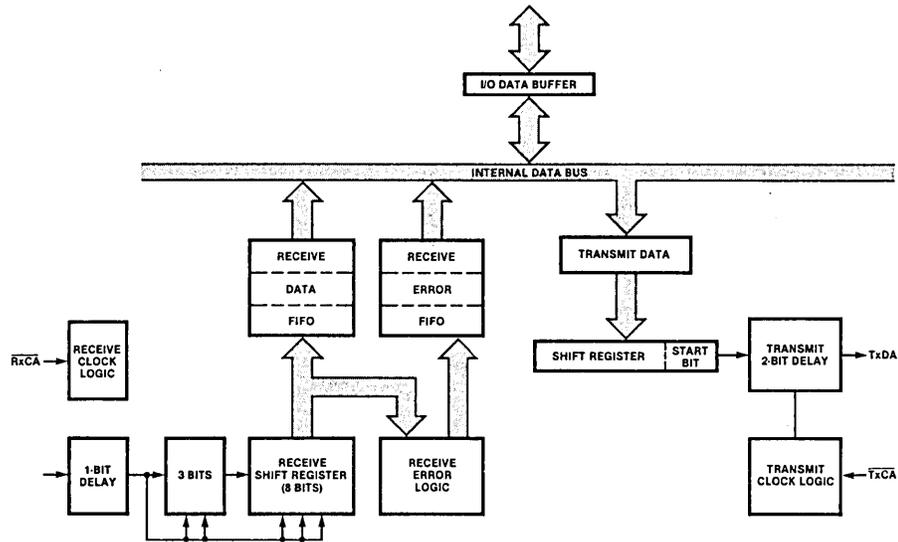


Figure 4. Data Path

**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 out-

put instruction to write a Data or Control byte into the Z-80 DART.

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

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

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

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

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

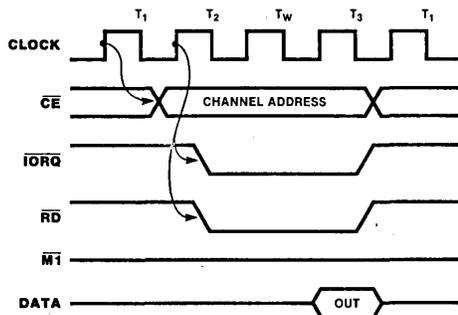


Figure 5a. Read Cycle

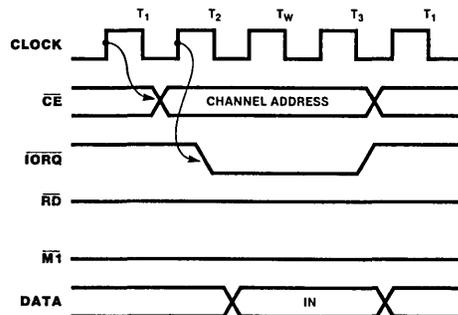


Figure 5b. Write Cycle

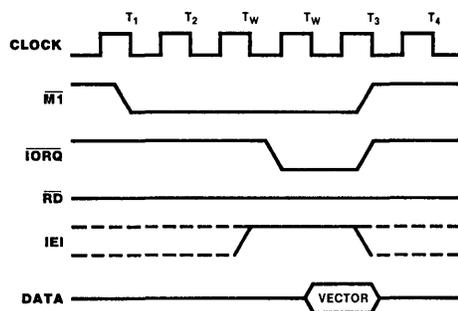


Figure 5c. Interrupt Acknowledge Cycle

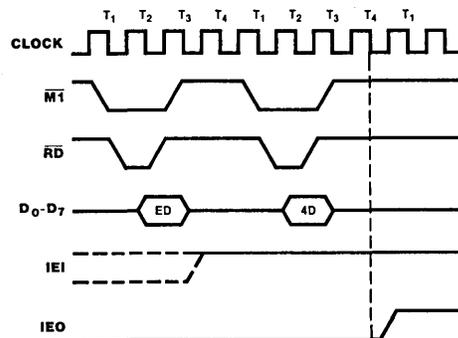


Figure 5d. Return from Interrupt Cycle

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

Both channels contain command registers that must be programmed via the system program prior to operation. The Channel Select input ( $B/\bar{A}$ ) and the Control/Data input ( $C/\bar{D}$ ) 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).

---

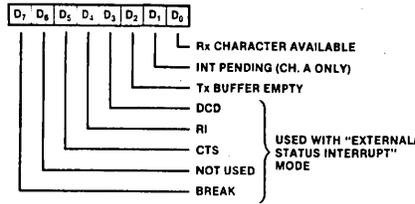
**Read Register Functions**

---

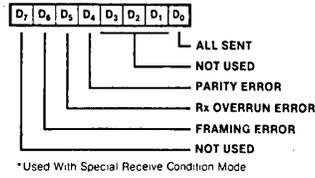
- |     |  |
|-----|--|
| RR0 | Transmit/Receive buffer status, interrupt status and external status |
| RR1 | Special Receive Condition status                                     |
| RR2 | Modified interrupt vector (Channel B only)                           |

**Z-80 DART**  
Read and Write  
Registers

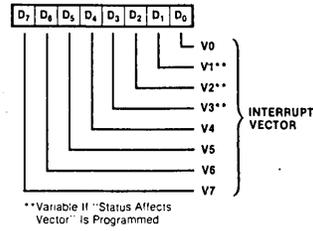
**READ REGISTER 0**



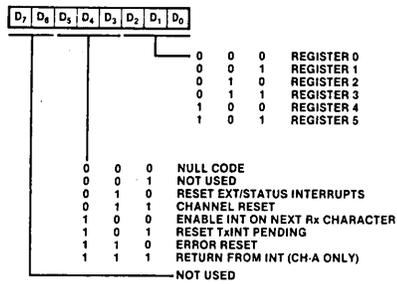
**READ REGISTER 1\***



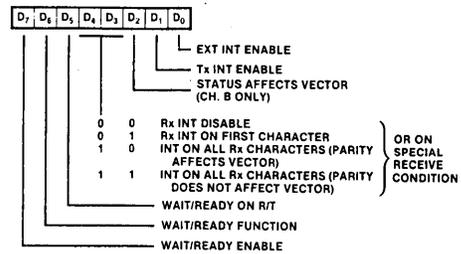
**READ REGISTER 2**



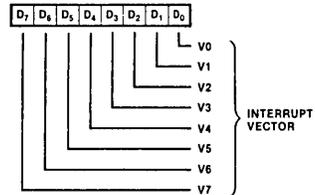
**WRITE REGISTER 0**



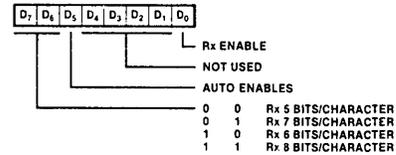
**WRITE REGISTER 1**



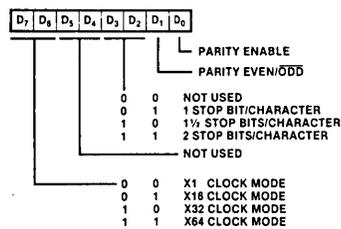
**WRITE REGISTER 2 (CHANNEL B ONLY)**



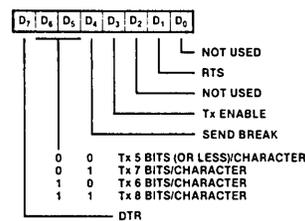
**WRITE REGISTER 3**



**WRITE REGISTER 4**



**WRITE REGISTER 5**



11.2 Appendix B - Z80A PIO .....

# Z8420 Z80<sup>®</sup> PIO Parallel Input/Output Controller



## 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.

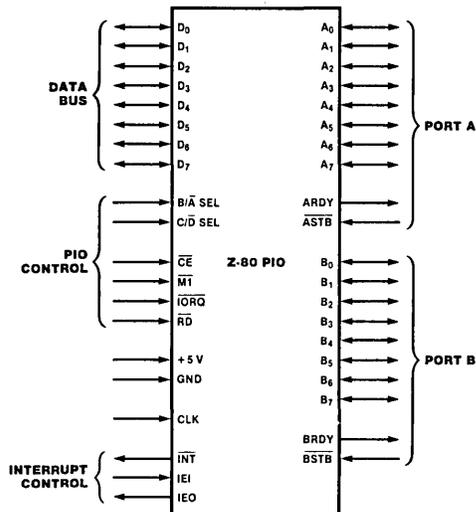


Figure 1. Pin Functions

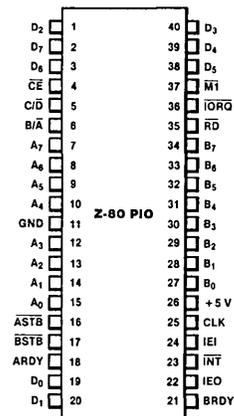


Figure 2. Pin Assignments



**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.

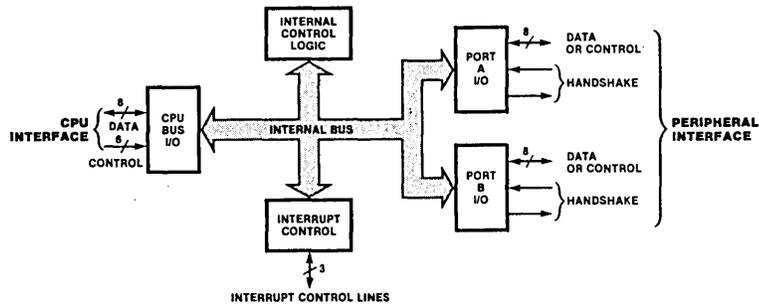


Figure 4. Block Diagram

**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.

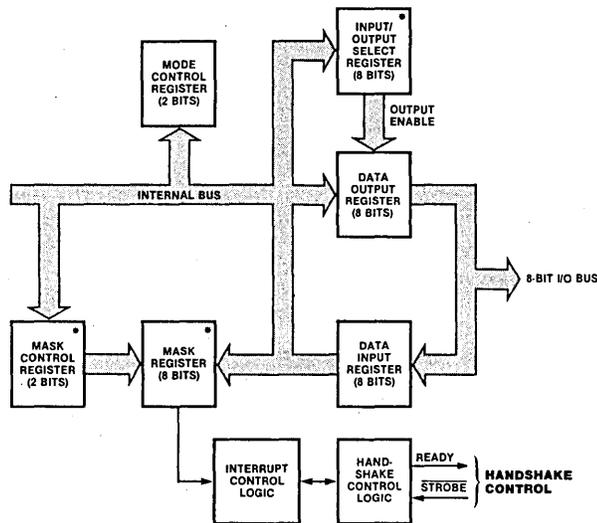
Unlike the other Z-80 peripherals, the PIO does not enable interrupts immediately after programming. It waits until M1 goes Low (e.g., during an opcode fetch). This condition is unimportant in the Z-80 environment but might not be if another type of CPU is used.

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  $\overline{RD}$ ,  $\overline{CE}$ ,  $\overline{C/D}$  and  $\overline{IORQ}$  signals generate the write input internally.



\*Used in the bit mode only to allow generation of an interrupt if the peripheral I/O pins go to the specified state.

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<sub>6</sub> 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<sub>5</sub>.

**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<sub>4</sub> must be set. When D<sub>4</sub> 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).

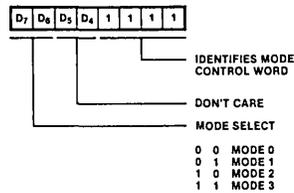
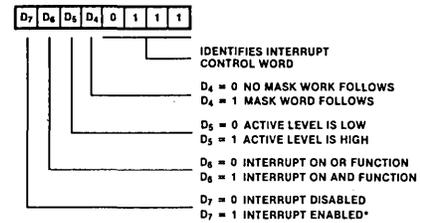


Figure 6. Mode Control Word



\*NOTE: THE PORT IS NOT ENABLED UNTIL THE INTERRUPT ENABLE IS FOLLOWED BY AN ACTIVE INT.

Figure 9. Interrupt Control Word

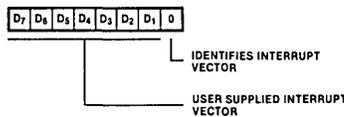


Figure 7. Interrupt Vector Word

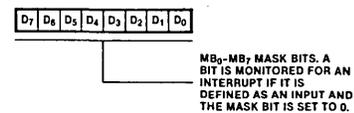


Figure 10. Mask Control Word

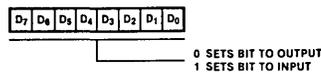


Figure 8. I/O Register Control Word

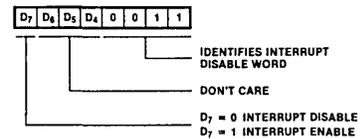


Figure 11. Interrupt Disable Word

**Pin  
Description**

**A<sub>0</sub>-A<sub>7</sub>. 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<sub>0</sub> 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<sub>0</sub>-B<sub>7</sub>. 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<sub>0</sub> 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<sub>0</sub> 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<sub>1</sub> 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<sub>0</sub>-D<sub>7</sub>. 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<sub>0</sub> 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  $\overline{MI}$  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)

**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.

**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.

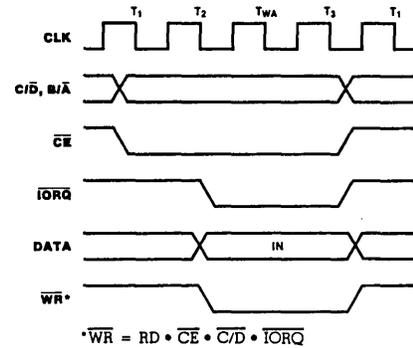


Figure 12. Write Cycle Timing

**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.

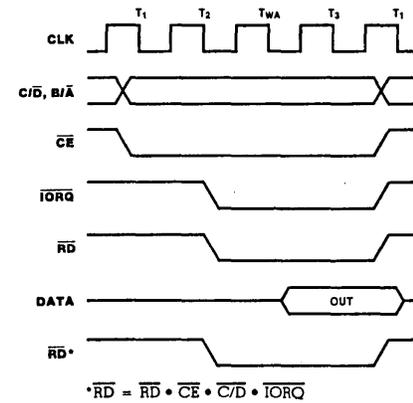


Figure 13. Read Cycle Timing

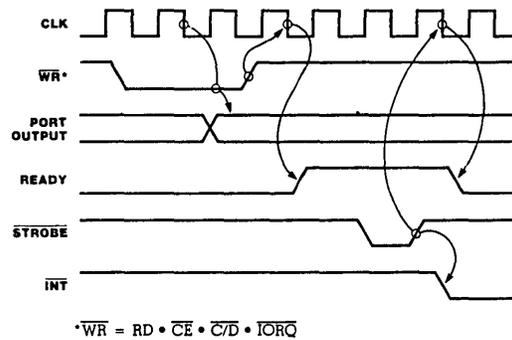


Figure 14. Mode 0 Output Timing

**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.

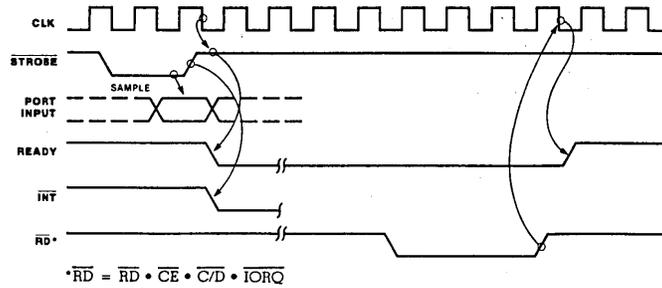


Figure 15. Mode 1 Input Timing

**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.

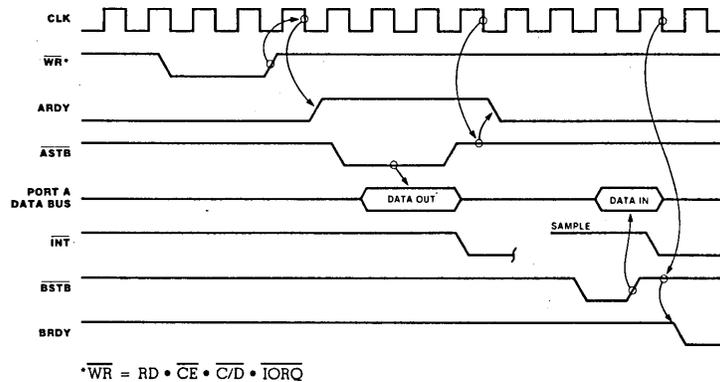


Figure 16. Mode 2 Bidirectional Timing

**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.

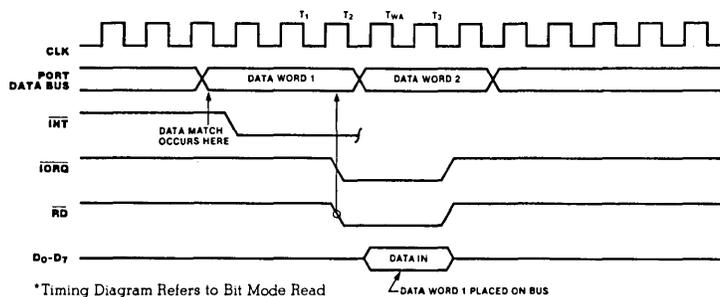


Figure 17. Mode 3 Bit Mode Timing

**Interrupt Acknowledge Timing.** During  $\overline{MI}$  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 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.

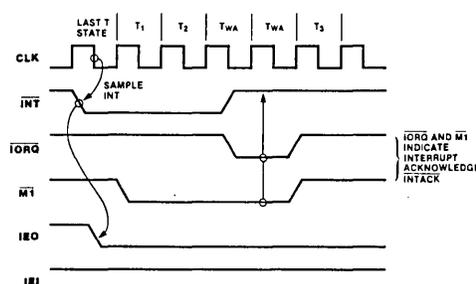


Figure 18. Interrupt Acknowledge Timing

**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.

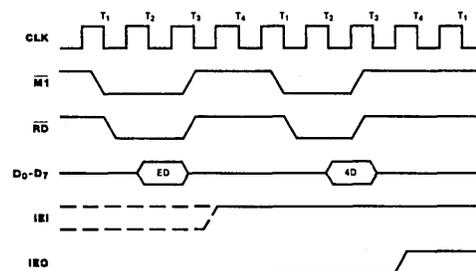
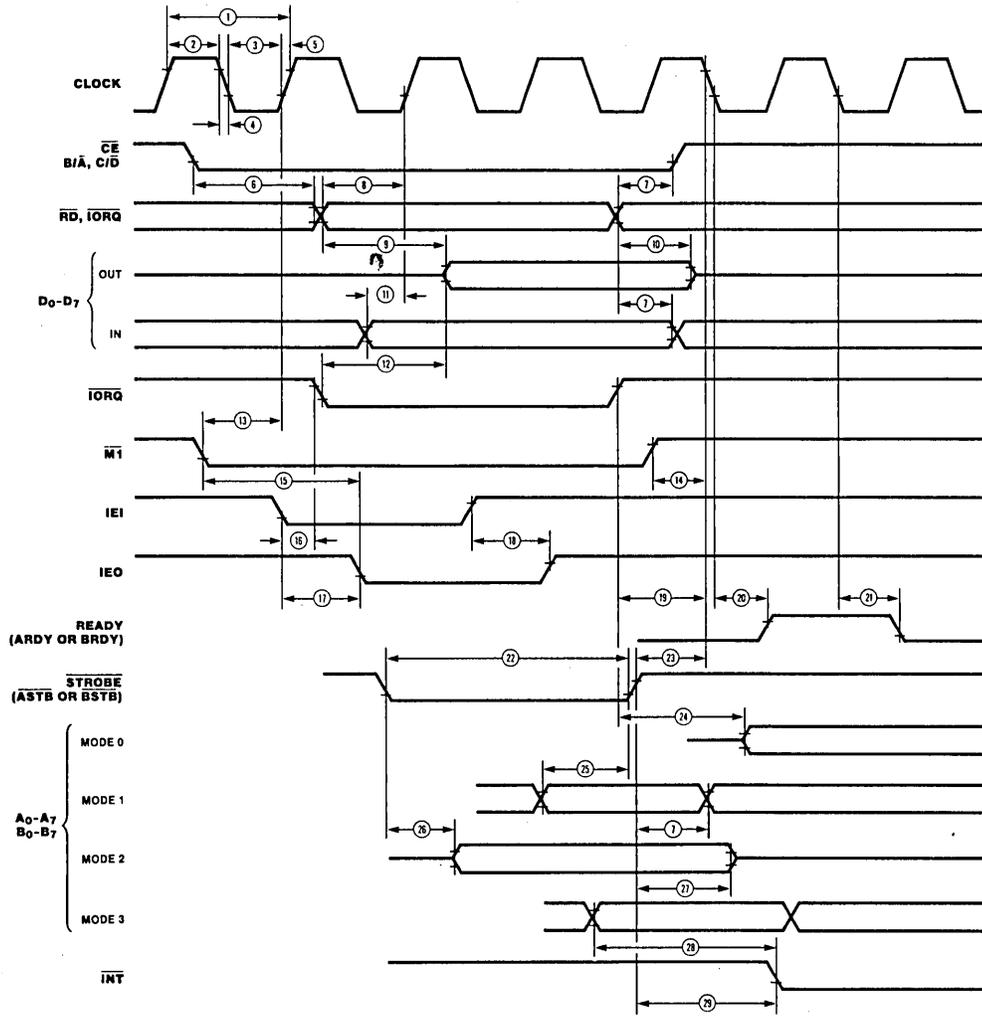


Figure 19. Return From Interrupt

**AC  
Charac-  
teristics**



Number	Symbol	Parameter	Z-80 PIO		Z-80A PIO		Z-80B PIO <sup>[9]</sup>		Comment
			Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	
1	T <sub>c</sub> C	Clock Cycle Time	400	[1]	250	[1]	165	[1]	
2	T <sub>w</sub> Ch	Clock Width (High)	170	2000	105	2000	65	2000	
3	T <sub>w</sub> Cl	Clock Width (Low)	170	2000	105	2000	65	2000	
4	T <sub>i</sub> C	Clock Fall Time		30		30		20	
5	T <sub>r</sub> C	Clock Rise Time		30		30		20	
6	T <sub>s</sub> CS(RI)	$\overline{CE}$ , B/ $\overline{A}$ , C/ $\overline{D}$ to $\overline{RD}$ , $\overline{IORQ}$ ↓ Setup Time	50		50		50		[6]
7	T <sub>h</sub>	Any Hold Times for Specified Setup Time	0		0		0	0	
8	T <sub>s</sub> RI(C)	$\overline{RD}$ , $\overline{IORQ}$ to Clock ↑ Setup Time	115		115		70		
9	T <sub>d</sub> RI(DO)	$\overline{RD}$ , $\overline{IORQ}$ ↓ to Data Out Delay		430		380		300	[2]
10	T <sub>d</sub> RI(DOs)	$\overline{RD}$ , $\overline{IORQ}$ ↑ to Data Out Float Delay		160		110		70	
11	T <sub>s</sub> DI(C)	Data In to Clock ↑ Setup Time	50		50		40		CL = 50 pF
12	T <sub>d</sub> IO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)	340		160		120		[3]
13	T <sub>s</sub> M1(Cr)	$\overline{M1}$ ↓ to Clock ↑ Setup Time	210		90		70		
14	T <sub>s</sub> M1(Cf)	$\overline{M1}$ ↑ to Clock ↓ Setup Time ( $\overline{M1}$ Cycle)	0		0		0		[8]
15	T <sub>d</sub> M1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt Immediately Preceding $\overline{M1}$ ↓)		300		190		100	[5, 7]
16	T <sub>s</sub> IEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		140		100		[7]
17	T <sub>d</sub> IEI(IEOf)	IEI ↓ to IEO ↓ Delay		190		130		120	[5] CL = 50 pF
18	T <sub>d</sub> IEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED Decode)		210		160		160	[5]
19	T <sub>c</sub> IO(C)	$\overline{IORQ}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)	220		200		170		
20	T <sub>d</sub> C(RDYr)	Clock ↓ to READY ↑ Delay	200		190		170		[5] CL = 50 pF
21	T <sub>d</sub> C(RDYf)	Clock ↓ to READY ↑ Delay	150		140		120		[5]
22	T <sub>w</sub> STB	$\overline{STROBE}$ Pulse Width	150		150		120		[4]
23	T <sub>s</sub> STB(C)	$\overline{STROBE}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)	220		220		150		[5]
24	T <sub>d</sub> IO(PD)	$\overline{IORQ}$ ↑ to PORT DATA Stable Delay (Mode 0)		200		180		160	[5]
25	T <sub>s</sub> PD(STB)	PORT DATA to $\overline{STROBE}$ ↑ Setup Time (Mode 1)	260		230		190		
26	T <sub>d</sub> STB(PD)	$\overline{STROBE}$ ↓ to PORT DATA Stable (Mode 2)		230		210		180	[5]
27	T <sub>d</sub> STB(PDr)	$\overline{STROBE}$ ↑ to PORT DATA Float Delay (Mode 2)		200		180		160	CL = 50 pF
28	T <sub>d</sub> PD(INT)	PORT DATA Match to $\overline{INT}$ ↓ Delay (Mode 3)		540		490		430	
29	T <sub>d</sub> STB(INT)	$\overline{STROBE}$ ↑ to $\overline{INT}$ ↓ Delay		490		440		350	

NOTES:

- [1] T<sub>c</sub>C = T<sub>w</sub>Ch + T<sub>w</sub>Cl + T<sub>r</sub>C + T<sub>i</sub>C.
- [2] Increase T<sub>d</sub>RI(DO) by 10 ns for each 50 pF increase in load up to 200 pF max.
- [3] Increase T<sub>d</sub>IO(DOI) by 10 ns for each 50 pF, increase in loading up to 200 pF max.
- [4] For Mode 2: T<sub>w</sub>STB > T<sub>s</sub>PD(STB).
- [5] Increase these values by 2 ns for each 10 pF increase in loading up to 100 pF max.

- [6] T<sub>s</sub>CS(RI) may be reduced. However, the time subtracted from T<sub>s</sub>CS(RI) will be added to T<sub>d</sub>RI(DO).
- [7] 2.5 T<sub>c</sub>C > (N-2)T<sub>d</sub>IEI(IEOf) + T<sub>d</sub>M1(IEO) + T<sub>s</sub>IEI(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.

Z80 PIO

**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.

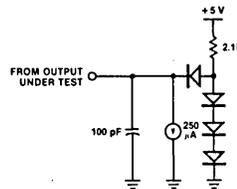
**Test Conditions**  
 The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0 V). Positive current flows into the referenced pin. Available operating temperature ranges are:

- 0° to +70°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.25 V
- -40°C to +85°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +5.25 V
- -55° to +125°C,  
+4.75 V ≤ V<sub>CC</sub> ≤ +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 <sub>ILC</sub>	Clock Input Low Voltage	-0.3	+0.45	V	
	V <sub>IHC</sub>	Clock Input High Voltage	V <sub>CC</sub> -0.6	+5.5	V	
	V <sub>IL</sub>	Input Low Voltage	-0.3	+0.8	V	
	V <sub>IH</sub>	Input High Voltage	+2.0	+5.5	V	
	V <sub>OL</sub>	Output Low Voltage		+0.4	V	I <sub>OL</sub> = 2.0 mA
	V <sub>OH</sub>	Output High Voltage	+2.4		V	I <sub>OH</sub> = -250 μA
	I <sub>LI</sub>	Input Leakage Current	-10.0	+10.0	μA	0 < V <sub>IN</sub> < V <sub>CC</sub>
	I <sub>Z</sub>	3-State Output/Data Bus Input Leakage Current	-10.0	+10.0	μA	0 < V <sub>IN</sub> < V <sub>CC</sub>
	I <sub>CC</sub>	Power Supply Current		100.0	mA	V <sub>OH</sub> = 1.5V
	I <sub>OHD</sub>	Darlington Drive Current	-1.5	3.8	mA	R <sub>EXT</sub> = 390 Ω

Over specified temperature and voltage range.

Capacitance	Symbol	Parameter	Min	Max	Unit	Test Condition
	C	Clock Capacitance		10	pF	Unmeasured
	C <sub>IN</sub>	Input Capacitance		5	pF	pins returned to ground
	C <sub>OUT</sub>	Output Capacitance		10	pF	

Over specified temperature range; f = 1MHz

11.3 Appendix C - Z80A CTC .....

# Z8430 Z80<sup>®</sup> CTC Counter/ Timer Circuit



## 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.

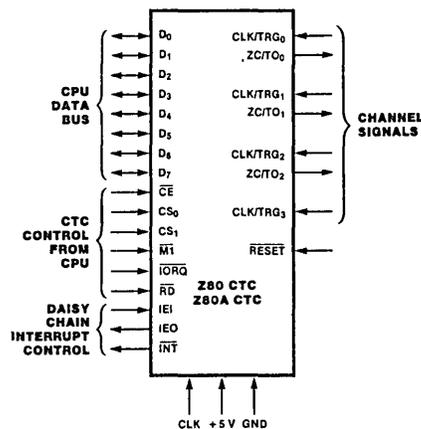


Figure 1. Pin Functions

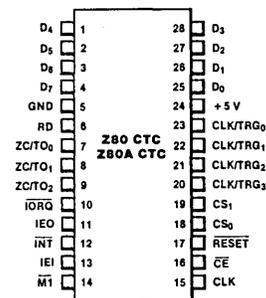


Figure 2. Pin Assignments

**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  $\mu$ s (Z-80A) or 6.4  $\mu$ s (Z-80) without additional logic or software timing loops. Time intervals are generated by dividing the system clock with a prescaler that decrements

a preset down-counter.

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

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

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

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

**Architecture**

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

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

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

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

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

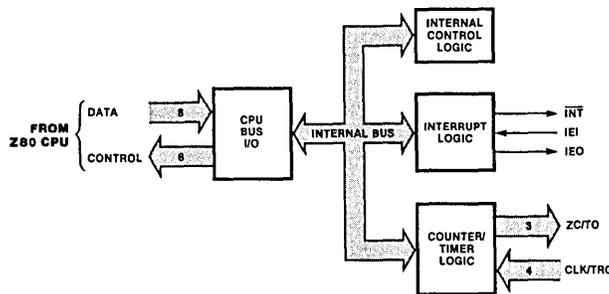


Figure 3. Functional Block Diagram

**Architecture**  
(Continued)

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

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

If an interrupt 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 ( $\text{ED}_{16}$ ). If the device has a pending interrupt, it raises IEO (High) for one  $\overline{\text{M1}}$  cycle. This ensures that all lower priority devices can decode the entire RETI instruction and reset properly.

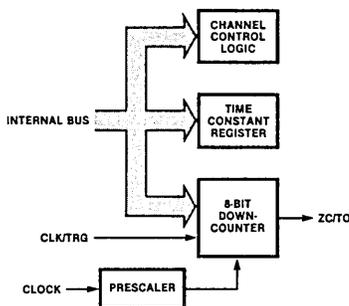


Figure 4. Counter/Timer Block Diagram

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

**Channel Control Logic.** The channel control logic receives the 8-bit channel control word when the counter/timer channel is programmed. The channel control logic decodes

the control word and sets the following operating conditions:

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

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

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

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

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

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

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

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

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

A control word is identified by a 1 in bit 0. A 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<sub>1</sub> and CS<sub>2</sub>. A 2-bit binary code selects the appropriate channel as shown in the following table.

Channel	CS <sub>1</sub>	CS <sub>0</sub>
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 IEL, and

D<sub>0</sub>-D<sub>7</sub> 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<sub>1</sub> and D<sub>2</sub> 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<sub>3</sub> = 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<sub>7</sub> enables the interrupt, so that an interrupt output (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<sub>6</sub> selects either timer or counter mode.

**Prescaler Factor. (Timer Mode Only).** D<sub>5</sub> selects factor—either 16 or 256.

**Trigger Slope.** D<sub>4</sub> 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.

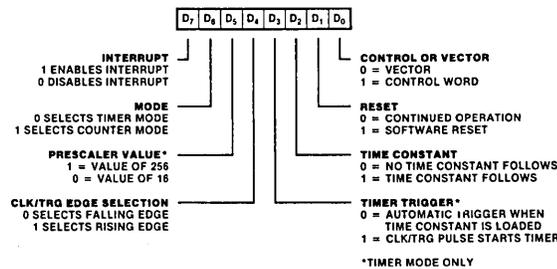


Figure 5. Channel Control Word

**Programming Trigger Mode (Timer Mode Only)** (Continued)

$D_3$  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.

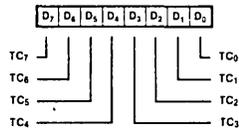


Figure 6. Time Constant Word

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

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

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

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

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

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

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

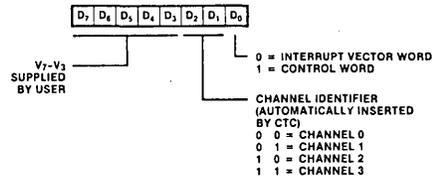


Figure 7. Interrupt Vector Word

**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<sub>0</sub>-CLK/TRG<sub>3</sub>.** *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<sub>0</sub>-CS<sub>1</sub>.** *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<sub>0</sub> and A<sub>1</sub>).

**D<sub>0</sub>-D<sub>7</sub>.** *System Data Bus* (bidirectional, 3-state). Transfers all data and commands between the Z-80 CPU and the Z-80 CTC.

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

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

**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  $\overline{MI}$  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.

**$\overline{MI}$ .** *Machine Cycle One* (input from CPU, active Low). When  $\overline{MI}$  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.

**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<sub>0</sub>-D<sub>7</sub> go to the high-impedance state.

**ZC/TO<sub>0</sub>-ZC/TO<sub>2</sub>.** *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.

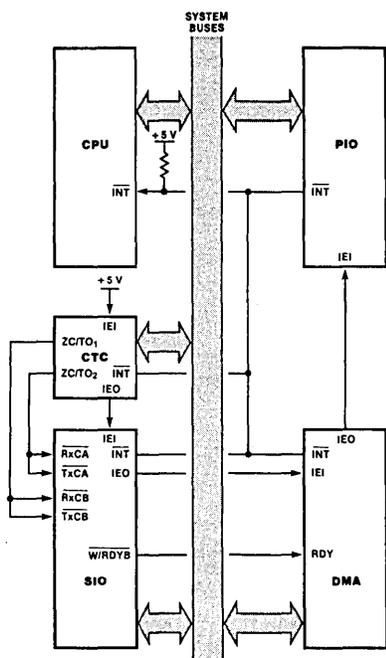


Figure 8. A Typical Z-80 Environment

**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.

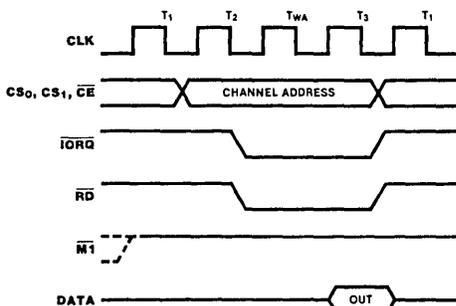


Figure 9. Read Cycle Timing

**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

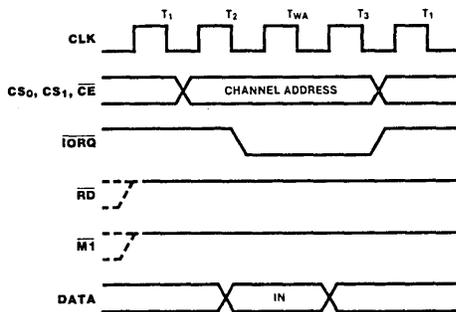


Figure 10. Write Cycle Timing

latched into the appropriate register with the rising edge of clock cycle  $T_{WA}$ . No additional wait states are allowed.

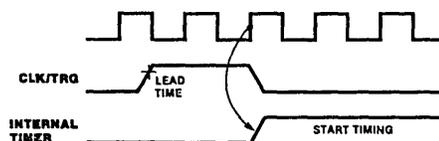


Figure 11. Timer Mode Timing

**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.

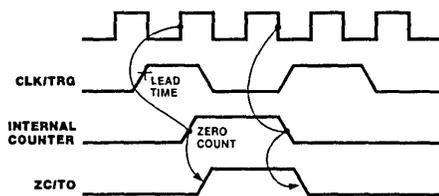


Figure 12. Counter Mode Timing

**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.

## 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*.

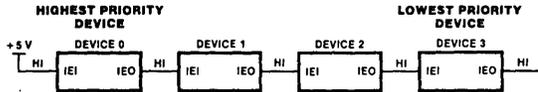


Figure 13. Daisy-Chain Interrupt Priorities

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 ( $\overline{MI}$  and  $\overline{IORQ}$ ). All channels are inhibited from changing their interrupt request status when  $\overline{MI}$  is active—about two clock cycles earlier than  $\overline{IORQ}$ . 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.

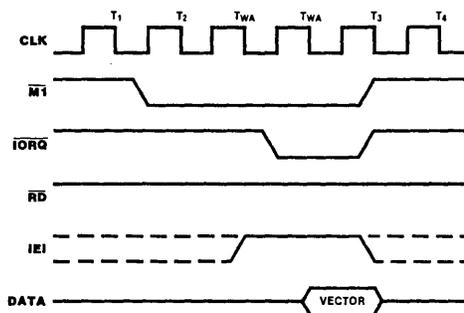


Figure 14. Interrupt Acknowledge Timing

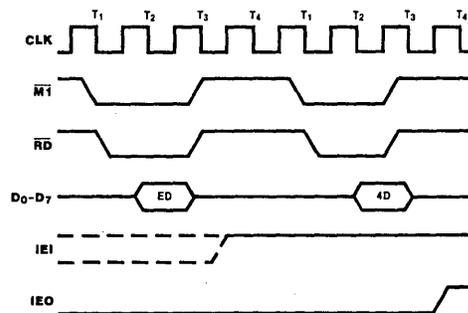


Figure 15. Return From Interrupt Timing

11.4    Appendix D - Floppy Disk controller ....

# WESTERN DIGITAL

C O R P O R A T I O N

## FD 179X-02 Floppy Disk Formatter/Controller Family

### FEATURES

- TWO VFO CONTROL SIGNALS
- SOFT SECTOR FORMAT COMPATIBILITY
- AUTOMATIC TRACK SEEK WITH VERIFICATION
- ACCOMMODATES SINGLE AND DOUBLE DENSITY FORMATS
  - IBM 3740 Single Density (FM)
  - IBM System 34 Double Density (MFM)
- READ MODE
  - Single/Multiple Sector Read with Automatic Search or Entire Track Read
  - Selectable 128 Byte or Variable length Sector
- WRITE MODE
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- SYSTEM COMPATIBILITY
  - Double Buffering of Data 8 Bit Bi-Directional Bus for Data, Control and Status
  - DMA or Programmed Data Transfers
  - All Inputs and Outputs are TTL Compatible
  - On-Chip Track and Sector Registers/Comprehensive Status Information

- PROGRAMMABLE CONTROLS
  - Selectable Track to Track Stepping Time
  - Side Select Compare
- WRITE PRECOMPENSATION
- WINDOW EXTENSION
- INCORPORATES ENCODING/DECODING AND ADDRESS MARK CIRCUITRY
- FD1792/4 IS SINGLE DENSITY ONLY
- FD1795/7 HAS A SIDE SELECT OUTPUT

MAY 1980

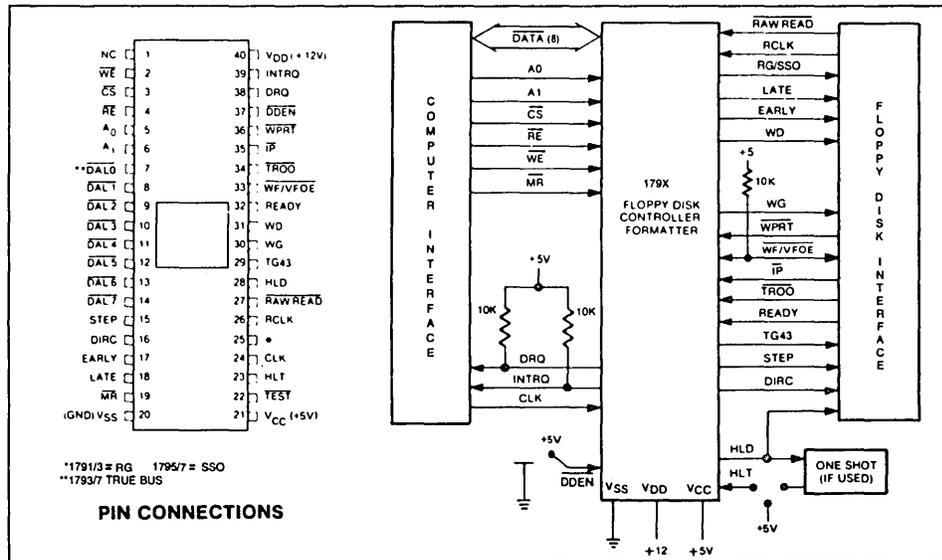


### 179X-02 FAMILY CHARACTERISTICS

FEATURES	1791	1793	1795	1797
Single Density (FM)	X	X	X	X
Double Density (MFM)	X	X	X	X
True Data Bus		X		X
Inverted Data Bus	X		X	
Write Precomp	X	X	X	X
Side Selection Output			X	X

### APPLICATIONS

- FLOPPY DISK DRIVE INTERFACE
- SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
- NEW MINI-FLOPPY CONTROLLER



FD179X SYSTEM BLOCK DIAGRAM

N 20-10100

**GENERAL DESCRIPTION**

The FD179X are MOS LSI devices which perform the functions of a Floppy Disk Formatter/Controller in a single chip implementation. The FD179X, which can be considered the end result of both the FD1771 and FD1781 designs, is IBM 3740 compatible in single density mode (FM) and System 34 compatible in Double Density Mode (MFM). The FD179X contains all the features of its predecessor the FD1771, plus the added features necessary to read/write and format a double density diskette. These include address mark detection, FM and MFM encode and decode logic, window extension, and write precompensation. In order to maintain compatibility, the FD1771, FD1781, and FD179X 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 bidirectional bus for data, status, and control word transfers. The FD179X is set up to operate on a multiplexed bus with other bus-oriented devices.

The FD179X is fabricated in N-channel Silicon Gate MOS technology and is TTL compatible on all inputs and outputs. The 1793 is identical to the 1791 except the DAL lines are TRUE for systems that utilize true data busses.

The 1795/7 has a side select output for controlling double sided drives, and the 1792 and 1794 are "Single Density Only" versions of the 1791 and 1793. On these devices, DDEN must be left open.

**PIN OUTS**

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION																				
1	NO CONNECTION	NC	Pin 1 is internally connected to a back bias generator and must be left open by the user.																				
19	MASTER RESET	$\overline{MR}$	A logic low on this input resets the device and loads HEX 03 into the command register. The Not Ready (Status Bit 7) is reset during $\overline{MR}$ 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 <sub>ss</sub>	Ground																				
21		V <sub>cc</sub>	+5V ±5%																				
40		V <sub>dd</sub>	+12V ±5%																				
<b>COMPUTER INTERFACE:</b>																							
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: <table border="0" style="margin-left: 20px;"> <tr> <td>A1</td> <td>A0</td> <td><math>\overline{RE}</math></td> <td><math>\overline{WE}</math></td> </tr> <tr> <td>0</td> <td>0</td> <td>Status Reg</td> <td>Command Reg</td> </tr> <tr> <td>0</td> <td>1</td> <td>Track Reg</td> <td>Track Reg</td> </tr> <tr> <td>1</td> <td>0</td> <td>Sector Reg</td> <td>Sector Reg</td> </tr> <tr> <td>1</td> <td>1</td> <td>Data Reg</td> <td>Data Reg</td> </tr> </table>	A1	A0	$\overline{RE}$	$\overline{WE}$	0	0	Status Reg	Command Reg	0	1	Track Reg	Track Reg	1	0	Sector Reg	Sector Reg	1	1	Data Reg	Data Reg
A1	A0	$\overline{RE}$	$\overline{WE}$																				
0	0	Status Reg	Command Reg																				
0	1	Track Reg	Track Reg																				
1	0	Sector Reg	Sector Reg																				
1	1	Data Reg	Data Reg																				
7-14	DATA ACCESS LINES	$\overline{DAL0-DAL7}$	Eight bit inverted Bidirectional bus used for transfer of data, control, and status. This bus is receiver enabled by $\overline{WE}$ or transmitter enabled by $\overline{RE}$ .																				
24	CLOCK	CLK	This input requires a free-running square wave clock for internal timing reference, 2 MHz for 8" drives, 1 MHz for mini-drives.																				

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
38	DATA REQUEST	DRQ	This open drain output indicates that the DR 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 in Read or Write operations, respectively. Use 10K pull-up resistor to +5.
39	INTERRUPT REQUEST	INTRQ	This open drain 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. Use 10K pull-up resistor to +5.
<b>FLOPPY DISK INTERFACE:</b>			
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	EARLY	EARLY	Indicates that the WRITE DATA pulse occurring while Early is active (high) should be shifted early for write precompensation.
18	LATE	LATE	Indicates that the write data pulse occurring while Late is active (high) should be shifted late for write precompensation.
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	This input is used for testing purposes only and should be tied to +5V or left open by the user unless interfacing to voice coil actuated motors.
23	HEAD LOAD TIMING	HLT	When a logic high is found on the HLT input the head is assumed to be engaged.
25	READ GATE (1791/3)	RG	A high level on this output indicates to the data separator circuitry that a field of zeros (or ones) has been encountered, and is used for synchronization.
25	SIDE SELECT OUTPUT (1795, 1797)	SSO	The logic level of the Side Select Output is directly controlled by the 'S' flag in Type II or III commands. When S = 1, SSO is set to a logic 1. When S = 0, SSO is set to a logic 0. 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	READ CLOCK	RCLK	A nominal square-wave clock signal derived from the data stream must be provided to this input. Phasing (i.e. RCLK transitions) relative to RAW READ is important but polarity (RCLK high or low) is not.
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.
	WRITE GATE	WG	This output is made valid before writing is to be performed on the diskette.

2 ZO-10MS

20-10ms

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
31	WRITE DATA	WD	A 250 ns (MFM) or 500 ns (FM) 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	$\overline{\text{WRITE FAULT}}$ $\overline{\text{VFO ENABLE}}$	$\overline{\text{WF/VFOE}}$	This is a bi-directional signal used to signify writing faults at the drive, and to enable the external PLO data separator. When $\text{WG} = 1$ , Pin 33 functions as a WF input. If $\text{WF} = 0$ , any write command will immediately be terminated. When $\text{WG} = 0$ , Pin 33 functions as a VFOE output. VFOE will go low during a read operation after the head has loaded and settled ( $\text{HLT} = 1$ ). On the 1795/7, it will remain low until the last bit of the second CRC byte in the ID field. VFOE will then go high until 8 bytes (MFM) or 4 bytes (FM) before the Address Mark. It will then go active until the last bit of the second CRC byte of the Data Field. On the 1791/3, VFOE will remain low until the end of the Data Field.
34	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This input informs the FD179X that the Read/Write head is positioned over Track 00.
35	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	This input informs the FD179X when the index hole is encountered on the diskette.
36	$\overline{\text{WRITE PROTECT}}$	$\overline{\text{WPRT}}$	This input is sampled whenever a Write Command is received. A logic low terminates the command and sets the Write Protect Status bit.
37	$\overline{\text{DOUBLE DENSITY}}$	$\overline{\text{DDEN}}$	This pin selects either single or double density operation. When $\overline{\text{DDEN}} = 0$ , double density is selected. When $\overline{\text{DDEN}} = 1$ , single density is selected. This line must be left open on the 1792/4

#### ORGANIZATION

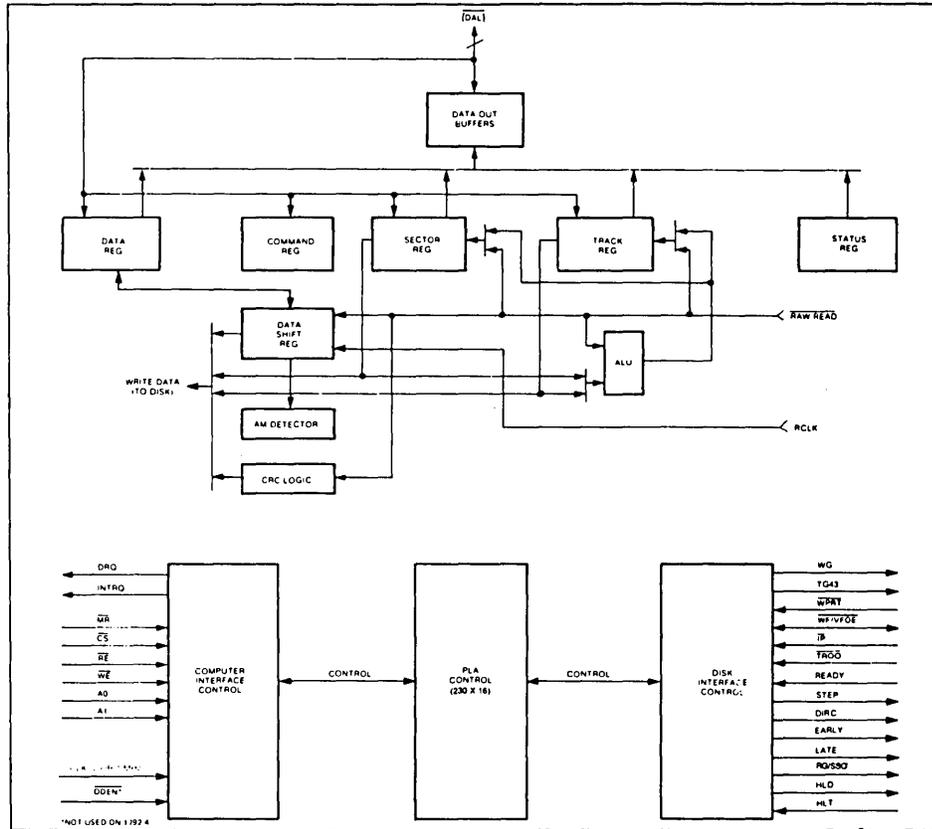
The Floppy Disk Formatter block diagram is illustrated on page 5. The primary sections include the parallel processor interface and the Floppy Disk interface.

**Data Shift Register**—This 8-bit register assembles serial data from the Read Data input ( $\overline{\text{RAW READ}}$ ) 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.



FD179X BLOCK DIAGRAM

**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 decremter 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.

The FD1791/3 has two different modes of operation according to the state of DDEN. When DDEN = 0 double density (MF) is assumed. When DDEN = 1, single density (FM) is assumed.

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

### 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 FD179X. The DAL are three state buffers that are enabled as output drivers when Chip Select (CS) and Read Enable ( $\overline{RE}$ ) are active (low logic state) or act as input receivers when CS and Write Enable (WE) are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and 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 FD179X 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.

### FLOPPY DISK INTERFACE

The 179X has two modes of operation according to the state of DDEN (Pin 37). When DDEN = 1, single density is selected. In either case, the CLK input (Pin 24) is at 2 MHz. However, when interfacing with the mini-floppy, the CLK input is set at 1 MHz for both single density and double density. When the clock is at 2 MHz, the stepping rates of 3, 6, 10, and 15 ms are obtainable. When CLK equals 1 MHz these times are doubled.

### HEAD POSITIONING

Five commands cause positioning of the Read-Write head (see Command Section). The period of each positioning step is specified by the r field in bits 1 and 0 of the command word. 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. If  $\overline{TEST} = 0$ , there is zero settling time. There is also a 15 ms head settling time if the E flag is set in any Type II or III command.

The rates (shown in Table 1) can be applied to a Step-Direction Motor through the device interface.

**Step**—A 2  $\mu$ s (MFM) or 4  $\mu$ s (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output.

**Direction (DIRC)**—The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 12  $\mu$ s before the first stepping pulse is generated.

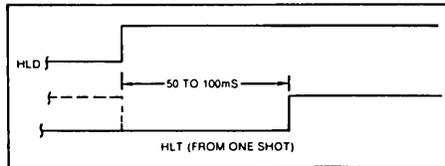
When a Seek, Step or Restore command is executed an optional verification of Read-Write head position can be performed by setting bit 2 ( $V = 1$ ) in the command word to a logic 1. The verification operation begins at the end of the 15 millisecond settling time after the head is loaded against the media. 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. The FD179X 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.

Table 1. STEPPING RATES

CLK	2 MHz	2 MHz	1 MHz	1 MHz	2 MHz	1 MHz
DDEN	0	1	0	1	x	x
R1 R0	TEST=1	TEST=1	TEST=1	TEST=1	TEST=0	TEST=0
0 0	3 ms	3 ms	6 ms	6 ms	184 $\mu$ s	368 $\mu$ s
0 1	6 ms	6 ms	12 ms	12 ms	190 $\mu$ s	380 $\mu$ s
1 0	10 ms	10 ms	20 ms	20 ms	198 $\mu$ s	396 $\mu$ s
1 1	15 ms	15 ms	30 ms	30 ms	208 $\mu$ s	416 $\mu$ s

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 FD179X is in an idle state (non-busy) and 15 index pulses have occurred.

Head Load Timing (HLT) is an input to the FD179X which is used for the head engage time. When HLT = 1, the FD179X 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 FD179X.



HEAD LOAD TIMING

When both HLD and HLT are true, the FD179X will then read from or write to the media. The "and" of HLD and HLT appears as a status bit 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 FD179X 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 FD179X 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.

#### 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 a special byte in the "ID" field. If this Sector length byte in the ID field is zero, then the sector length is 128 bytes. If 01 then 256 bytes. If 02, then 512 bytes. If 03, then the sector length is 1024 bytes. The number of sectors per track as far as the FD179X is concerned can be from 1 to 255 sectors. The number of tracks as far as the FD179X 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. (See Sector Length Table.)

For read operations, the FD179X requires  $\overline{RAW}$  READ Data (Pin 27) signal which is a 250 ns pulse per flux transition and a Read clock (RCLK) signal to indicate flux transition spacings. The RCLK (Pin 26) signal is provided by some drives but if not it may be

derived externally by Phase lock loops, one shots, or counter techniques. In addition, a Read Gate Signal is provided as an output (Pin 25) which can be used to inform phase lock loops when to acquire synchronization. When reading from the media in FM, RG is made true when 2 bytes of zeroes are detected. The FD179X must find an address mark within the next 10 bytes; otherwise RG is reset and the search for 2 bytes of zeroes begins all over again. If an address mark is found within 10 bytes, RG remains true as long as the FD179X is deriving any useful information from the data stream. Similarly for MFM, RG is made active when 4 bytes of "00" or "FF" are detected. The FD179X must find an address mark within the next 16 bytes, otherwise RG is reset and search resumes.

During read operations ( $WG = 0$ ), the  $\overline{VFOE}$  (Pin 33) is provided for phase lock loop synchronization.  $\overline{VFOE}$  will go active when:

- Both HLT and HLD are True
- Settling Time, if programmed, has expired
- The 179X is inspecting data off the disk

If  $\overline{WF/VFOE}$  is not used, leave open or tie to a 10K resistor to +5.

#### 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 FD179X 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. The Write Fault input, when activated, signifies a writing fault condition detected in disk drive electronics such as failure to detect write current flow when the Write Gate is activated. On detection of this fault the FD179X terminates the current command, and sets the Write Fault bit (bit 5) in the Status Word. The Write Fault input should be made inactive when the Write Gate output becomes inactive.

For write operations, the FD179X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of 500 ns pulses in FM ( $\overline{DDEN} = 1$ ) and 250 ns pulses in MFM ( $\overline{DDEN} = 0$ ). Write Data provides the unique address marks in both formats.

Also during write, two additional signals are provided for write precompensation. These are EARLY (Pin 17) and LATE (Pin 18). EARLY is active true when the WD pulse appearing on (Pin 30) is to be written early. LATE is active true when the WD pulse is to be written LATE. If both EARLY and LATE are low when the WD pulse is present, the WD pulse is to be written at nominal. Since write precompensation values vary from disk manufacturer to disk manufacturer, the actual value is determined by several one shots or delay lines which are located external to the FD179X. The write precompensation signals EARLY and LATE are valid for the duration of WD in both FM and MFM formats.

Whenever a Read or Write command (Type II or III) is received the FD179X 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.

**COMMAND DESCRIPTION**

The FD179X 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 2.

**Table 2 COMMAND SUMMARY**

		BITS							
TYPE	COMMAND	7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r <sub>1</sub>	r <sub>0</sub>
I	Seek	0	0	0	1	h	V	r <sub>1</sub>	r <sub>0</sub>
I	Step	0	0	1	u	h	V	r <sub>1</sub>	r <sub>0</sub>
I	Step In	0	1	0	u	h	V	r <sub>1</sub>	r <sub>0</sub>
I	Step Out	0	1	1	u	h	V	r <sub>1</sub>	r <sub>0</sub>
II	Read Sector	1	0	0	m	F <sub>2</sub>	E	F <sub>1</sub>	0
II	Write Sector	1	0	1	m	F <sub>2</sub>	E	F <sub>1</sub>	a <sub>0</sub>
III	Read Address	1	1	0	0	0	0	0	0
III	Read Track	1	1	1	0	0	0	0	0
III	Write Track	1	1	1	1	0	0	0	0
IV	Force Interrupt	1	1	0	1	l <sub>3</sub>	l <sub>2</sub>	l <sub>1</sub>	l <sub>0</sub>

Note: Bits shown in TRUE form.

**Table 3 FLAG SUMMARY**

TYPE I COMMANDS
<b>h = Head Load Flag (Bit 3)</b> h = 1, Load head at beginning h = 0, Unload head at beginning
<b>V = Verify flag (Bit 2)</b> V = 1, Verify on destination track V = 0, No verify
<b>r<sub>1</sub>r<sub>0</sub> = Stepping motor rate (Bits 1-0)</b> Refer to Table 1 for rate summary
<b>u = Update flag (Bit 4)</b> u = 1, Update Track register u = 0, No update

**Table 4 FLAG SUMMARY**

TYPE II & III COMMANDS																			
<b>m = Multiple Record flag (Bit 4)</b> m = 0, Single Record m = 1, Multiple Records																			
<b>a<sub>0</sub> = Data Address Mark (Bit 0)</b> a <sub>0</sub> = 0, FB (Data Mark) a <sub>0</sub> = 1, F8 (Deleted Data Mark)																			
<b>E = 15 ms Delay (2MHz)</b> E = 1, 15 ms delay E = 0, no 15 ms delay																			
<b>(F<sub>2</sub>) S = Side Select Flag (1791/3 only)</b> S = 0, Compare for Side 0 S = 1, Compare for Side 1																			
<b>(F<sub>1</sub>) C = Side Compare Flag (1791/3 only)</b> C = 0, disable side select compare C = 1, enable side select compare																			
<b>(F<sub>1</sub>) S = Side Select Flag (Bit 1, 1795/7 only)</b> S = 0 Update SSO to 0 S = 1 Update SSO to 1																			
<b>(F<sub>2</sub>) b = Sector Length Flag (Bit 3, 1975/7 only)</b>																			
<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th rowspan="2"></th> <th colspan="4">Sector Length Field</th> </tr> <tr> <th>00</th> <th>01</th> <th>10</th> <th>11</th> </tr> </thead> <tbody> <tr> <td>b = 0</td> <td>256</td> <td>512</td> <td>1024</td> <td>128</td> </tr> <tr> <td>b = 1</td> <td>128</td> <td>256</td> <td>512</td> <td>1024</td> </tr> </tbody> </table>		Sector Length Field				00	01	10	11	b = 0	256	512	1024	128	b = 1	128	256	512	1024
		Sector Length Field																	
	00	01	10	11															
b = 0	256	512	1024	128															
b = 1	128	256	512	1024															

**Table 5 FLAG SUMMARY**

TYPE IV COMMAND
<b>li = Interrupt Condition flags (Bits 3-0)</b> l <sub>0</sub> = 1, Not-Ready to Ready Transition l <sub>1</sub> = 1, Ready to Not-Ready Transition l <sub>2</sub> = 1, Index Pulse l <sub>3</sub> = 1, Immediate Interrupt l <sub>3</sub> -l <sub>0</sub> = 0, Terminate with no Interrupt

**TYPE I COMMANDS**

The Type I Commands include the Restore, Seek, Step, Step-In, and Step-Out commands. Each of the Type I Commands contains a rate field (r<sub>0</sub>r<sub>1</sub>), which determines the stepping motor rate as defined in Table 1.

The Type I Commands contain a head load flag (h) which determines if the head is to be loaded at the beginning of the command. If  $h = 1$ , the head is loaded at the beginning of the command (HLD output is made active). If  $h = 0$ , HLD is deactivated. Once the head is loaded, the head will remain engaged until the FD179X receives a command that specifically disengages the head. If the FD179X is idle (busy = 0) for 15 revolutions of the disk, the head will be automatically disengaged (HLD made inactive).

The Type I Commands also contain a verification (V) flag which determines if a verification operation is to take place on the destination track. If  $V = 1$ , a verification is performed, if  $V = 0$ , no verification is performed.

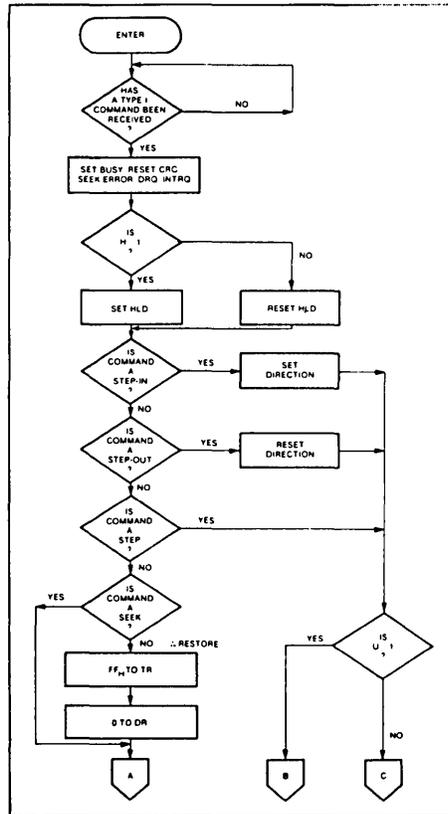
During verification, the head is loaded and after an internal 15 ms delay, the HLT input is sampled. When HLT is active (logic true), the first encountered ID field is read off the disk. The track address of the

ID field is then compared to the Track Register; if there is a match and a valid ID CRC, the verification is complete, an interrupt is generated and the Busy status bit is reset. If there is not a match but there is valid ID CRC, an interrupt is generated, and Seek Error Status bit (Status bit 4) is set and the Busy status bit is reset. 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. If an ID field with a valid CRC cannot be found after four revolutions of the disk, the FD179X terminates the operation and sends an interrupt, (INTRQ).

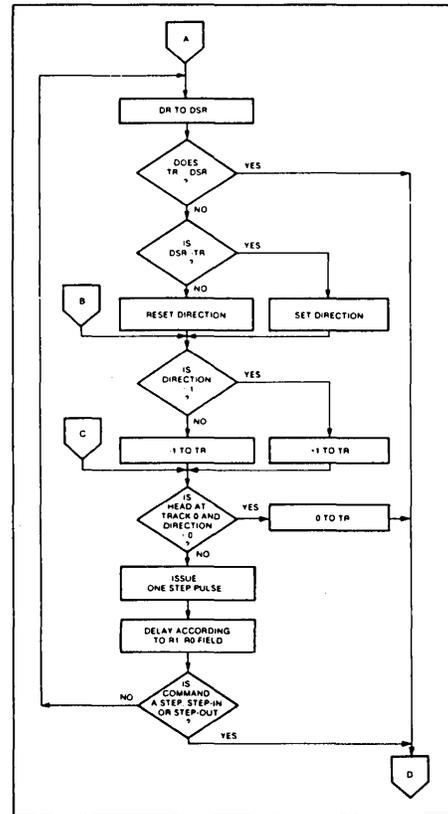
The Step, Step-In, and Step-Out commands contain an Update flag (U). When  $U = 1$ , the track register is updated by one for each step. When  $U = 0$ , the track register is not updated.

On the 1795/7 devices, the SSO output is not affected during Type 1 commands, and an internal side compare does not take place when the (V) Verify Flag is on.

NO-10MS



TYPE I COMMAND FLOW



TYPE I COMMAND FLOW

### RESTORE (SEEK TRACK 0)

Upon receipt of this command the Track 00 (TROO) input is sampled. If TROO 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 TROO is not active low, stepping pulses (pins 15 to 16) at a rate specified by the rro field are issued until the TROO input is activated. At this time the Track Register is loaded with zeroes and an interrupt is generated. If the TROO input does not go active low after 255 stepping pulses, the FD179X 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 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 FD179X will update the Track register and issue stepping pulses in the appropriate direction until the contents of the Track register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

### STEP

Upon receipt of this command, the FD179X 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 rro field, a verification takes place if the V flag is on. If the u flag is on, the Track Register is updated. The h bit allows the 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 FD179X issues one stepping pulse in the direction towards track 76. If the u flag is on, the Track Register is incremented by one. After a delay determined by the rro 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.

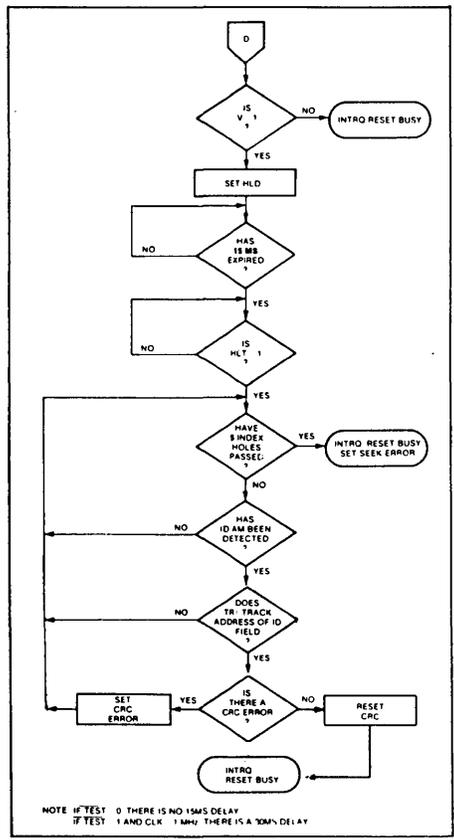
### STEP-OUT

Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 0. If the u flag is on, the Track Register is decremented by one. After a delay determined by the rro 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.

### 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. The ID field and Data Field format are shown on page 13.

When an ID field is located on the disk, the FD179X compares the Track Number on the ID field with the Track Register. If there is not a match, the next en-



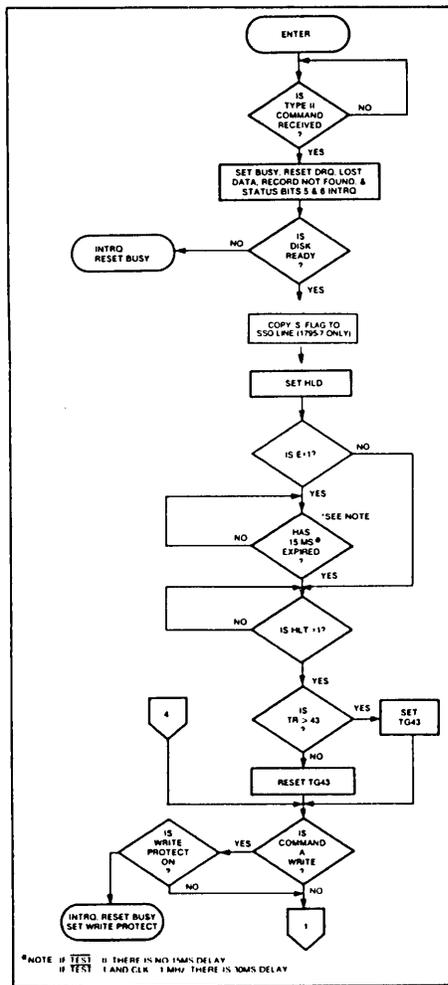
TYPE I COMMAND FLOW

countered ID field is read and a comparison is again made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from depending upon the command. The FD179X must find an ID field with a Track number, Sector number, side number, and CRC within four revolutions of the disk; otherwise, the Record not found status bit is set (Status bit 3) and the command is terminated with an interrupt.

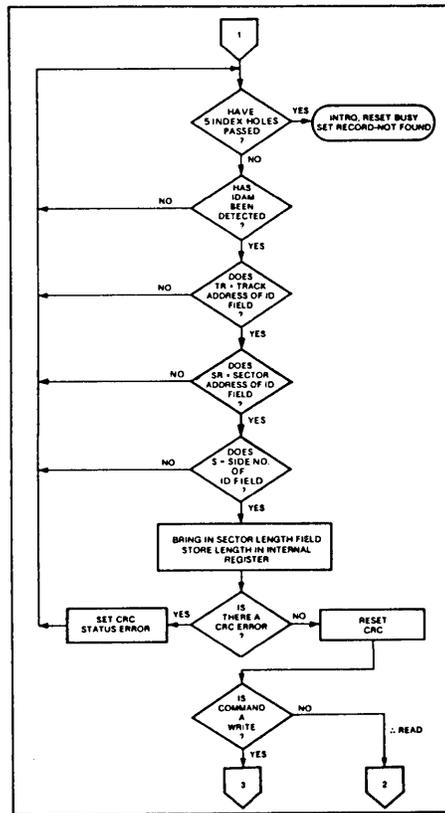
Sector Length Table	
Sector Length Field (hex)	Number of Bytes in Sector (decimal)
00	128
01	256
02	512
03	1024

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 FD179X will continue to read or write multiple records and update the sector register until the sector regis-

20-1010



TYPE II COMMAND



TYPE II COMMAND

2 ZO-01010

ter 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.

If the Sector Register exceeds the number of sectors on the track, the Record-Not-Found status bit will be set.

The Type II commands also contain side select compare flags. When C = 0, 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. If the S flag compares with the side number recorded in the ID field, the 179X 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.

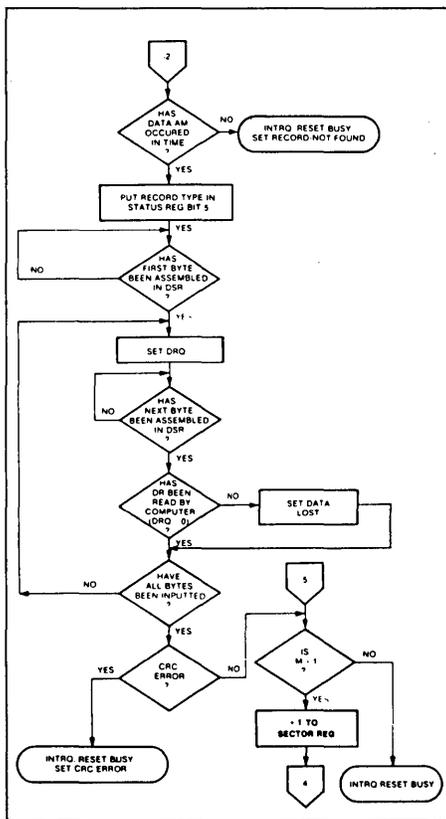
The 1795/7 READ SECTOR and WRITE SECTOR commands include a 'b' flag. The 'b' flag, in conjunction with the sector length byte of the ID Field, allows different byte lengths to be implemented in each sector. For IBM compatibility, the 'b' flag should be set to a one. The

's' flag allows direct control over the SSO Line (Pin 25) and is set or reset at the beginning of the command, dependent upon the value of this flag.

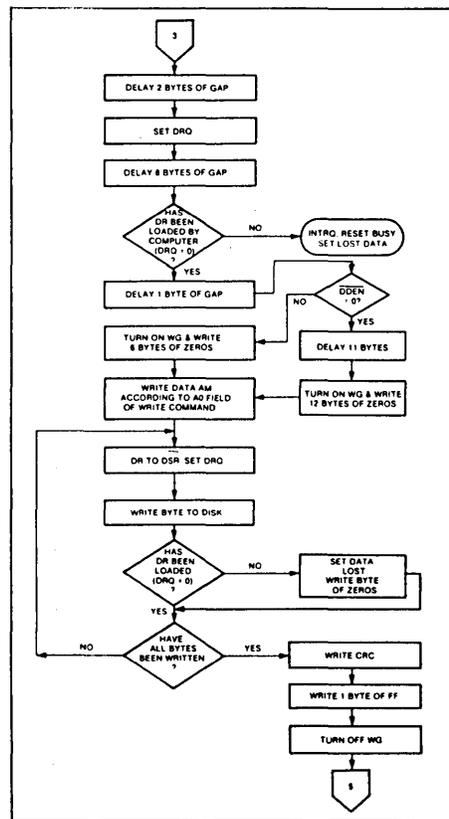
### READ SECTOR

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 Record Not Found status bit is set and the operation is terminated.

When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the Computer has not read the previous contents of the DR before a new character is transferred that character is lost and



TYPE II COMMAND



TYPE II COMMAND

the Lost Data Status bit is set. This sequence continues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple record command).

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

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 FD179X 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 zeros 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 FD179X 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 zeros is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one byte of logic ones in FM or in MFM. The WG output is then deactivated.

**TYPE 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 FD179X 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. At the end of the operation an interrupt is generated and the Busy Status is reset.

**READ TRACK**

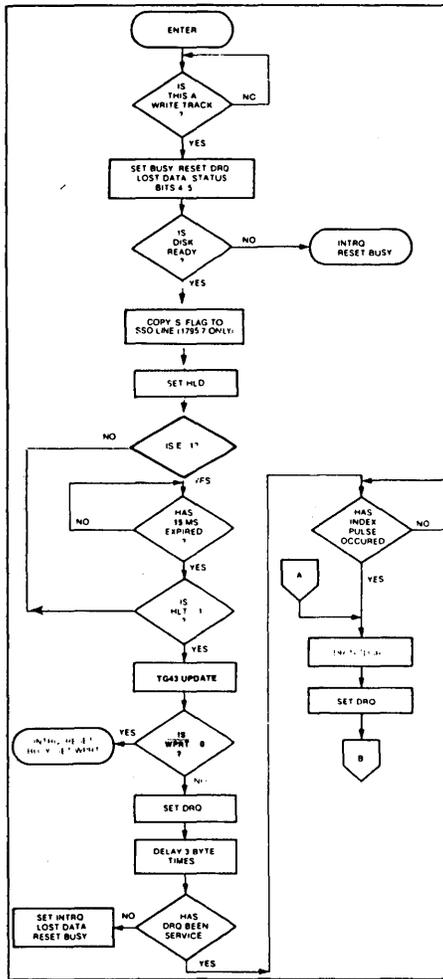
Upon receipt of the Read Track command, the head is loaded and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. As each byte is assembled it is transferred to the Data Register and the Data Request is generated for each byte. No CRC checking is performed. Gaps are included in the input data stream. The accumulation of bytes is synchronized to each Address Mark encountered. Upon completion of the command, the interrupt is activated. RG is not activated during the Read Track Command. An internal side compare is not performed during a Read Track.

**WRITE TRACK**

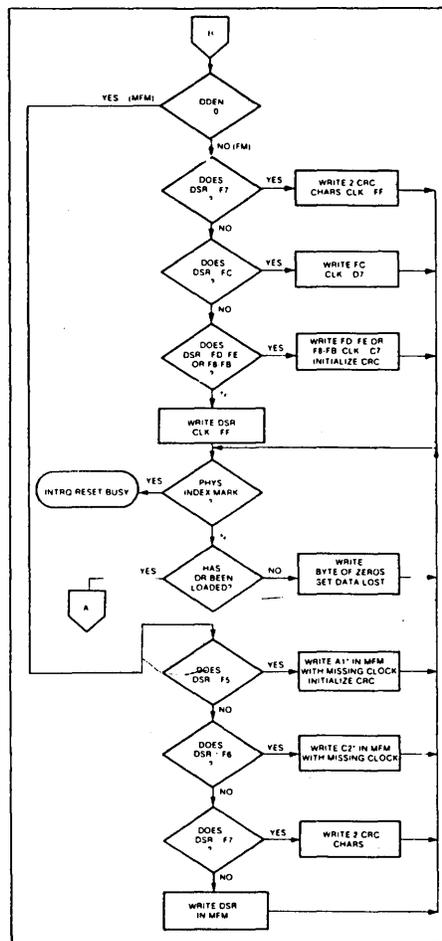
Upon receipt of the Write Track command, the head is loaded and the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving 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 zeros is substituted. Address Marks and CRC characters are written on the disk by detecting certain data byte patterns in the outgoing data stream as shown in the table below. The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR in FM or by receipt of F5 in MFM.

GAP III	ID AM	TRACK NUMBER	SIDE NUMBER	SECTOR NUMBER	SECTOR LENGTH	CRC 1	CRC 2	GAP II	DATA AM	DATA FIELD	CRC 1	CRC 2
ID FIELD										DATA FIELD		

In MFM only, IDAM and DATA AM are preceded by three bytes of A1 with clock transition between bits 4 and 5 missing.



TYPE III COMMAND WRITE TRACK



TYPE III COMMAND WRITE TRACK

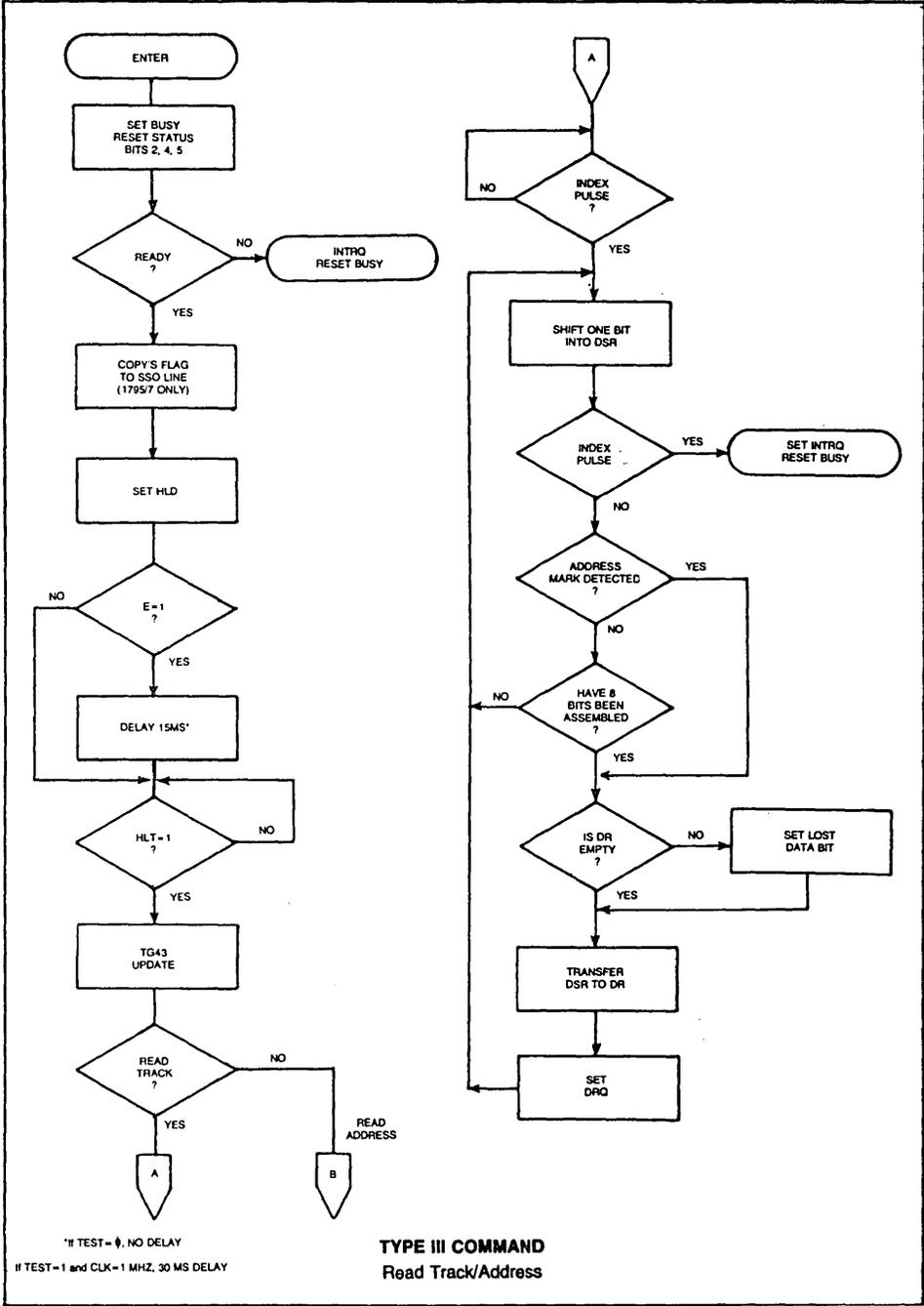
CONTROL BYTES FOR INITIALIZATION

DATA PATTERN IN DR (HEX)	FD179X INTERPRETATION IN FM (DDEN = 1)	FD1791/3 INTERPRETATION IN MFM (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

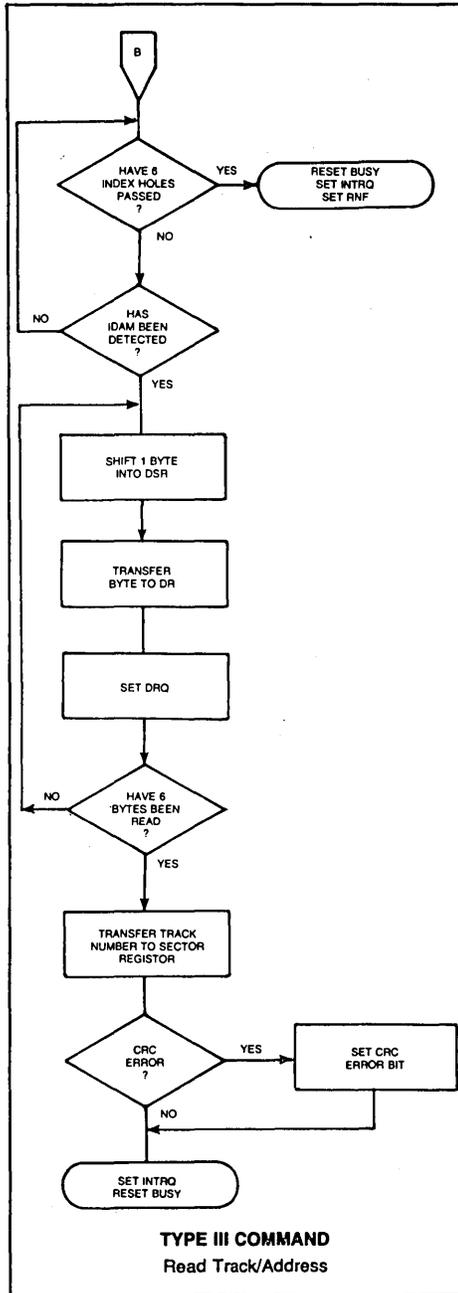
\*Missing clock transition between bits 4 and 5

\*\*Missing clock transition between bits 3 & 4

20-1010



SECTION 2



**TYPE IV COMMAND**

**FORCE INTERRUPT**

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 an interrupt will be generated when the condition specified in the  $I_0$  through  $I_3$  field is detected. The interrupt conditions are shown below:

- $I_0$  = Not-Ready-To-Ready Transition
- $I_1$  = Ready-To-Not-Ready Transition
- $I_2$  = Every Index Pulse
- $I_3$  = Immediate Interrupt (requires reset, see Note)

**NOTE:** If  $I_0 - I_3 = 0$ , there is no interrupt generated but the current command is terminated and busy is reset. This is the only command that will enable the immediate interrupt to clear on a subsequent Load Command Register or Read Status Register.

**STATUS DESCRIPTION**

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 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 6.

## 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 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 FD179X raises the Data Request signal. At this point in time, the user loads the data register with desired data to be written on the disk. For every byte of information to be written on the disk, a data request is generated. 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 FD179X 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. For instance, in FM an FE pattern will be interpreted as an ID address mark (DATA-FE, CLK-C7) and the CRC will be initialized. 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.

## IBM 3740 FORMAT—128 BYTES/SECTOR

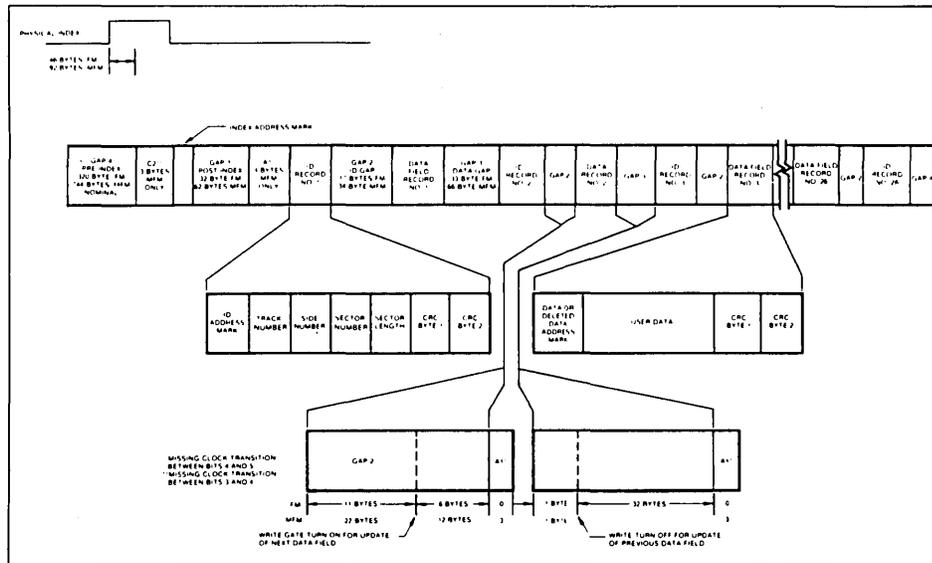
Shown below is the IBM single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00) <sup>1</sup>
6	00
1	FC (Index Mark)
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
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**	FF (or 00)

\*Write bracketed field 26 times

\*\*Continue writing until FD179X interrupts out. Approx. 247 bytes.

1-Optional '00' on 1795/7 only.



**IBM TRACK FORMAT**

5  
 20-10MS  
 2

**IBM SYSTEM 34 FORMAT-  
256 BYTES/SECTOR**

Shown below is the IBM dual-density format with 256 bytes/sector. In order to format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
80	4E
12	00
3	F6
1	FC (Index Mark)
50*	4E
12	00
3	F5
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
1	F7 (2 CRCs written)
22	4E
12	00
3	F5
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
598**	4E

\* Write bracketed field 26 times  
 \*\*Continue writing until FD179X interrupts out.  
 Approx. 598 bytes.

**1. NON-IBM FORMATS**

Variations in the IBM format are possible to a limited extent if the following requirements are met: sector size must be a choice of 128, 256, 512, or 1024 bytes; gap size must be according to the following table. Note that the Index Mark is not required by the 179X. The minimum gap sizes shown are that which is required by the 179X, with PLL lock-up time, motor speed variation, etc., adding additional bytes.

	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 3 bytes A1
**	4 bytes 00	8 bytes 00
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**

**MAXIMUM RATINGS**

V<sub>DD</sub> With Respect to V<sub>SS</sub> (Ground) = 15 to -0.3V  
 Max. Voltage to Any Input With Respect to V<sub>SS</sub> = 15 to -0.3V

Operating Temperature 0°C to 70°C  
 Storage Temperature -55°C to +125°C

V<sub>DD</sub> = 10 ma Nominal V<sub>CC</sub> = 35 ma Nominal

**OPERATING CHARACTERISTICS (DC)**

TA = 0°C to 70°C, V<sub>DD</sub> = + 12V ± .6V, V<sub>SS</sub> = 0V, V<sub>CC</sub> = + 5V ± .25V

SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNITS	CONDITIONS
I <sub>IL</sub>	Input Leakage		10	μA	V <sub>IN</sub> = V <sub>DD</sub>
I <sub>OL</sub>	Output Leakage		10	μA	V <sub>OUT</sub> = V <sub>DD</sub>
V <sub>IH</sub>	Input High Voltage	2.6		V	
V <sub>IL</sub>	Input Low Voltage		0.8	V	
V <sub>OH</sub>	Output High Voltage	2.8		V	I <sub>O</sub> = -100 μA
V <sub>OL</sub>	Output Low Voltage		0.45	V	I <sub>O</sub> = 1.6 mA
P <sub>D</sub>	Power Dissipation		0.5	W	

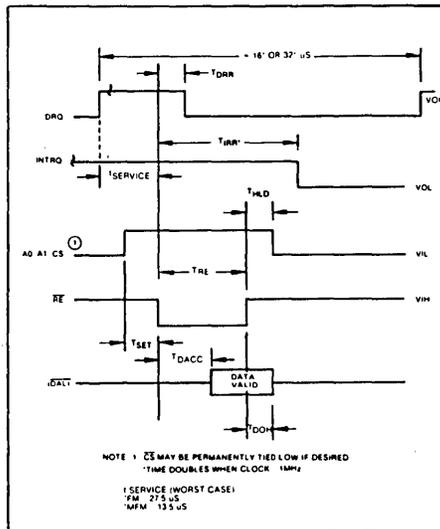
### TIMING CHARACTERISTICS

T<sub>A</sub> = 0°C to 70°C, V<sub>DD</sub> = +12V ± .6V, V<sub>SS</sub> = 0V, V<sub>CC</sub> = +5V ± .25V

#### READ ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{RE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{RE}$	10			nsec	
TRE	$\overline{RE}$ Pulse Width	400			nsec	C <sub>L</sub> = 50 pf
TDRR	DRQ Reset from $\overline{RE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{RE}$		500	3000	nsec	See Note 5
TDACC	Data Access from $\overline{RE}$			350	nsec	C <sub>L</sub> = 50 pf
TDOH	Data Hold From $\overline{RE}$	50		150	nsec	C <sub>L</sub> = 50 pf

20-010ms



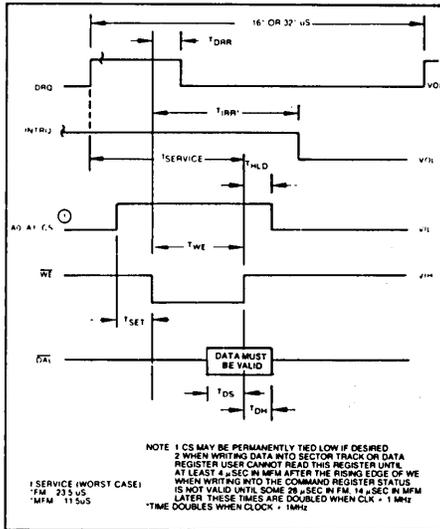
READ ENABLE TIMING

**WRITE ENABLE TIMING**

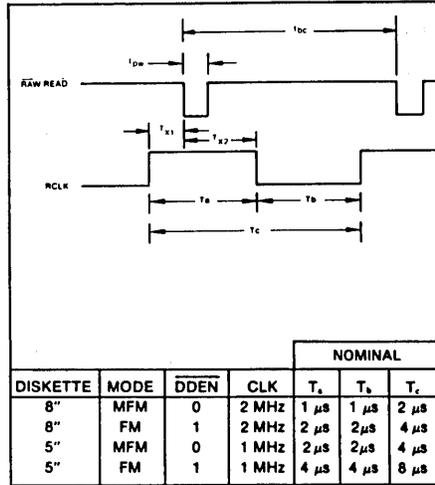
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{WE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{WE}$	10			nsec	
TWE	$\overline{WE}$ Pulse Width	350			nsec	
TDRR	DRQ Reset from $\overline{WE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{WE}$		500	3000	nsec	See Note 5
TDS	Data Setup to $\overline{WE}$	250			nsec	
TDH	Data Hold from $\overline{WE}$	70			nsec	

**INPUT DATA TIMING:**

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Tpw	Raw Read Pulse Width	100	200		nsec	See Note 1
tbc	Raw Read Cycle Time		1500		nsec	1800 ns @ 70°C
Tc	RCLK Cycle Time		1500		nsec	1800 ns @ 70°C
Tx1	RCLK hold to Raw Read	40			nsec	See Note 1
Tx2	Raw Read hold to RCLK	40			nsec	



**WRITE ENABLE TIMING**

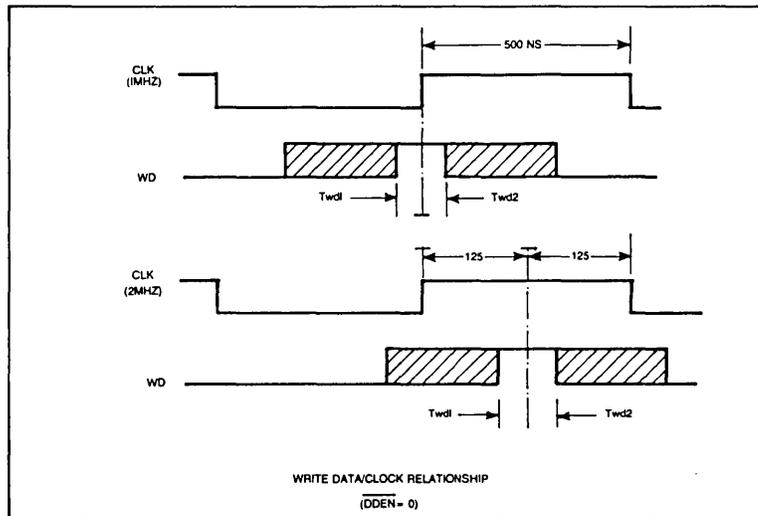


**INPUT DATA TIMING**

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

SYMBOL	CHARACTERISTICS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Twp	Write Data Pulse Width	450	500	550	nsec	FM
		150	200	250	nsec	MFM
Twg	Write Gate to Write Data		2		$\mu$ sec	FM
			1		$\mu$ sec	MFM
Tbc	Write data cycle Time		2,3, or 4		$\mu$ sec	$\pm$ CLK Error
Ts	Early (Late) to Write Data	125			nsec	MFM
Th	Early (Late) From Write Data	125			nsec	MFM
			2		$\mu$ sec	FM
Twf	Write Gate off from WD		1		$\mu$ sec	MFM
					$\mu$ sec	
Twd1	WD Valid to Clk	100			nsec	CLK=1 MHZ
Twd2	WD Valid after CLK	50			nsec	CLK=2 MHZ
		100			nsec	CLK=1 MHZ
		30			nsec	CLK=2 MHZ

2 Z0-0016

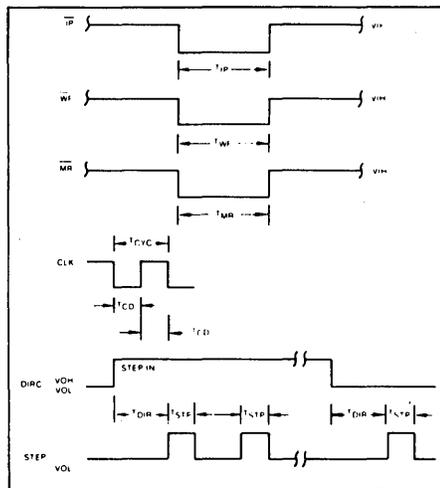


**WRITE DATA TIMING**

**MISCELLANEOUS TIMING:**

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TCD <sub>1</sub>	Clock Duty (low)	230	250	20000	nsec	See Note 5 ± CLK ERROR
TCD <sub>2</sub>	Clock Duty (high)	200	250	20000	nsec	
TSTP	Step Pulse Output	2 or 4			μsec	
TDIR	Dir Setup to Step		12		μsec	See Note 5
TMR	Master Reset Pulse Width	50			μsec	
TIP	Index Pulse Width	10			μsec	
TWF	Write Fault Pulse Width	10			μsec	

N 20-0000



**MISCELLANEOUS TIMING**

**NOTES:**

1. Pulse width on RAW READ (Pin 27) is normally 100-300 ns. However, pulse may be any width if pulse is entirely within window. If pulse occurs in both windows, then pulse width must be less than 300 ns for MFM at CLK = 2 MHz and 600 ns for FM at 2 MHz. Times double for 1 MHz.
2. A PPL Data Separator is recommended for 8" MFM.
3. tbc should be 2 μs, nominal in MFM and 4 μs nominal in FM. Times double when CLK = 1 MHz.
4. RCLK may be high or low during RAW READ (Polarity is unimportant).
5. Times double when clock = 1 MHz.

# WESTERN DIGITAL

C O R P O R A T I O N

## FD179X Application Notes

NOVEMBER, 1980

SD-10-2

### INTRODUCTION

Over the past several years, the Floppy Disk Drive has become the most popular on-line storage device for mini and microcomputer systems. Its fast access time, reliability and low cost-per-bit ratio enables the Floppy Disk Drive to be the solution in mass storage for microprocessor systems. The drive interface to the Host system is standardized, allowing the OEM to substitute one drive for another with minimum hardware/software modifications.

Since Floppy Disk Data is stored and retrieved as a self-clocking serial data stream, some means of separating the clock from the data and assembling this data in parallel form must be accomplished. Data is stored on individual Tracks of the media, requiring control of a stepper motor to move the Read/Write head to a predetermined Track. Byte synchronization must also be accomplished to insure that the parallel data is properly assembled. After all the design considerations are met, the final controller can consist of 40 or more TTL packages.

To alleviate the burden of Floppy Disk Controller design, Western Digital has developed a Family of LSI Floppy Disk controller devices. Through its own set of macro commands, the FD179X Controller Family will perform all the functions necessary to read and write data to the drive. Both the 8" standard and 5¼" mini-floppy are supported with single or double density recording techniques. The FD179X is compatible with the IBM 3740 (FM) data format, or the System 34 (MFM) standards. Provisions for non-standard formats and variable sector lengths have been included to provide more storage capability per track. Requiring standard +5, +12 power supplies the FD179X is available in a standard 40 pin dual-in-line package.

The FD179X Family consists of 6 devices. The differences between these devices is summarized in Figure 1. The 1792 and 1794 are "single density only" devices, with the Double Density Enable pin (DDEN) left open by the user. Both True and inverted Data bus devices are available. Since the 179X can only drive one TTL Load, a true data bus system may use the 1791 with external inverting buffers to arrive at a true bus scheme. The 1795 and 1797 are identical to the 1791 and 1793, except a side select output has been added that is controlled through the Command Register.

### SYSTEM DESIGN

The first consideration in Floppy Disk Design is to determine which type of drive to use. The choice ranges from single-density single sided mini-floppy to the 8" double-density double-sided drive. Figure 2 illustrates the various drive and data capacities associated with each type. Although the 8" double-density drive offers twice as much storage, a more complex data separator and the addition of Write Precompensation circuits are mandatory for reliable data transfers. Whether to go with 8" double-density or not is dependent upon PC board space and the additional circuitry needed to accurately recover data with extreme bit shifts. The byte transfer time defines the nominal time required to transfer one byte of data from the drive. If the CPU used cannot service a byte in this time, then a DMA scheme will probably be required. The 179X also needs a few microseconds for overhead, which is subtracted from the transfer time. Figure 3 shows the actual service times that the CPU must provide on a byte-by-byte basis. If these times are not met, bytes of data will be lost during a read or write operation. For each byte transferred, the 179X generates a DRQ (Data Request) signal on Pin 38. A bit is provided in the status register which is also set upon receipt of a byte from the Disk. The user has the option of reading the status register through program control or using the DRQ Line with DMA or interrupt schemes. When the data register is read, both the status register DRQ bit and the DRQ Line are automatically reset. The next full byte will again set the DRQ and the process continues until the sector(s) are read. The Write operation works exactly the same way, except a WRITE to the Data Register causes a reset of both DRQ's.

### RECORDING FORMATS

The FD179X accepts data from the disk in a Frequency-Modulated (FM) or Modified-Frequency-Modulated (MFM) Format. Shown in Figures 4A and 4B are both these Formats when writing a Hexidecimal byte of 'D2'. In the FM mode, the 8 bits of data are broken up into "bit cells." Each bit cell begins with a clock pulse and the center of the bit cell defines the data. If the data bit = 0, no pulse is written; if the data = 1, a pulse is written in the center of the cell. For the 8" drive, each clock is written 4 microseconds apart.

In the MFM mode, clocks are decoded into the data stream. The byte is again broken up into bit cells, with the data bit written in the center of the bit cell if data = 1. Clocks are only written if both surrounding data bits are zero. Figure 4B shows that this occurs only once between Bit cell 4 and 5. Using this encoding scheme, pulses can occur 2, 3 or 4 microseconds apart. The bit cell time is now 2 microseconds; twice as much data can be recorded without increasing the Frequency rate due to this encoding scheme.

The 179X was designed to be compatible with the IBM 3740 (FM) and System 34 (MFM) Formats. Although most users do not have a need for data exchange with IBM mainframes, taking advantage of these well studied formats will insure a high degree of system performance. The 179X will allow a change in gap fields and sector lengths to increase usable storage capacity, but variations away from these standards is not recommended. Both IBM standards are soft-sector format. Because of the wide variation in address marks, the 179X can only support soft-sectored media. Hard sectored diskettes have continued to lose popularity, mainly due to the unavailability of a standard and the limitation of sector lengths imposed by the physical sector holes in the diskette.

**PROCESSOR INTERFACE**

The Interface of the 179X to the CPU consists of an 8-bit Bi-directional bus, read/write controls and optional interrupt lines. By selecting the device via the CHIP SELECT Line, each of the five internal registers can be accessed.

Shown below are the registers and their addresses:

PIN 3 CS	PIN 6 A <sub>1</sub>	PIN 5 A <sub>0</sub>	PIN 4 RE=σ	PIN 2 WE=σ
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
1	X	X	H1-Z	H1-Z

Each time a command is issued to the 179X, the Busy bit is set and the INTRQ (Interrupt Request) Line is reset. The user has the option of checking the busy bit or use the INTRQ Line to denote command completion. The Busy bit will be reset whenever the 179X is idle and awaiting a new command. The INTRQ Line, once set, can only be reset by a READ of the status register or issuing a new command. The MR (Master Reset) Line does not affect INTRQ.

The A<sub>0</sub>, A<sub>1</sub> Lines used for register selections can be configured at the CPU in a variety of ways. These lines may actually tie to CPU address lines, in which case the 179X will be memory-mapped and addressed like RAM. They may also be used under Program Control by tying to a port device such as the 8255, 6820, etc. As a diagnostic tool when checking out the CPU interface, the Track and Sector registers should respond like "RAM" when the 179X is idle (Busy = INTRQ = 0).

Because of internal synchronization cycles, certain time delays must be introduced when operating under Programmed I/O. The worst case delays are:

OPERATION	NEXT OPERATION	DELAY REQ'D
WRITE TO COMMAND REG	READ STATUS REGISTER	MFM = 14μs* FM = 28μs*
WRITE TO ANY REGISTER	READ FROM A DIFFERENT REG	NO DELAY

\*NOTE: Times Double when CLK = 1MHz (5¼" drive)

Other CPU interface lines are CLK, MR and DDEN. The CLK line should be 2MHz (8" drive) or 1MHz (5¼" drive) with a 50% duty cycle. Accuracy should be ±1% (crystal source) since all internal timing, including stepping rates, are based upon this clock.

The MR or Master Reset Line should be strobed a minimum of 50 microseconds upon each power-on condition. This line clears and initializes all internal registers and issues a restore command (Hex '03') on the rising edge. A quicker stepping rate can be written to the command register after a MR, in which case the remaining steps will occur at the faster programmed rate. The 179X will issue a maximum of 255 stepping pulses in an attempt to expect the TROO line to go active low. This line should be connected to the drive's TROO sensor.

The DDEN line causes selection of either single density (DDEN = 1) or double density operation. DDEN should not be switched during a read or write operation.

### FLOPPY DISK INTERFACE

The Floppy Disk Interface can be divided into three sections: Motor Control, Write Signals and Read Signals. All of these lines are capable of driving one TTL load and not compatible for direct connection to the drive. Most drives require an open-collector TTL interface with high current drive capability. This must be done on all outputs from the 179X. Inputs to the 179X may be buffered or tied to the Drives outputs, providing the appropriate resistor termination networks are used. Undershoot should not exceed -0.3 volts, while integrity of  $V_{IH}$  and  $V_{OH}$  levels should be kept within spec.

### MOTOR CONTROL

Motor Control is accomplished by the STEP and DIRC Lines. The STEP Line issues stepping pulses with a period defined by the rate field in all Type I commands. The DIRC Line defines the direction of steps (DIRC = 1 STEP IN/DIRC = 0 STEP OUT).

Other Control Lines include the IP or Index Pulse. This Line is tied to the drives' Index L.E.D. sensor and makes an active transition for each revolution of the diskette. The TROO Line is another L.E.D. sensor that informs the 179X that the stepper motor is at its furthest position, over Track 00. The READY Line can be used for a number of functions, such as sensing "door open", Drive motor on, etc. Most drives provide a programmable READY Signal selected by option jumpers on the drive. The 179X will look at the ready signal prior to executing READ/WRITE commands. READY is not inspected during any Type I commands. All Type I commands will execute regardless of the Logic Level on this Line.

### WRITE SIGNALS

Writing of data is accomplished by the use of the WD, WG, WF, TG43, EARLY and LATE Lines. The WG or Write Gate Line is used to enable write current at the drive's R/W head. It is made active prior to writing data on the disk. The WF or WRITE FAULT Line is used to inform the 179X of a failure in drive electronics. This signal is multiplexed with the VFOE Line and must be logically separated if required. Figure 5 illustrates three methods of demultiplexing.

The TG43 or "TRACK GREATER than 43" Line is used to decrease the Write current on the inner tracks, where bit densities are the highest. If not required on the drive, TG43 may be left open.

### WRITE PRECOMPENSATION

The 179X provides three signals for double density Write Precompensation use. These signals are WRITE DATA, EARLY and LATE. When using single density drives (either 8" or 5 1/4"), Write Precompensation is not necessary and the WRITE DATA line is generally TTL Buffered and sent directly to the drive. In this mode, EARLY and LATE are left open.

For double density use, Write Precompensation is a function of the drive. Some manufacturers recommend Precompensating the 5 1/4" drive, while others do not. With the 8" drive, Precompensation may be specified from TRACK 43 on, or in most cases, all TRACKS. If the recommended Precompensation is not specified,

check with the manufacturer for the proper configuration required.

The amount of Precompensation time also varies. A typical value will usually be specified from 100-300ns. Regardless of the parameters used, Write Precompensation must be done external to the 179X. When DDEN is tied low, EARLY or LATE will be activated at least 125ns. before and after the Write Data pulse. An Algorithm internal the 179X decides whether to raise EARLY or LATE, depending upon the previous bit pattern sent. As an example, suppose the recommended Precomp value has been specified at 150ns. The following action should be taken:

EARLY	LATE	ACTION TAKEN
0	0	delay WD by 150ns (nominal)
0	1	delay WD by 300ns (2X value)
1	0	do not delay WD

There are two methods of performing Write Precompensation:

- 1) External Delay elements
- 2) Digitally

Shown in Figure 6 is a Precomp circuit using the Western Digital 2143 clock generator as the delay element. The WD pulse from the 179X creates a strobe to the 2143, causing subsequent output pulses on the  $\phi 1$ ,  $\phi 2$  and  $\phi 3$  signals. The 5K Precomp adjust sets the desired Precomp value. Depending upon the condition of EARLY and LATE,  $\phi 1$  will be used for EARLY,  $\phi 2$  for nominal (EARLY = LATE = 0), and  $\phi 3$  for LATE. The use of "one-shots" or delay line in a Write Precompensation scheme offers the user the ability to vary the Precomp value. The  $\phi 4$  output resets the 74LS175 Latch in anticipation of the next WD pulse. Figure 7 shows the WD-EARLY/LATE relationship, while Figure 8 shows the timing of this write Precomp scheme.

Another method of Precomp is to perform the function digitally. Figure 9 illustrates a relationship between the WD pulse and the CLK pin, allowing a digital Precomp scheme. Figure 10 shows such a scheme with a preset Write Precompensation value of 250ns. The synchronous counter is used to generate 2MHz and 4MHz clock signals. The 2MHz clock is sent to the CLK input of the 179X and the 4MHz is used by the 4-bit shift register. When a WD pulse is not present, the 4MHz clock is shifting "ones" through the shift register and maintaining  $Q_0$  at a zero level. When a WD pulse is present, a zero is loaded at either A, B, or C depending upon the states of LATE, EN PRECOMP and EARLY. The zero is then shifted by the 4MHz clock until it reaches the  $Q_0$  output. The number of shift operations determines whether the WRITE DATA pulse is written early, nominal or late. If both FM and MFM operations is a system requirement, the output of this circuit should be disabled and the WD pulse should be sent directly to the drive.

**DATA SEPARATION**

The 179X has two inputs (RAW READ & RCLK) and one output (VFOE) for use by an external data separator. The RAW READ input must present clock and data pulses to the 179X, while the RCLK input provides a "window" or strobe signal to clock each RAW READ pulse into the device. An ideal Data Separator would have the leading edge of the RAW READ pulse occur in the exact center of the RCLK strobe.

Motor Speed Variation, Bit shifts and read amplifier recovery circuits all cause the RAW READ pulses to drift away from their nominal positions. As this occurs, the RAW READ pulses will shift left or right with respect to RCLK. Eventually, a pulse will make its transition outside of its RCLK window, causing either a CRC error or a Record-not-Found error at the 179X.

A Phase-Lock-Loop circuit is one method of achieving synchronization between the RCLK and RAW READ signals. As RAW READ pulses are fed to the PLL, minor adjustments of the free-running RCLK frequency can be made. If pulses are occurring too far apart, the RCLK frequency is *decreased* to keep synchronization. If pulses begin to occur closer together, RCLK is *increased* until this new higher frequency is achieved. In normal read operations, RCLK will be constantly adjusted in an attempt to match the incoming RAW READ frequency.

Another method of Data Separation is the Counter-Separator technique. The RCLK signal is again free-running at a nominal rate, until a RAW READ pulse occurs. The Separator then denotes the position of the pulse with respect to RCLK (by the counter value), and counts down to increase or decrease the current RCLK window. The next RCLK window will occur at a nominal rate and will continue to run at this frequency until another RAW READ pulse adjusts RCLK, but only the present window is adjusted.

Both PPL and Counter/Separator are acceptable methods of Data Separation. The PPL has the highest reliability because of its "tracking" capability and is recommended for 8" double density designs.

As a final note, the term "Data Separator" may be misleading, since the physical separation of clock and data bits are not actually performed. This term is used throughout the industry, and can better be described as a "Data Recovery Circuit" rather than a Data Separator.

The VFOE signal is an output from the 179X that signifies the head has been loaded and valid data pulses are appearing on the RAW READ line. It can be used to enable the Data Separator and to insure clean RCLK transitions to the 179X. Since some drives will output random pulses when the head is disengaged, VFOE can prevent an erratic RCLK signal during this time. If the Data Separator requires synchronization during a known pattern of one's or zero's, then RG (READ GATE) can be used. The RG signal will go active when the 179X is currently over a field of zeros or ones. RG is not available on the 1795/1797 devices, since this signal was replaced with the SSO (Side Select Output) Line.

Shown in Figure 11 is a 2½ IC Counter/Separator. The 74LS193 free runs at a frequency determined by the CRYCLK input. When a RAW READ pulse occurs, the counter is loaded with a starting count of '5'. When the RAW READ Line returns to a Logic 1, the counter counts down to zero and again free runs. The 74LS74 insures a 50% duty cycle to the 179X and performs a divide-by-two of the Q<sub>B</sub> output.

Figure 12 illustrates another Counter/Separator utilizing a PROM as the count generator. Depending upon the RAW READ phase relationship to RCLK, the PROM is addressed and its data output is used as the counter value. A 16MHz clock is required for 8" double density, while an 8MHz clock can be used for single density.

Figure 13 shows a Phase-Lock-Loop data recovery circuit. The phase detector (U2, Figure 2) compares the phase of the SHAPED DATA pulse to the phase of VFO CLK ÷ 2. If VFO CLK ÷ 2 is lagging the SHAPED DATA pulse an output pulse on #9, U2 is generated. The filter/amplifier converts this pulse into a DC signal which increases the frequency of the VCO. If, correspondingly, CLK ÷ 2 is leading the SHAPED DATA pulse, an output pulse on #5, U2 is generated. This pulse is converted into a DC signal which decreases the frequency of the VCO. These two actions cause the VCO to track the frequency of the incoming READ DATA pulses. This correction process to keep the two signals in phase is constantly occurring because of spindle speed variation and circuit parameter variations.

The operating specifications for this circuit are as follows:

Free Running Frequency	2MHz
Capture Range	± 15%
Lock Up Time	50 microsec. "1111" or "0000" Pattern
	100 Microsec "1010" Pattern

The RAW READ pulses are generated from the falling edge of the SHAPED DATA pulses. The pulses are also reshaped to meet the 179X requirements. VFO CLK ÷ 2 OR 4 is divided by 2 once again to obtain VFO CLK OUT whose frequency is that required by the 179X RCLK input. RCLK must be controlled by VFOE so VFOE is sampled on each rising edge of VFO CLK OUT. When VFOE goes active EN RCLK goes active in synchronization with VFO CLK OUT preventing any glitches on the RCLK output. When VFOE goes inactive EN RCLK goes inactive in synchronization with VFO CLK OUT, again preventing any glitches on the RCLK output.

Figure 14 illustrates a PPL data recovery circuit using the Western Digital 1691 Floppy Support device. Both data recovery and Write Precomp Logic is contained within the 1691, allowing low chip count and PLL reliability. The 74S124 supplies the free-running VCO output. The PUMP UP and PUMP DOWN signals from the 1691 are used to control the 74S124's frequency.

### COMMAND USAGE

Whenever a command is successfully or unsuccessfully completed, the busy bit of the status register is reset and the INTRQ line is forced high. Command termination may be detected either way. The INTRQ can be tied to the host processor's interrupt with an appropriate service routine to terminate commands. The busy bit may be monitored with a user program and will achieve the same results through software. Performing both an INTRQ and a busy bit check is not recommended because a read of the status register to determine the condition of the busy bit will reset the INTRQ line. This can cause an INTRQ from not occurring.

### RESTORE COMMAND

On some disk drives, it is possible to position the R/W head outward past Track 00 and prevent the TROO line from going low unless a STEP IN is first performed. If this condition exists in the drive used, the RESTORE command will never detect a TROO. Issuing several STEP IN pulses before a RESTORE command will remedy this situation. The RESTORE and all other Type I commands will execute even though the READY bit indicates the drive is not ready (NOT READY = 1).

### READ TRACK COMMAND

The READ TRACK command can be used to manually inspect data on a hard copy printout. Gaps, address marks and all data are brought in to the data register during this command. The READ TRACK command may be used to inspect diskettes for valid formatting and data fields as well as address marks. Since the 179X does not synchronize clock and data until the Index Address Mark is detected, data previous to this ID mark will not be valid. READ GATE (RG) is not actuated during this command.

### READ ADDRESS COMMAND

In systems that use either multiple drives or sides, the read address command can be used to tell the host processor which drive or side is selected. The current position of the R/W head is also denoted in the six bytes of data that are sent to the computer.

TRACK	SIDE	SECTOR	CRS LENGTH	CRC 1	CRC 2
-------	------	--------	------------	-------	-------

The READ ADDRESS command as well as all other Type II and Type III commands will not execute if the READY line is inactive (READY = 0). Instead, an interrupt will be generated and the NOT READY status bit will be set to a 1.

### FORCED INTERRUPT COMMAND

The Forced Interrupt command is generally used to terminate a multiple sector command or to insure Type I status in the status register. The lower four bits of the command determine the conditional interrupt as follows:

1 <sub>0</sub>	=	NOT-READY TO READY TRANSITION
1 <sub>1</sub>	=	READY TO NOT-READY TRANSITION
1 <sub>2</sub>	=	EVERY INDEX PULSE
1 <sub>3</sub>	=	IMMEDIATE INTERRUPT

Regardless of the conditional interrupt set, any command that is currently being executed when the Forced Interrupt command is loaded will immediately be terminated and the busy bit will be reset indicating an idle condition.

Then, when the condition for interrupt is met, the INTRQ line will go high signifying that the condition specified has occurred.

The conditional interrupt is enabled when the corresponding bit positions of the command (1<sub>3</sub> - 1<sub>0</sub>) are set to a 1. If 1<sub>3</sub> - 1<sub>0</sub> are all set to zero, no interrupt will occur, but any command presently under execution will be immediately terminated upon receipt of the Force Interrupt command (HEX D0).

As usual, to clear the interrupt a read of the status register or a write to the command register is required. The exception is when using the immediate interrupt condition (1<sub>3</sub> = 1). If this command is loaded into the command register, 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; another forced interrupt command with 1<sub>3</sub> - 1<sub>0</sub> = 0 must be loaded into the command register in order to reset the INTRQ from this condition.

More than one condition may be set at a time. If for example, the READY TO NOT-READY condition (1<sub>1</sub> = 1) and the Every Index Pulse (1<sub>2</sub> = 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.

### DATA RECOVERY

Occasionally, the R/W head of the disk drive may get "off track", and dust or dirt may get trapped on the media. Both of these conditions will cause a RECORD NOT FOUND and/or a CRC error to occur. This "soft error" can usually be recovered by the following procedure:

1. Issue the command again
2. Unload and load the head and repeat step
3. Issue a restore, seek the track, and repeat step 1

If RNF or CRC errors are still occurring after trying these methods, a "hard error" may exist. This is usually caused by improper disk handling, exposure to high magnetic fields, etc. and generally results in destroying portions or tracks of the diskette.

FIGURE 1 DEVICE CHARACTERISTICS

DEVICE	SNGL DENSITY	DBLE DENSITY	INVERTED BUS	TRUE BUS	DOUBLE-SIDED
1791	X	X	X		
1792	X		X		
1793	X	X		X	
1794	X			X	
1795	X	X	X		X
1797	X	X		X	X

FIGURE 2 STORAGE CAPACITIES

SIZE	DENSITY	SIDES	UNFORMATTED CAPACITY (NOMINAL)		BYTE TRANSFER TIME	FORMATTED CAPACITY	
			PER TRACK	PER DISK		PER TRACK	PER DISK
5¼"	SINGLE	1	3125	109,375*	64µs	2304**	80,640
5¼"	DOUBLE	1	6250	218,750	32µs	4608***	161,280
5¼"	SINGLE	2	3125	218,750	64µs	2304	161,280
5¼"	DOUBLE	2	6250	437,500	32µs	4608	322,560
8"	SINGLE	1	5208	401,016	32µs	3328	256,256
8"	DOUBLE	1	10,416	802,032	16µs	6656	512,512
8"	SINGLE	2	5208	802,032	32µs	3328	512,512
8"	DOUBLE	2	10,416	1,604,064	16µs	6656	1,025,024

\*Based on 35 Tracks/Side

\*\*Based on 18 Sectors/Track (128 byte/sec)

\*\*\*Based on 18 Sectors/Track (256 bytes/sec)

11.5 Appendix E - Z80A CPU .....

# Z8400 Z80<sup>®</sup> CPU Central Processing Unit



## 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.

Z80 CPU

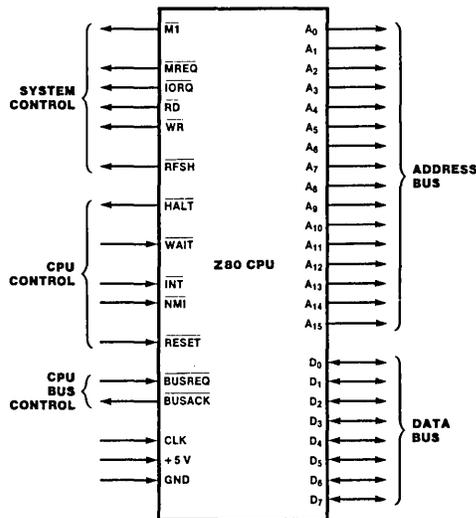


Figure 1. Pin Functions

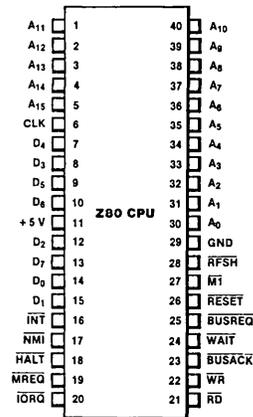


Figure 2. Pin Assignments

**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.

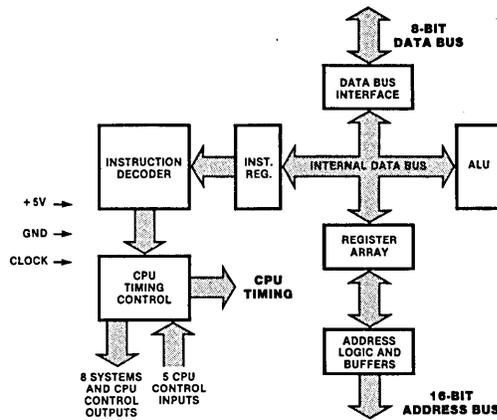


Figure 3. Z80 CPU Block Diagram

**Z80 Micro-processor 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 interrupt mode at any particular time. Table 1 provides further information on these registers.

Z80 CPU

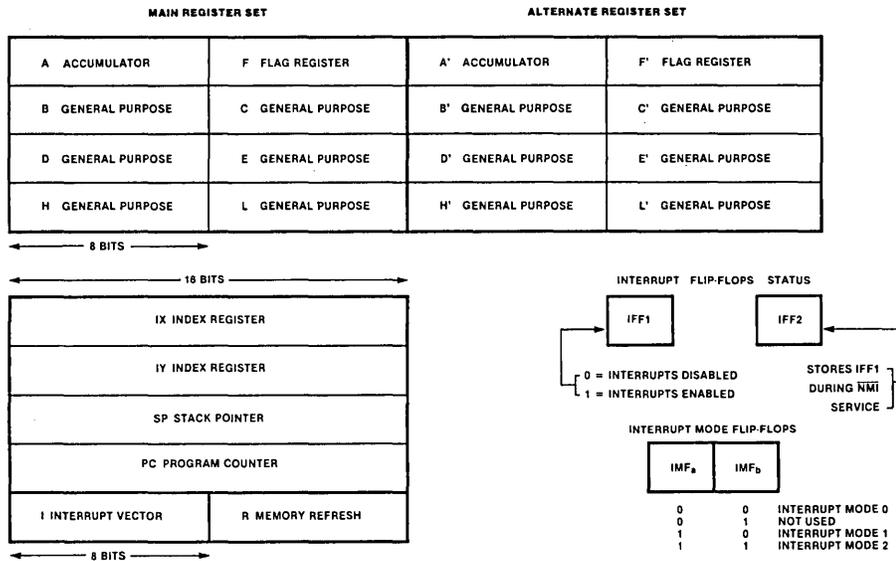


Figure 4. CPU Registers

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 <sub>1</sub> -IFF <sub>2</sub>	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:  $\overline{\text{NMI}}$  and  $\overline{\text{INT}}$ . The  $\overline{\text{NMI}}$  is a non-maskable interrupt and has the highest priority.  $\overline{\text{INT}}$  is a lower priority interrupt since it requires that interrupts be enabled in software in order to operate. Either  $\overline{\text{NMI}}$  or  $\overline{\text{INT}}$  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,  $\overline{\text{INT}}$ , 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  $\overline{\text{NMI}}$  and  $\overline{\text{INT}}$  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 (NMI).** The non-maskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. NMI is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shut-down after power failure has been detected. After recognition of the NMI signal (providing  $\overline{\text{BUSREQ}}$  is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

**Maskable Interrupt (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  $\overline{\text{BUSREQ}}$  is not active) a special interrupt processing cycle begins. This is a special fetch (M1) cycle in which  $\overline{\text{IORQ}}$  becomes active rather than  $\overline{\text{MREQ}}$ , as in a normal M1 cycle. In addition, this special M1 cycle is automatically extended by two 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 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 hardwired to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

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

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

**Interrupt Enable/Disable Operation.** Two flip-flops, IFF<sub>1</sub> and IFF<sub>2</sub>, 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 <sub>1</sub>	IFF <sub>2</sub>	Comments
CPU Reset	0	0	Maskable interrupt INT disabled
DI instruction execution	0	0	Maskable interrupt INT disabled
EI instruction execution	1	1	Maskable interrupt INT enabled
LD A,I instruction execution	•	•	IFF <sub>2</sub> — Parity flag
LD A,R instruction execution	•	•	IFF <sub>2</sub> — Parity flag
Accept $\overline{\text{NMI}}$	0	IFF <sub>1</sub>	IFF <sub>1</sub> — IFF <sub>2</sub> (Maskable interrupt INT disabled)
RETN instruction execution	IFF <sub>2</sub>	•	IFF <sub>2</sub> — IFF <sub>1</sub> at completion of an 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	Flags			Opcode		No. of Bytes	No. of M Cycles	No. of T States	Comments		
				H	P/V	N	C	76 543 210					Hex	
LD r, r'	r - r'	.	.	X	.	X	.	.	01 r r'	1	1	4	r, r' Reg.	
LD r, n	r - n	.	.	X	.	X	.	.	00 r 110	2	2	7	000 B	
									- n -				001 C	
LD r, (HL)	r - (HL)	.	.	X	.	X	.	.	01 r 110	1	2	7	010 D	
LD r, (IX+d)	r - (IX+d)	.	.	X	.	X	.	.	11 011 101	DD	3	5	19	011 E
									01 r 101				100 H	
									- d -				101 L	
LD r, (IY+d)	r - (IY+d)	.	.	X	.	X	.	.	11 111 101	FD	3	5	19	111 A
									01 r 110					
									- d -					
LD (HL), r	(HL) - r	.	.	X	.	X	.	.	01 110 r	1	2	7		
LD (IX+d), r	(IX+d) - r	.	.	X	.	X	.	.	11 011 101	DD	3	5	19	
									01 110 r					
									- d -					
LD (IY+d), r	(IY+d) - r	.	.	X	.	X	.	.	11 111 101	FD	3	5	19	
									01 110 r					
									- d -					
LD (HL), n	(HL) - n	.	.	X	.	X	.	.	00 110 110	36	2	3	10	
									- n -					
LD (IX+d), n	(IX+d) - n	.	.	X	.	X	.	.	11 011 101	DD	4	5	19	
									00 110 110	36				
									- d -					
									- n -					
LD (IY+d), n	(IY+d) - n	.	.	X	.	X	.	.	11 111 101	FD	4	5	19	
									00 110 110	36				
									- d -					
									- n -					
LD A, (BC)	A - (BC)	.	.	X	.	X	.	.	00 001 010	0A	1	2	7	
LD A, (DE)	A - (DE)	.	.	X	.	X	.	.	00 011 010	1A	1	2	7	
LD A, (nn)	A - (nn)	.	.	X	.	X	.	.	00 111 010	3A	3	4	13	
									- n -					
									- n -					
LD (BC), A	(BC) - A	.	.	X	.	X	.	.	00 000 010	02	1	2	7	
LD (DE), A	(DE) - A	.	.	X	.	X	.	.	00 010 010	12	1	2	7	
LD (nn), A	(nn) - A	.	.	X	.	X	.	.	00 110 010	32	3	4	13	
									- n -					
									- n -					
LD A, I	A - I	1	1	X	0	X	IFF	0	11 101 101	ED	2	2	9	
									01 010 111	57				
LD A, R	A - R	1	1	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	Flags							Opcode 76 543 210 Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N	C	dd					Pair		
LD dd, nn	dd - nn	.	.	X	.	X	.	.	.	00 dd0 001	3	3	10	dd	Pair
		.	.	.	.	.	.	.	.	-- n --				00	BC
		.	.	.	.	.	.	.	.	-- n --				01	DE
LD IX, nn	IX - nn	.	.	X	.	X	.	.	.	11 011 101 DD	4	4	14	10	HL
		.	.	.	.	.	.	.	.	00 100 001 21				11	SP
		.	.	.	.	.	.	.	.	-- n --					
LD IY, nn	IY - nn	.	.	X	.	X	.	.	.	11 111 101 FD	4	4	14		
		.	.	.	.	.	.	.	.	00 100 001 21					
		.	.	.	.	.	.	.	.	-- n --					
LD HL, (nn)	H - (nn+1) L - (nn)	.	.	X	.	X	.	.	.	00 101 010 2A	3	5	16		
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD dd, (nn)	dd <sub>H</sub> - (nn+1) dd <sub>L</sub> - (nn)	.	.	X	.	X	.	.	.	11 101 101 ED	4	6	20		
		.	.	.	.	.	.	.	.	01 dd1 011					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD IX, (nn)	IX <sub>H</sub> - (nn+1) IX <sub>L</sub> - (nn)	.	.	X	.	X	.	.	.	11 011 101 DD	4	6	20		
		.	.	.	.	.	.	.	.	00 101 010 2A					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD IY, (nn)	IY <sub>H</sub> - (nn+1) IY <sub>L</sub> - (nn)	.	.	X	.	X	.	.	.	11 111 101 FD	4	6	20		
		.	.	.	.	.	.	.	.	00 101 010 2A					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD (nn), HL	(nn+1) - H (nn) - L	.	.	X	.	X	.	.	.	00 100 010 22	3	5	16		
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD (nn), dd	(nn+1) - dd <sub>H</sub> (nn) - dd <sub>L</sub>	.	.	X	.	X	.	.	.	11 101 101 ED	4	6	20		
		.	.	.	.	.	.	.	.	01 dd0 011					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD (nn), IX	(nn+1) - IX <sub>H</sub> (nn) - IX <sub>L</sub>	.	.	X	.	X	.	.	.	11 011 101 DD	4	6	20		
		.	.	.	.	.	.	.	.	00 100 010 22					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
LD (nn), IY	(nn+1) - IY <sub>H</sub> (nn) - IY <sub>L</sub>	.	.	X	.	X	.	.	.	11 111 101 FD	4	6	20		
		.	.	.	.	.	.	.	.	00 100 010 22					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
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	2	2	10		
		.	.	.	.	.	.	.	.	11 111 001 F9					
LD SP, IY	SP - IY	.	.	X	.	X	.	.	.	11 111 101 FD	2	2	10		
		.	.	.	.	.	.	.	.	11 111 001 F9					
PUSH qq	(SP-2) - qq <sub>L</sub> (SP-1) - qq <sub>H</sub> SP - SP - 2	.	.	X	.	X	.	.	.	11 qq0 101	1	3	11	qq	Pair
		.	.	.	.	.	.	.	.	-- n --				00	BC
		.	.	.	.	.	.	.	.	-- n --				01	DE
		.	.	.	.	.	.	.	.	-- n --				10	HL
PUSH IX	(SP-2) - IX <sub>L</sub> (SP-1) - IX <sub>H</sub> SP - SP - 2	.	.	X	.	X	.	.	.	11 011 101 DD	2	4	15	11	AF
		.	.	.	.	.	.	.	.	11 100 101 ES					
		.	.	.	.	.	.	.	.	-- n --					
PUSH IY	(SP-2) - IY <sub>L</sub> (SP-1) - IY <sub>H</sub> SP - SP - 2	.	.	X	.	X	.	.	.	11 111 101 FD	2	4	15		
		.	.	.	.	.	.	.	.	11 100 101 ES					
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
POP qq	qq <sub>H</sub> - (SP+1) qq <sub>L</sub> - (SP) SP - SP + 2	.	.	X	.	X	.	.	.	11 qq0 001	1	3	10		
		.	.	.	.	.	.	.	.	-- n --					
		.	.	.	.	.	.	.	.	-- n --					
POP IX	IX <sub>H</sub> - (SP+1) IX <sub>L</sub> - (SP) SP - SP + 2	.	.	X	.	X	.	.	.	11 011 101 DD	2	4	14		
		.	.	.	.	.	.	.	.	11 100 001 E1					
		.	.	.	.	.	.	.	.	-- n --					
POP IY	IY <sub>H</sub> - (SP+1) IY <sub>L</sub> - (SP) SP - SP + 2	.	.	X	.	X	.	.	.	11 111 101 FD	2	4	14		
		.	.	.	.	.	.	.	.	11 100 001 E1					
		.	.	.	.	.	.	.	.	-- n --					

NOTES: dd is any of the register pairs BC, DE, HL, SP.  
 qq is any of the register pairs AF, BC, DE, HL.  
 (PAIR)<sub>H</sub>, (PAIR)<sub>L</sub> refer to high order and low order eight bits of the register pair respectively.  
 e.g., BC<sub>L</sub> = C, AF<sub>H</sub> = A.

**Exchange, Block Transfer, Block Search Groups**

EX DE, HL	DE - HL	.	.	X	.	X	.	.	.	11 101 011 EB	1	1	4		
EX AF, AF'	AF - AF'	.	.	X	.	X	.	.	.	00 001 000 08	1	1	4		
EXX	BC - BC' 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 <sub>H</sub> - (SP+1) IX <sub>L</sub> - (SP)	.	.	X	.	X	.	.	.	11 011 101 DD	2	6	23		
		.	.	.	.	.	.	.	.	11 100 011 E3					
EX (SP), IY	IY <sub>H</sub> - (SP+1) IY <sub>L</sub> - (SP)	.	.	X	.	X	.	.	.	11 111 101 FD	2	6	23		
		.	.	.	.	.	.	.	.	11 100 011 E3					
LDI	(DE) - (HL) DE - DE + 1 HL - HL + 1 BC - BC - 1	.	.	X	0	X	1	0	.	11 101 101 ED	2	4	16		Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
		.	.	.	.	.	.	.	.	10 100 000 A0					
LDIR	(DE) - (HL) DE - DE + 1 HL - HL + 1 BC - BC - 1 Repeat until BC = 0	.	.	X	0	X	0	0	.	11 101 101 ED	2	5	21		If BC ≠ 0
		.	.	.	.	.	.	.	.	10 110 000 B0	2	4	16		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	Flags						Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments
		S	Z	H	P/V	N	C	76	543	210	Hex				
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	X	0	X	1	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	
CPI	A ← (HL) HL ← HL+1 BC ← BC-1	1	1	X	1	X	1	1	•	11 101 101 ED 10 100 001 A1	2	4	16		
CPIR	A ← (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	1	1	X	1	X	1	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)	
CPD	A ← (HL) HL ← HL-1 BC ← BC-1	1	1	X	1	X	1	1	•	11 101 101 ED 10 101 001 A9	2	4	16		
CPDR	A ← (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	1	1	X	1	X	1	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**

ADD A, r	A ← A + r	1	1	X	1	X	V	0	1	10 <u>000</u> r	1	1	4	r Reg.
ADD A, n	A ← A + n	1	1	X	1	X	V	0	1	11 <u>000</u> 110 - n -	2	2	7	000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, (HL)	A ← A + (HL)	1	1	X	1	X	V	0	1	10 <u>000</u> 110	1	2	7	
ADD A, (IX+d)	A ← A + (IX+d)	1	1	X	1	X	V	0	1	11 011 101 DD 10 <u>000</u> 110 - d -	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	1	1	X	1	X	V	0	1	11 111 101 FD 10 <u>000</u> 110 - d -	3	5	19	
ADC A, s	A ← A + s + CY	1	1	X	1	X	V	0	1	<u>001</u>				s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the <u>000</u> in the ADD set above.
SUB s	A ← A - s	1	1	X	1	X	V	1	1	<u>010</u>				
SBC A, s	A ← A - s - CY	1	1	X	1	X	V	1	1	<u>011</u>				
AND s	A ← A ∧ s	1	1	X	1	X	P	0	0	<u>100</u>				
OR s	A ← A ∨ s	1	1	X	0	X	P	0	0	<u>110</u>				
XOR s	A ← A ⊕ s	1	1	X	0	X	P	0	0	<u>101</u>				
CP s	A ← s	1	1	X	1	X	V	1	1	<u>111</u>				
INC r	r ← r + 1	1	1	X	1	X	V	0	•	00 r <u>100</u>	1	1	4	
INC (HL)	(HL) ← (HL)+1	1	1	X	1	X	V	0	•	00 110 <u>100</u>	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	1	1	X	1	X	V	0	•	11 011 101 DD 00 110 <u>100</u> - d -	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	1	1	X	1	X	V	0	•	11 111 101 FD 00 110 <u>100</u> - d -	3	6	23	
DEC m	m ← m - 1	1	1	X	1	X	V	1	•	- <u>101</u>				m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace <u>100</u> with <u>101</u> in opcode.

General-Purpose Arithmetic and CPU Control Groups	Mnemonic	Symbolic Operation	Flags								Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments
			S	Z	H	P/V	N	C	78	543	210	Hex						
DAA		Converts acc. content into packed BCD following add or subtract with packed BCD operands.	1	1	X	1	X	P	*	1	00	100	111	27	1	1	4	Decimal adjust accumulator.
CPL		$A - \bar{A}$	*	*	X	1	X	*	1	*	00	101	111	2F	1	1	4	Complement accumulator (one's complement).
NEG		$A - 0 - A$	1	1	X	1	X	V	1	1	11	101	101	ED	2	2	8	Negate acc. (two's complement).
CCF		$CY - \bar{CY}$	*	*	X	X	X	*	0	1	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	2	2	8	
IM 1		Set interrupt mode 1	*	*	X	*	X	*	*	*	01	000	110	46				
IM 2		Set interrupt mode 2	*	*	X	*	X	*	*	*	11	101	101	ED	2	2	8	
											01	011	110	5E				

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

16-Bit Arithmetic Group	Mnemonic	Symbolic Operation	Flags								Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments
			S	Z	H	P/V	N	C	78	543	210	Hex						
ADD HL, ss		$HL - HL + ss$	*	*	X	X	X	*	0	1	00	ss1	001		1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss		$HL - HL + ss + CY$	1	1	X	X	X	V	0	1	11	101	101	ED	2	4	15	
											01	ss1	010					
SBC HL, ss		$HL - HL - ss - CY$	1	1	X	X	X	V	1	1	11	101	101	ED	2	4	15	
											01	ss0	010					
ADD IX, pp		$IX - IX + pp$	*	*	X	X	X	*	0	1	11	011	101	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
											01	pp1	001					
ADD IY, rr		$IY - IY + rr$	*	*	X	X	X	*	0	1	11	111	101	FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
											00	rr1	001					
INC ss		$ss - ss + 1$	*	*	X	*	X	*	*	*	00	ss0	011		1	1	6	
INC IX		$IX - IX + 1$	*	*	X	*	X	*	*	*	11	011	101	DD	2	2	10	
INC IY		$IY - IY + 1$	*	*	X	*	X	*	*	*	11	111	101	FD	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	2	2	10	
DEC IY		$IY - IY - 1$	*	*	X	*	X	*	*	*	00	101	011	2B				
											11	111	101	FD	2	2	10	
											00	101	011	2B				

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

Rotate and Shift Group	Mnemonic	Diagram	Flags								Opcode				No. of Bytes	No. of M Cycles	No. of T States	Comments
			S	Z	H	P/V	N	C	78	543	210	Hex						
RLCA			*	*	X	0	X	*	0	1	00	000	111	07	1	1	4	Rotate left circular accumulator.
RLA			*	*	X	0	X	*	0	1	00	010	111	17	1	1	4	Rotate left accumulator.
RRCA			*	*	X	0	X	*	0	1	00	001	111	0F	1	1	4	Rotate right circular accumulator.
RRA			*	*	X	0	X	*	0	1	00	011	111	1F	1	1	4	Rotate right accumulator.
RLC r			1	1	X	0	X	P	0	1	11	001	011	CB	2	2	8	Rotate left circular register r.
RLC (HL)			1	1	X	0	X	P	0	1	11	001	011	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)			1	1	X	0	X	P	0	1	11	011	101	DD	4	6	23	
		$r, (HL), (IX+d), (IY+d)$									11	001	011	CB				
											00	000	110					
RLC (IY+d)			1	1	X	0	X	P	0	1	11	111	101	FD	4	6	23	
											11	001	011	CB				
											00	000	110					
RL m			1	1	X	0	X	P	0	1								
		$m = r, (HL), (IX+d), (IY+d)$																
RRC m			1	1	X	0	X	P	0	1								
		$m = r, (HL), (IX+d), (IY+d)$																
																		Instruction format and states are as shown for RLC's. To form new opcode replace 000 or RLC's with shown code.

**Rotate and Shift Group**  
(Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H P/V N C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RR m	 $m = r.(HL).(IX+d).(IY+d)$	1	1	X 0 X P 0 1	011					
SLA m	 $m = r.(HL).(IX+d).(IY+d)$	1	1	X 0 X P 0 1	100					
SRA m	 $m = r.(HL).(IX+d).(IY+d)$	1	1	X 0 X P 0 1	101					
SRL m	 $m = r.(HL).(IX+d).(IY+d)$	1	1	X 0 X P 0 1	111					
RLD	 A (HL)	1	1	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	 A (HL)	1	1	X 0 X P 0 *	11 101 101 01 100 111	ED 6F	2	5	18	The content of the upper half of the accumulator is unaffected.

**Bit Set, Reset and Test Group**

BIT b, r	$Z - \bar{r}_b$	X	1	X 1 X X 0 *	11 001 011 01 b r	CB	2	2	8	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
BIT b, (HL)	$Z - (\overline{HL})_b$	X	1	X 1 X X 0 *	11 001 011 01 b 110	CB	2	3	12	
BIT b, (IX+d) <sub>b</sub>	$Z - (\overline{IX+d})_b$	X	1	X 1 X X 0 *	11 011 101 11 001 011 - d - 01 b 110	DD CB	4	5	20	101 L 111 A b Bit Tested
BIT b, (IY+d) <sub>b</sub>	$Z - (\overline{IY+d})_b$	X	1	X 1 X X 0 *	11 111 101 11 001 011 - d - 01 b 110	FD CB	4	5	20	000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
SET b, r	$r_b - 1$	.	.	X . X . . .	11 001 011 11 b r	CB	2	2	8	
SET b, (HL)	$(HL)_b - 1$	.	.	X . X . . .	11 001 011 11 b 110	CB	2	4	15	
SET b, (IX+d)	$(IX+d)_b - 1$	.	.	X . X . . .	11 011 101 11 001 011 - d - 11 b 110	DD CB	4	6	23	
SET b, (IY+d)	$(IY+d)_b - 1$	.	.	X . X . . .	11 111 101 11 001 011 - d - 11 b 110	FD CB	4	6	23	
RES b, m	$m_b - 0$ $m = r.(HL),$ $(IX+d),$ $(IY+d)$	.	.	X . X . . .	11 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) or location m.

**Jump Group**

JP nn	PC - nn	.	.	X . X . . .	11 000 011 - n - - n -	C3	3	3	10	
JP cc, nn	If condition cc is true PC - nn, otherwise continue	.	.	X . X . . .	11 cc 010 - n - - n -		3	3	10	cc Condition 000 NZ non-zero 001 Z zero 010 NC non-carry 011 C carry 100 PO parity odd 101 PE parity even 110 P sign positive 111 M sign negative
JR e	PC - PC+e	.	.	X . X . . .	00 011 000 - e-2 - - e-2 -		2	3	12	
JR C, e	If C = 0, continue If C = 1, PC - PC+e	.	.	X . X . . .	00 111 000 - e-2 - - e-2 -		2	2	7	If condition not met.
JR NC, e	If C = 1, continue If C = 0, PC - PC+e	.	.	X . X . . .	00 110 000 - e-2 - - e-2 -		2	2	7	If condition not met.
JP Z, e	If Z = 0, continue If Z = 1, PC - PC+e	.	.	X . X . . .	00 101 000 - e-2 - - e-2 -		2	2	7	If condition not met.
JR NZ, e	If Z = 1, continue If Z = 0, PC - PC+e	.	.	X . X . . .	00 100 000 - e-2 - - e-2 -		2	2	7	If condition not 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	Flags						Opcode			No. of Bytes	No. of M Cycles	No. of T States	Comments		
		S	Z	H	P/V	N	C	78	543	210					Hex	
JP (IV)	PC - IV	.	.	X	.	X	.	.	.	.	11 111 101	FD	2	2	8	
DJNZ, e	B - B - 1	.	.	X	.	X	.	.	.	.	00 101 001	E9	2	2	8	If B = 0.
	If B = 0, continue If B ≠ 0, PC - PC + e															

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**

CALL nn	(SP - 1) - PC <sub>H</sub> (SP - 2) - PC <sub>L</sub> PC - nn	.	.	X	.	X	.	.	.	.	11 001 101	CD	3	5	17				
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	.	.	X	.	X	.	.	.	.	11 cc 100		3	3	10	If cc is false.			
													3	5	17	If cc is true.			
RET	PC <sub>L</sub> - (SP) PC <sub>H</sub> - (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	2	4	14				
RETN <sup>1</sup>	Return from non-maskable interrupt	.	.	X	.	X	.	.	.	.	01 001 101	4D	2	4	14				
																01 101 101	ED		
RST p	(SP - 1) - PC <sub>H</sub> (SP - 2) - PC <sub>L</sub> PC <sub>H</sub> - 0 PC <sub>L</sub> - p	.	.	X	.	X	.	.	.	.	11 t 111		1	3	11				
																	t		
																		p	
																		000 00H	
																		001 08H	
																			010 10H
																			011 18H
																			100 20H
																			101 28H
																			110 30H
																			111 38H

NOTE: <sup>1</sup>RETN loads IFF<sub>2</sub> - IFF<sub>1</sub>

**Input and Output Group**

IN A, (n)	A - (n)	.	.	X	.	X	.	.	.	.	11 011 011	DB	2	3	11	n to A <sub>0</sub> - A <sub>7</sub> Acc. to A <sub>8</sub> - A <sub>15</sub>	
IN r, (C)	r - (C) if r = 110 only the flags will be effected	1	1	X	1	X	P	0	.	.	11 101 101	ED	2	3	12	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												01 r 000					
INI	(HL) - (C) B - B - 1 HL - HL + 1	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												10 100 010					A2
INIR	(HL) - (C) B - B - 1 HL - HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												10 110 010					B2
IND	(HL) - (C) B - B - 1 HL - HL - 1	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												10 101 010					AA
INDR	(HL) - (C) B - B - 1 HL - HL - 1 Repeat until B = 0	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												10 111 010					BA
OUT (n), A	(n) - A	.	.	X	.	X	.	.	.	.	11 010 011	D3	2	3	11	n to A <sub>0</sub> - A <sub>7</sub> Acc. to A <sub>8</sub> - A <sub>15</sub>	
OUT (C), r	(C) - r	.	.	X	.	X	.	.	.	.	11 101 101	ED	2	3	12	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												01 r 001					
OUTI	(C) - (HL) B - B - 1 HL - HL + 1	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												10 100 011					A3
OTIR	(C) - (HL) B - B - 1 HL - HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	5 (If B ≠ 0) 4 (If B = 0)	21	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	
												10 110 011					B3
OUTD	(C) - (HL) B - B - 1 HL - HL - 1	X	1	X	X	X	X	1	.	.	11 101 101	ED	2	4	16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>	

NOTE: ① 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	Flags H	P/V	N	C	Opcode 78 543 210 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 2	5 (If B≠0) 4 (If B=0)	21 16	C to A <sub>0</sub> - A <sub>7</sub> B to A <sub>8</sub> - A <sub>15</sub>

**Summary of Flag Operation**

Instruction	D <sub>7</sub> S	Z	H	P/V	N	D <sub>0</sub> C	Comments
ADD A, s; ADC A, s	1	1	X	1	X	V 0 1	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	1	1	X	1	X	V 1 1	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	1	1	X	1	X	P 0 0	Logical operations.
OR s; XOR s	1	1	X	0	X	P 0 0	
INC s	1	1	X	1	X	V 0 •	
DEC s	1	1	X	1	X	V 1 •	8-bit decrement.
ADD DD, ss	•	•	X	X	X	• 0 1	16-bit add.
ADC HL, ss	1	1	X	X	X	V 0 1	16-bit add with carry.
SBC HL, ss	1	1	X	X	X	V 1 1	16-bit subtract with carry.
RLA, RLCA, RRA; RRCA	•	•	X	0	X	• 0 1	Rotate accumulator.
RL m; RLC m; RR m;	1	1	X	0	X	P 0 1	Rotate and shift locations.
RRC m; SLA m;							
SRA m; SRL m							
RLD; RRD	1	1	X	0	X	P 0 •	Rotate digit left and right.
DAA	1	1	X	1	X	P • 1	Decimal adjust accumulator.
CPL	•	•	X	1	X	• 1 •	Complement accumulator.
SCF	•	•	X	0	X	• 0 1	Set carry.
CCF	•	•	X	X	X	• 0 1	Complement carry.
IN r (C)	1	1	X	0	X	P 0 •	Input register indirect.
INI, IND, OUTI; OUTD	X	1	X	X	X	X 1 •	Block input and output. Z = 0 if B ≠ 0 otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X 1 •	
LDI; LDD	X	X	X	0	X	1 0 •	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0 0 •	
CFI; CFIR; CPD; CPDR	X	1	X	X	X	1 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	1	1	X	0	X	IFF 0 •	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag.
BIT b, s	X	1	X	1	X	X 0 •	The state of bit b of location s is copied into the Z flag.

**Symbolic Notation**

Symbol	Operation	Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.	1	The flag is affected according to the result of the operation.
Z	Zero flag. Z = 1 if the result of the operation is 0.	•	The flag is unchanged by the operation.
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.	0	The flag is reset by the operation.
H	Half-carry flag. H = 1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.	1	The flag is set by the operation.
N	Add/Subtract flag. N = 1 if the previous operation was subtract.	X	The flag is a "don't care."
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.	V	P/V flag affected according to the overflow result of the operation.
C	Carry/Limit flag. C = 1 if the operation produced a carry from the MSB of the operand or result.	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
<b>A<sub>0</sub>-A<sub>15</sub></b>	<b>Address Bus</b> (output, active High, 3-state). A <sub>0</sub> -A <sub>15</sub> 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.
<b>BUSACK</b>	<b>Bus Acknowledge</b> (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals $\overline{\text{MREQ}}$ , $\overline{\text{IORQ}}$ , $\overline{\text{RD}}$ , and $\overline{\text{WR}}$ have entered their high-impedance states. The external circuitry can now control these lines.
<b>BUSREQ</b>	<b>Bus Request</b> (input, active Low). Bus Request has a higher priority than $\overline{\text{NMI}}$ and is always recognized at the end of the current machine cycle. $\overline{\text{BUSREQ}}$ forces the CPU address bus, data bus, and control signals $\overline{\text{MREQ}}$ , $\overline{\text{IORQ}}$ , $\overline{\text{RD}}$ , and $\overline{\text{WR}}$ to go to a high-impedance state so that other devices can control these lines. $\overline{\text{BUSREQ}}$ is normally wire-ORed and requires an external pullup for these applications. Extended $\overline{\text{BUSREQ}}$ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.
<b>D<sub>0</sub>-D<sub>7</sub></b>	<b>Data Bus</b> (input/output, active High, 3-state). D <sub>0</sub> -D <sub>7</sub> constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.
<b>HALT</b>	<b>Halt State</b> (output, active Low). $\overline{\text{HALT}}$ 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.
<b>INT</b>	<b>Interrupt Request</b> (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. $\overline{\text{INT}}$ is normally wire-ORed and requires an external pullup for these applications.
<b>IORQ</b>	<b>Input/Output Request</b> (output, active Low, 3-state). $\overline{\text{IORQ}}$ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. $\overline{\text{IORQ}}$ is also generated concurrently with $\overline{\text{MI}}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.
<b>MI</b>	<b>Machine Cycle One</b> (output, active Low). $\overline{\text{MI}}$ , together with $\overline{\text{MREQ}}$ , indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. $\overline{\text{MI}}$ , together with $\overline{\text{IORQ}}$ , indicates an interrupt acknowledge cycle.
<b>MREQ</b>	<b>Memory Request</b> (output, active Low, 3-state). $\overline{\text{MREQ}}$ indicates that the address bus holds a valid address for a memory read or memory write operation.
<b>NMI</b>	<b>Non-Maskable Interrupt</b> (input, active Low). $\overline{\text{NMI}}$ has a higher priority than $\overline{\text{INT}}$ . $\overline{\text{NMI}}$ 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.
<b>RD</b>	<b>Memory Read</b> (output, active Low, 3-state). $\overline{\text{RD}}$ 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.
<b>RESET</b>	<b>Reset</b> (input, active Low). $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ must be active for a minimum of three full clock cycles before the reset operation is complete.
<b>RFSH</b>	<b>Refresh</b> (output, active Low). $\overline{\text{RFSH}}$ , together with $\overline{\text{MREQ}}$ , 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.
<b>WAIT</b>	<b>Wait</b> (input, active Low). $\overline{\text{WAIT}}$ 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 $\overline{\text{WAIT}}$ periods can prevent the CPU from refreshing dynamic memory properly.
<b>WR</b>	<b>Memory Write</b> (output, active Low, 3-state). $\overline{\text{WR}}$ 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:

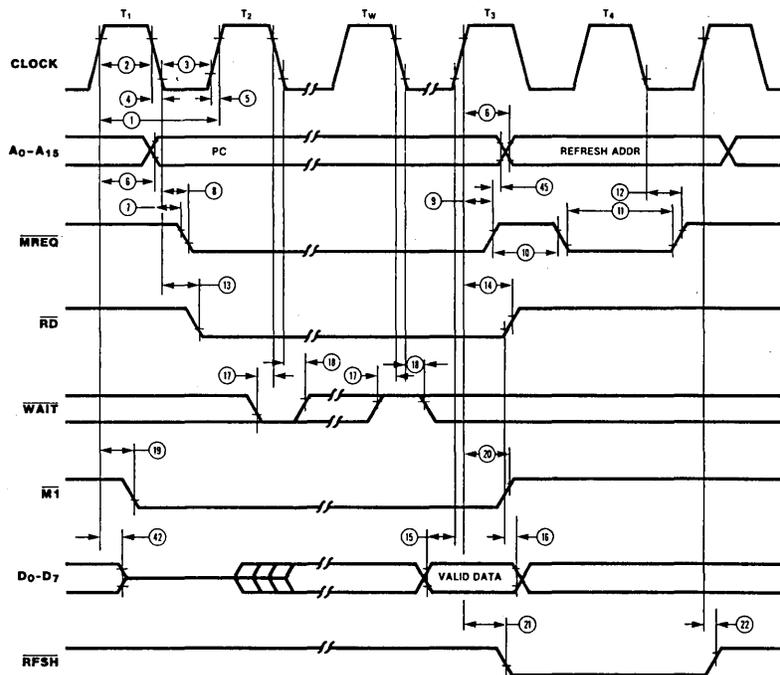
- Memory read or write
- I/O device read or write
- Interrupt acknowledge

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, MREQ goes active. The falling edge of MREQ can be used directly as a Chip Enable to dynamic memories. When active, RD indicates that the memory data can be enabled onto the CPU

data bus.

The CPU samples the  $\overline{\text{WAIT}}$  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.



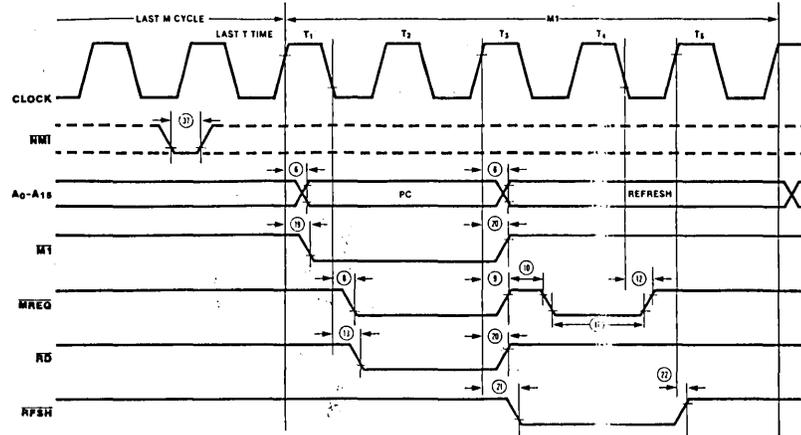
NOTE:  $T_w$ -Wait cycle added when necessary for slow ancillary devices.

Figure 5. Instruction Opcode Fetch

**CPU Timing**  
(Continued)

**Non-Maskable Interrupt Request Cycle.** 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 NMI service routine located at address 0066H (Figure 9).



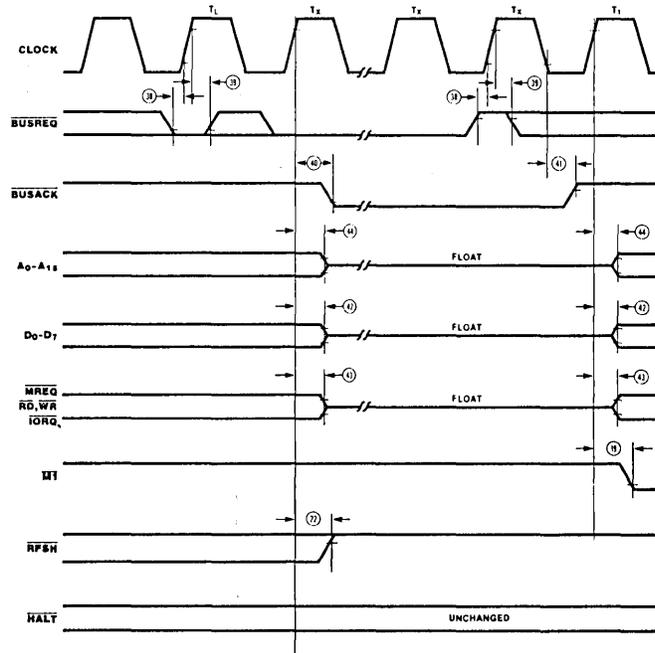
\* Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge

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.

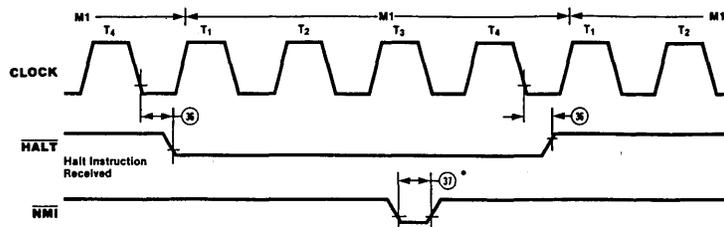
$T_X$  = An arbitrary clock cycle used by requesting device.

Figure 10. Bus Request/Acknowledge Cycle

**CPU  
Timing**  
(Continued)

**Halt Acknowledge Cycle.** When the CPU receives a  $\overline{\text{HALT}}$  instruction, it executes NOP states until either an  $\overline{\text{INT}}$  or  $\overline{\text{NMI}}$  input is

received. When in the Halt state, the  $\overline{\text{HALT}}$  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.**  $\overline{\text{RESET}}$  must be active for at least three clock cycles for the CPU to properly accept it. As long as  $\overline{\text{RESET}}$  remains active, the address and data buses float, and the control outputs are inactive. Once  $\overline{\text{RESET}}$  goes

inactive, two internal T cycles are consumed before the CPU resumes normal processing operation.  $\overline{\text{RESET}}$  clears the PC register, so the first opcode fetch will be to location 0000 (Figure 12).

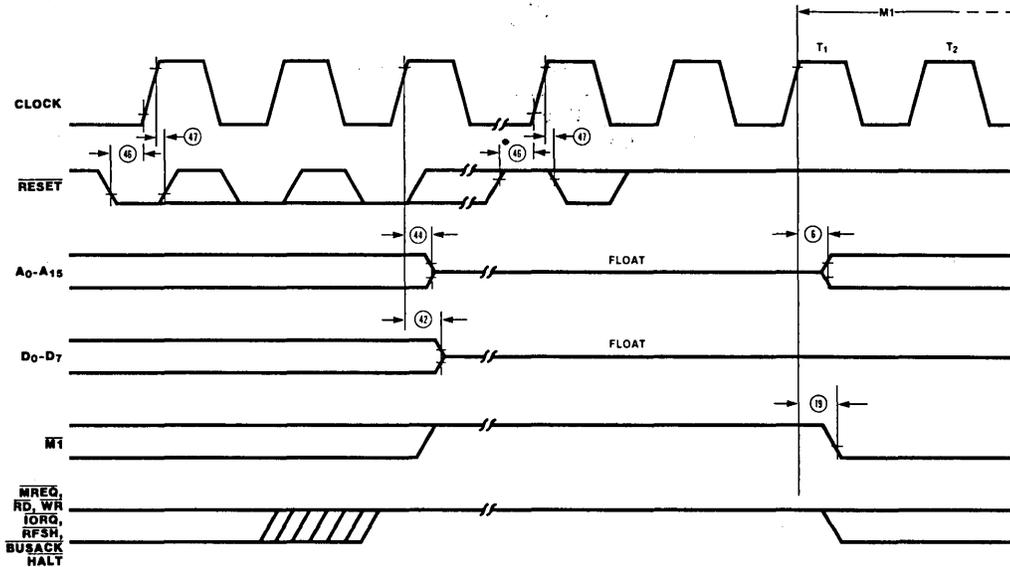


Figure 12. Reset Cycle

**AC  
Charac-  
teristics**

Number	Symbol	Parameter	Z80 CPU		Z80A CPU		Z80B CPU	
			Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	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 $\overline{\text{MREQ}}$ ↓ Delay	125*	—	65*	—	35*	—
8	TdCf(MREQf)	Clock ↓ to $\overline{\text{MREQ}}$ ↓ Delay	—	100	—	85	—	70
9	TdCr(MREQr)	Clock ↑ to $\overline{\text{MREQ}}$ ↑ Delay	—	100	—	85	—	70
10	TwMREQh	$\overline{\text{MREQ}}$ Pulse Width (High)	170*	—	110*	—	65*	—
11	TwMREQl	$\overline{\text{MREQ}}$ Pulse Width (Low)	360*	—	220*	—	135*	—
12	TdCf(MREQr)	Clock ↓ to $\overline{\text{MREQ}}$ ↑ Delay	—	100	—	85	—	70
13	TdCf(RDf)	Clock ↓ to $\overline{\text{RD}}$ ↓ Delay	—	130	—	95	—	80
14	TdCr(RDr)	Clock ↑ to $\overline{\text{RD}}$ ↑ Delay	—	100	—	85	—	70
15	TsD(Cr)	Data Setup Time to Clock ↑	50	—	35	—	30	—
16	ThD(RDr)	Data Hold Time to $\overline{\text{RD}}$ ↓	—	0	—	0	—	0
17	TsWAIT(Cf)	$\overline{\text{WAIT}}$ Setup Time to Clock ↓	70	—	70	—	60	—
18	ThWAIT(Cf)	$\overline{\text{WAIT}}$ Hold Time after Clock ↓	—	0	—	0	—	0
19	TdCr(MIf)	Clock ↑ to $\overline{\text{MI}}$ ↓ Delay	—	130	—	100	—	80
20	TdCr(MIr)	Clock ↑ to $\overline{\text{MI}}$ ↑ Delay	—	130	—	100	—	80
21	TdCr(RFSHf)	Clock ↑ to $\overline{\text{RFSH}}$ ↓ Delay	—	180	—	130	—	110
22	TdCr(RFSHr)	Clock ↑ to $\overline{\text{RFSH}}$ ↑ Delay	—	150	—	120	—	100
23	TdCf(RDr)	Clock ↓ to $\overline{\text{RD}}$ ↑ Delay	—	110	—	85	—	70
24	TdCr(RDf)	Clock ↑ to $\overline{\text{RD}}$ ↓ 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 $\overline{\text{IORQ}}$ ↓	320*	—	180*	—	110*	—
27	TdCr(IORQf)	Clock ↑ to $\overline{\text{IORQ}}$ ↓ Delay	—	90	—	75	—	65
28	TdCf(IORQr)	Clock ↓ to $\overline{\text{IORQ}}$ ↑ Delay	—	110	—	85	—	70
29	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	190*	—	80*	—	25*	—
30	TdCf(WRf)	Clock ↓ to $\overline{\text{WR}}$ ↓ Delay	—	90	—	80	—	70
31	TwWR	$\overline{\text{WR}}$ Pulse Width	360*	—	220*	—	135*	—
32	TdCf(WRr)	Clock ↓ to $\overline{\text{WR}}$ ↑ Delay	—	100	—	80	—	70
33	TdD(WRf)	Data Stable prior to $\overline{\text{WR}}$ ↓	20*	—	-10*	—	-55*	—
34	TdCr(WRf)	Clock ↑ to $\overline{\text{WR}}$ ↓ Delay	—	80	—	65	—	60
35	TdWRr(D)	Data Stable from $\overline{\text{WR}}$ ↑	120*	—	60*	—	30*	—
36	TdCf(HALT)	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓	—	300	—	300	—	260
37	TwNMI	$\overline{\text{NMI}}$ Pulse Width	80	—	80	—	70	—
38	TsBUSREQ(Cr)	$\overline{\text{BUSREQ}}$ 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		Z80A CPU		Z80B CPU	
				Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)
	39	ThBUSREQ(Cr)	BUSREQ Hold Time after Clock ↑	0	—	0	—	0	—
	40	TdCr(BUSACKf)	Clock ↑ to BUSACK ↓ Delay	—	120	—	100	—	90
	41	TdCf(BUSACKr)	Clock ↓ to 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 (MREQ, IORQ, RD, and WR)	—	110	—	80	—	70
	44	TdCr(Az)	Clock ↑ to Address Float Delay	—	110	—	90	—	80
	45	TdCTr(A)	Address Stable after MREQ ↑, IORQ ↑, RD ↑, and WR ↑	160*	—	80*	—	35*	—
	46	TsRESET(Cr)	RESET to Clock ↑ Setup Time	90	—	60	—	60	—
	47	ThRESET(Cr)	RESET to Clock ↑ Hold Time	—	0	—	0	—	0
	48	TsINTf(Cr)	INT to Clock ↑ Setup Time	80	—	80	—	70	—
	49	ThINTr(Cr)	INT to Clock ↑ Hold Time	—	0	—	0	—	0
	50	TdMf(IORQf)	M̄ ↑ to IORQ ↓ Delay	920*	—	565*	—	365*	—
	51	TdCf(IORQf)	Clock ↓ to IORQ ↓ Delay	—	110	—	85	—	70
	52	TdCf(IORQr)	Clock ↓ to 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 = TIC = 20 ns.

#### Footnotes to AC Characteristics

Number	Symbol	Z80	Z80A	Z80B
1	TcC	TwCh + TwCl + TrC + TIC	TwCh + TwCl + TrC + TIC	TwCh + TwCl + TrC + TIC
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 + TIC - 75	TwCh + TIC - 65	TwCh + TIC - 50
10	TwMREQh	TwCh + TIC - 30	TwCh + TIC - 20	TwCh + TIC - 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	TdMf(IORQf)	2TcC + TwCh + TIC - 80	2TcC + TwCh + TIC - 65	2TcC + TwCh + TIC - 50

AC Test Conditions:  
V<sub>IH</sub> = 2.0 V  
V<sub>IL</sub> = 0.8 V  
V<sub>IHC</sub> = V<sub>CC</sub> - 0.6 V  
V<sub>ILC</sub> = 0.45 V  
V<sub>OH</sub> = 2.0 V  
V<sub>OL</sub> = 0.8 V  
FLOAT = ±0.5 V

APPENDIX F.....FLOPPY ERROR CODE

Table 6 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	WRITE FAULT	WRITE FAULT
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	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

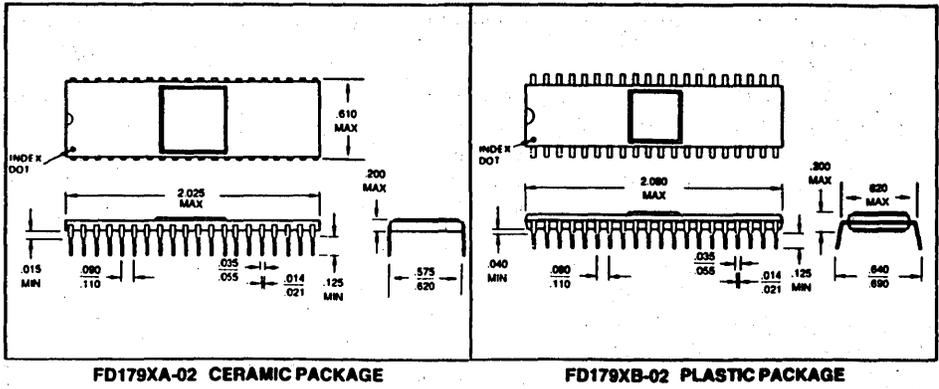
BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED	When set, indicates Write Protect is activated. This bit is an inverted copy of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

STATUS FOR TYPE II AND III COMMANDS

BIT NAME	MEANING
S7 NOT READY	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 'ored' with MR. The Type II and III Commands will not execute unless the drive is ready.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE/ WRITE FAULT	On Read Record: It indicates the record-type code from data field address mark. 1 = Deleted Data Mark. 0 = Data Mark. On any Write: It indicates a Write Fault. This bit is reset when updated.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when updated.
S1 DATA REQUEST	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.

20-1060

FORMS  
2



This is a preliminary specification with tentative device parameters and may be subject to change after final product characterization is completed.

Information furnished by Western Digital Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Western Digital Corporation for its use; nor any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Western Digital Corporation. Western Digital Corporation reserves the right to change said circuitry at anytime without notice.

**WESTERN DIGITAL**  
CORPORATION

3128 REDHILL AVENUE, BOX 2180  
NEWPORT BEACH, CA 92663 (714) 557-3550, TWX 910-595-1139



#### 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.

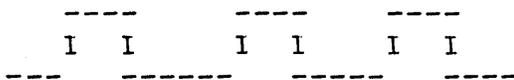
The warranty on the super quad is one year from the date of purchase.

TECHNICAL APPLICATION NOTES FOR THE SUPER QUAD SINGLE BOARD COMPUTER.

In case of problems with the SUPER QUAD; these are two things a customer may do:

1. send the product back for warranty service (turn around time is 1 day)
2. follow the following notes and then if there are still no results do the first step!

1. If the board is totally dead and won't communicate with the CRT, then do the following:
  - a. check your power supply for 5,+,-12 volts.
  - b. check pins 1,3,17 of the U53 (BR1941) baud rate gen. for step type pulses. (if you do not see anything on these pins then replace the BR1941 chip)



- c. Check your PS NET/I the serial port adapter cable for possible loss of + or - 12 volts or the 1488 chip.
  - d. check pin 3 of U11 (741s11). there should be a positive going signal (about 5 volts) and when you press RESET you should see that pulse going to ZERO voltage.
  - e. If the step d is OK then you have a problem with either one of the a or b or c.
  - f. check the 4MHZ clock signal going to pin 6 of the CPU.
2. If there is a problem with the floppy disk controller part only and you think that the phase lock loop needs to be adjusted, then follow the following steps:
  - a. This is done with the help of a dual trace scope.
  - b. put one channel on the 4MHZ signal (PIN 6 of the cpu chip) put the other channel on pin 7 of the U27 (74s124) and by tweaking the R26 (pot) those two signals are supposed to lock.
  - c. the above procedure is only recommended to engineers or technicians with experience of doing such things before.
3. Make sure the board is being properly cooled by a fan.
4. The power consumption of the board is:
  - +8v--- 1.8 amps max.
  - +16--- .4 amps max.
  - 16--- .2 amps max.
5. To ensure the proper operation of your floppy disk drives, make sure to put the termination resistor at the end of the cable.
6. Refer to the software manual for software problems.

Application notes to run TURBO-DOS operating system:

If you have one of the early revisions of the SUPERQUAD check the following:

1. pin 13 of the CTC should be hooked to +5V.
2. R17 should be hooked to +5.
3. for double sided drives, there should be a jumper from j3 connector to PIO bit3(PIO-29)
4. IE3 lin2 (PIO-22) should goto u18-9, on some older rev. boards its also going to pin 12 of u14. the place to cut would be on top of u8 right above pin 20 there is a pad hole; the etch going north should be cut, then install a jumper from u14-12 to u15-6

application notes on how to interface the Measurement Systems Memory (DMB6400)  
to the SUPERQUAD.

ON DMB6400 SET SWITCHES AS FOLLOWS:  
(TOGGLE SWITCH...UP MEANS PRESS TO THE TOP DOWN)  
S1-1 DOWN, S1-2 DOWN, S1-3 DOWN, S1-4 DOWN

S2-1, 2, 4, 5 UP-----S3-1, 2 UP----S4-1, 4 DOWN----S5-1, 2, 3, 4, 5, 6, 8 UP

HEADER 1 SHOULD BE WIRED AS: PIN 1 TO 16, 3 TO 9, 7 TO 11, 8 TO 10.

## SUPER NET MODIFICATION SHEET

To run the SUPER QUAD (Advanced Digital Corporation's S-100 single board computer) with the BSR 64/256 (PCE Systems' 256K RAM card) there are two modifications to be made to the SUPER QUAD. The first modification (as described below) is to buffer the Z-80 refresh signal, and bring it to pin 66 of the S-100 bus (see figure 1).

- 1) Solder a wire from U50, pin 28 (Z-80) to U43, pin 14 (8T97).
- 2) Solder a wire from U43, pin 13 (8T97) to pin 66 of the S-100 bus.

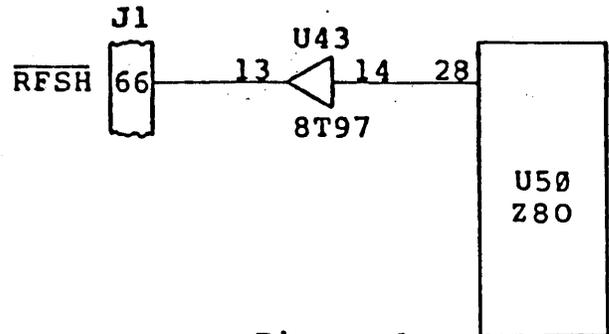


Figure 1

The second modification needs to be done so that the RAM card can sense when the Z80 is in a wait state. This is done by cutting the RAM RFSH and FDC WAIT signals on the SUPER QUAD away from the inputs of U56, and ANDING them and tying the result to PRDY of the S-100 bus (see figure 2). Because PRDY may be driven by a number of different cards, an open collector device is needed. For this purpose a 7409 is added to the board, as described below, at SPARE location U57.

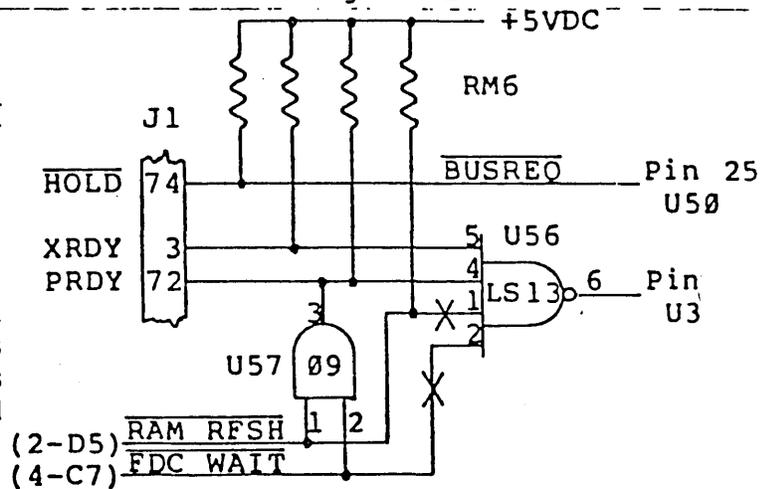


Figure 2

- 1) Solder a 14 pin socket in location U57 for a 7409.
- 2) Solder a wire from RM6, pin 2 to U57, pin 1.
- 3) Solder a wire from the plated through hole located under U55 between pins 13 & 14, to U57, pin 2.
- 4) Solder a wire from U57, pin 3 to U56, pin 4.
- 5) Cut two traces located on component side of board between U56 and RM6 (see figure 3).
- 6) Install a 7409 in U57.

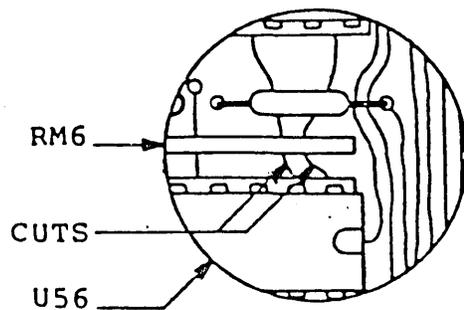


Figure 3

*Open hole plated through hole between C37, R32 to U57 (7409)*

12.0 Parts list  
-----

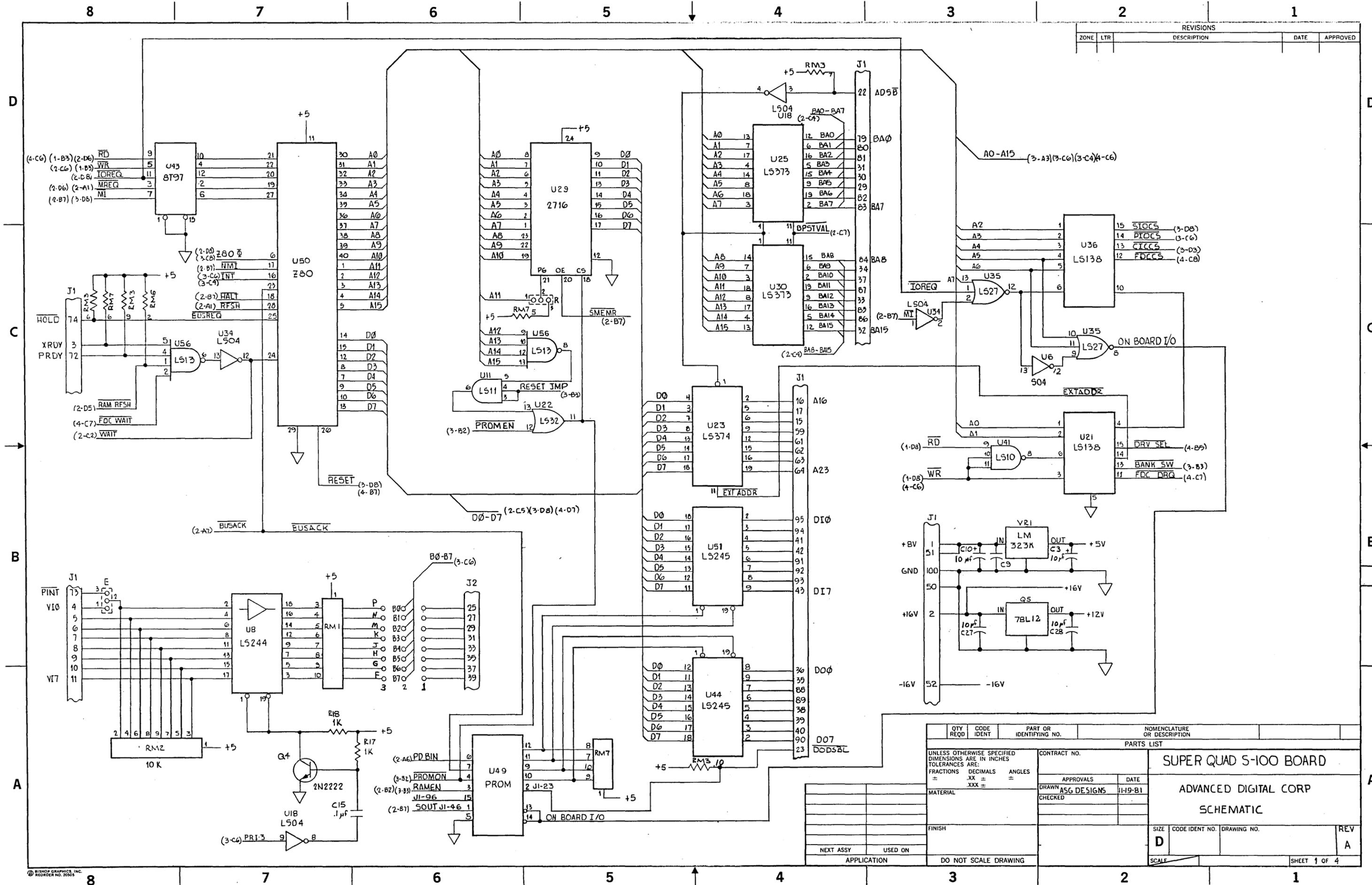
SUPER NET Parts list

Item	I	Part Number	I	Location on board	I	Quantity
1.	I	74LS00	I	U-13	I	1
2.	I	74S04	I	U-6	I	1
3.	I	74LS04	I	U-18,U-34	I	2
4.	I	7406	I	U-32	I	1
5.	I	7407	I	U-1	I	1
6.	I	74LS10	I	U-41	I	1
7.	I	74LS11	I	U-11	I	1
8.	I	74LS13	I	U-47,U-56	I	2
9.	I	74LS14	I	U-2	I	1
10.	I	74LS27	I	U-35	I	1
11.	I	74LS32	I	U-42,U-22	I	2
12.	I	7438	I	U-14	I	1
13.	I	74LS74	I	U-33,U-40	I	2
14.	I	74LS123	I	U-3,U-4	I	2
15.	I	74S124	I	U-27	I	1
16.	I	74LS132	I	U-16,U-39,U-46	I	3
17.	I	74LS138	I	U-21,U-36	I	2
18.	I	74LS139	I	U-5	I	1
19.	I	74LS153	I	U-28	I	1
20.	I	74LS175	I	U-10	I	1
21.	I	74LS240	I	U-59,U-24	I	2
22.	I	74S240	I	U-68,U-69	I	2
23.	I	74LS244	I	U-8	I	1
24.	I	74LS245	I	U-44,U-51	I	2
25.	I	74LS273	I	U-17	I	1
26.	I	74S287 (PROM)	I	U-49	I	1
27.	I	74LS373	I	U-25,U-30	I	2
28.	I	74LS374	I	U-23	I	1
29.	I	74LS393	I	U-12,U-7	I	2
30.	I	MB4164-20	I	U-60 THRU U-67	I	8
31.	I	Z-80A CPU	I	U-50	I	1
32.	I	Z-80A PIO	I	U-9	I	1
33.	I	Z-80A DART (SIO)	I	U-52	I	1
34.	I	Z-80A CTC	I	U-37	I	1
35.	I	WD-1793 (8877)	I	U-26	I	1
36.	I	WD-1691	I	U-19	I	1
37.	I	WD-2143	I	U-20	I	1
38.	I	BR-1941	I	U-53	I	1
39.	I	TTLDM-100	I	U-54	I	1
40.	I	2716 EPROM	I	U-29	I	1
41.	I	8T97	I	U-43	I	1
42.	I	8T98	I	U-55,U-15	I	2
43.	I	14 PIN HEADER	I	U-31	I	1
44.	I	LM323K VOLT.	I	VR1	I	1
45.	I	8.000 MHZ XTAL	I	Y1	I	1
46.	I	5.06 MHZ XTAL	I	Y2	I	1
47.	I	16 PIN SWITCH	I	SW1	I	1
48.	I	2N2222 TRANS.	I	Q1,Q2,Q4	I	3
49.	I	2N3906 TRANS.	I	Q3	I	1

50.	I	78L12	I	Q5	I	1
51.	I	220 MH CHOKE	I	L1,L2	I	2
52.	I	4.7K SIP	I	RM1, RM3, RM7	I	3
53.	I	10K SIP	I	RM2, RM5	I	2
54.	I	33 OHM SIP	I	RM6	I	1
55.	I	220/330 SIP	I	RM4	I	1
56.	I	33 OHM DIP	I	RM8	I	1
57.	I	1N914 DIODE	I	CR2	I	1
58.	I	5.1 V ZENER	I	CR1	I	1
59.	I	100 MF CAP	I	C2	I	1
60.	I	.1 MF CAP	I		I	21
61.	I	4.7 MF CAP	I	C6, C3, C10, C27, C28	I	8
62.	I	10K POT	I	R27, R28	I	2
63.	I	14 PIN HEADER	I	J4, J5	I	2
64.	I	50 PIN HEADER	I	J3	I	1
65.	I	40 PIN HEADER	I	J2	I	1

13.0 Schematic diagram

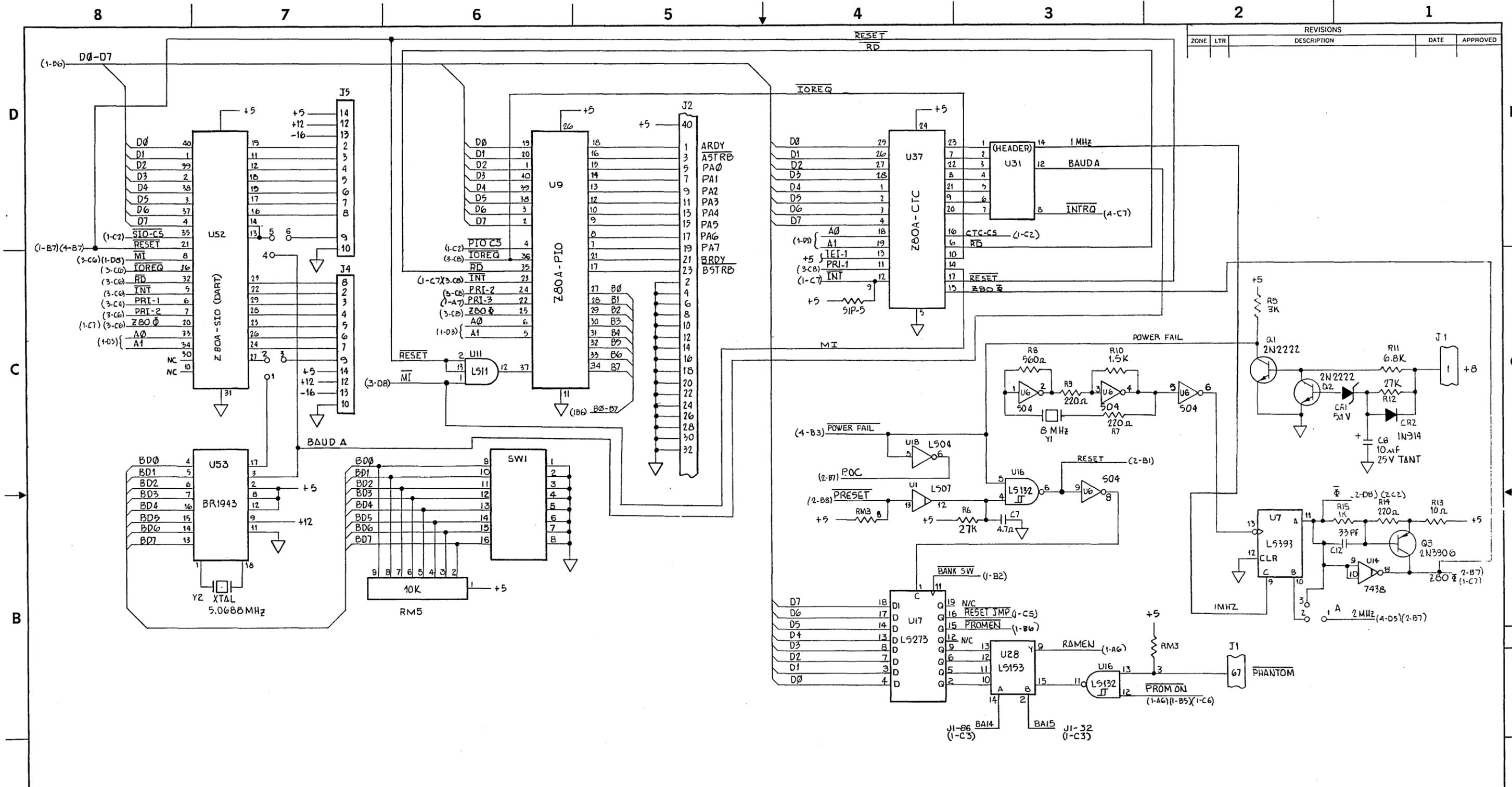
-----



ZONE		LTR		REVISIONS		DATE	APPROVED

QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION
PARTS LIST			
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES			
CONTRACT NO.			
APPROVALS		DATE	
DRAWN ASG DESIGNS		11-19-81	
CHECKED			
MATERIAL			
FINISH			
NEXT ASSY		USED ON	
APPLICATION			
DO NOT SCALE DRAWING			
SIZE	CODE IDENT NO.	DRAWING NO.	REV
D			A
SCALE			SHEET 1 of 4

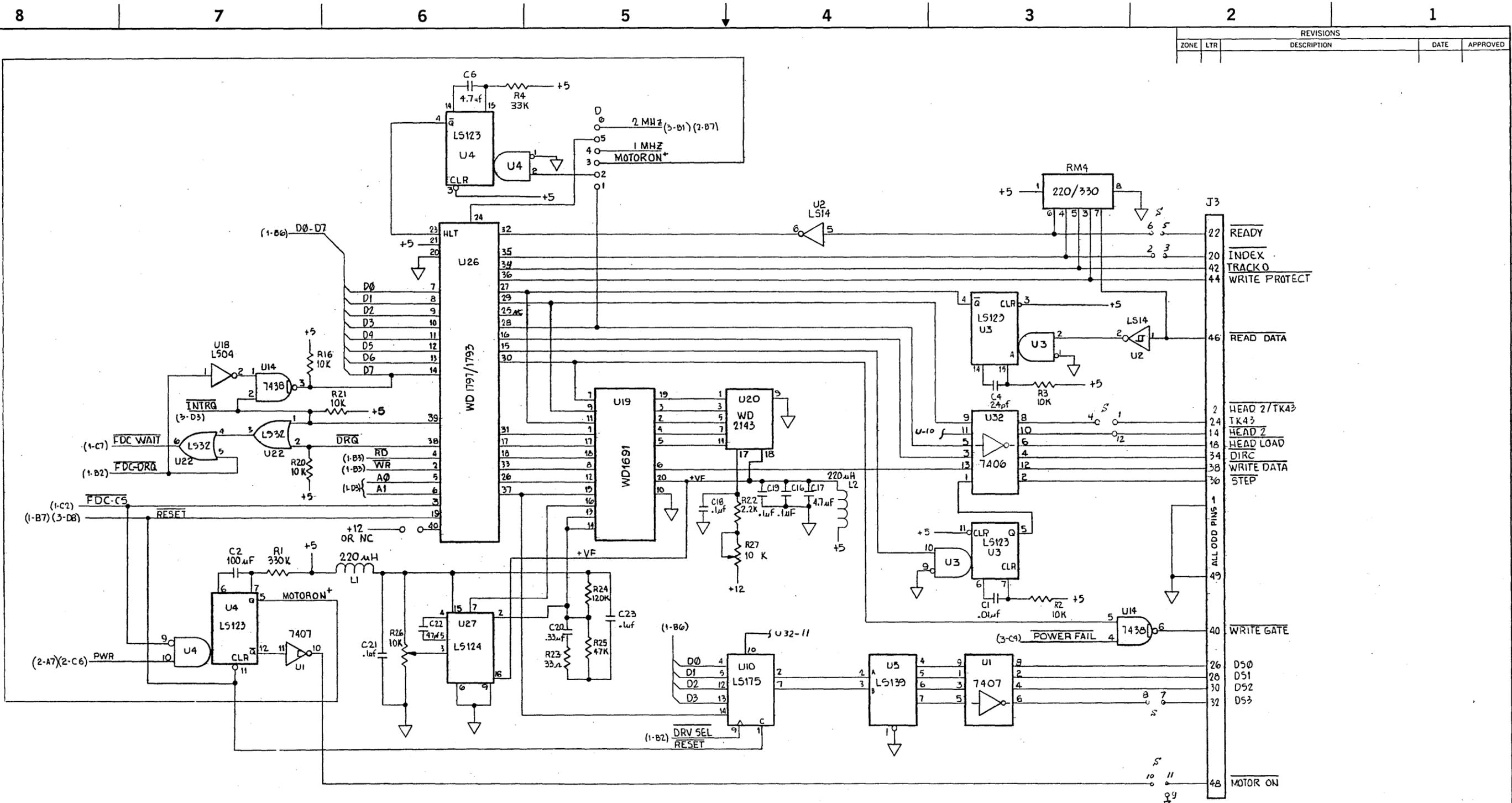




ZONE		LTR		REVISIONS	DATE	APPROVED

QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION
PARTS LIST			
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE:		CONTRACT NO.	
FRACTIONS	DECIMALS	ANGLES	
=	=	=	
MATERIAL		APPROVALS	DATE
FINISH		DRAWN ASG DESIGNS	11-19-81
NEXT ASSY USED ON		CHECKED	
APPLICATION		SUPER QUAD S-100 BOARD	
DO NOT SCALE DRAWING		ADVANCED DIGITAL CORP.	
		SCHEMATIC	
SIZE	CODE IDENT NO	DRAWING NO.	REV
D			A
SCALE		SHEET 3 OF 4	

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED



QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION
PARTS LIST			
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE:		CONTRACT NO.	
FRACTIONS .XX ±	DECIMALS .XXX ±	ANGLES ° ±	SUPER QUAD S-100 BOARD
APPROVALS			
DRAWN ASG DESIGNS			DATE 11-19-81
CHECKED			
MATERIAL			
FINISH			
NEXT ASSY USED ON			
APPLICATION			
DO NOT SCALE DRAWING			
SIZE D	CODE IDENT NO.	DRAWING NO.	REV A
SCALE			SHEET 4 OF 4