FDC-I

User Manual

Teletek Enterprises, Inc.
Sacramento, California, USA
Revision 5, September 1980

Errata
Revision 6 FDC-I Changes
4/8/81

Pre-Write Compensation, page 10

The jumper pads on the back of the revision 6 FDC-I labeled NC, FD, and PW have been replaced by posts and a shorting plug. The posts are located on the front of FDC-I in the upper left hand corner directly below U-43. The left post is now PW, the center post is FD, and the right post is NC.

Extended Head Load, page 12

On revision 6 FDC-I the extended head load capacitor can now be mounted on the component side of the board if so desired. The holes for the capacitor are located directly above U-16 (shown in dotted lines on the Appendix A layout). The three pads labled HL, HD, and HM have been moved from the back of U-16 to the back of U-28 near U-41. As before, the top pad is HL, the middle pad is HD, and the bottom is HM.

Errata for FDC-I User Manual, Revision 5
8-17-81

I. Appendix C

A. The configuration sheets for the following drives has been added:

1. Control Data 9406-2/3
2. NEC FD1160
3. Qume DataTrak 5
4. Pertec FD 250
5. Pertec FD650/651
6. Shugart SA850/851
7. Tandon TM 100

B. Page 4 - Item 4 should read as follows:

4. Since most mini-floppy drives no not provide a "Ready" signal, the "Ready" signal is tied to ground on the adaptor board.

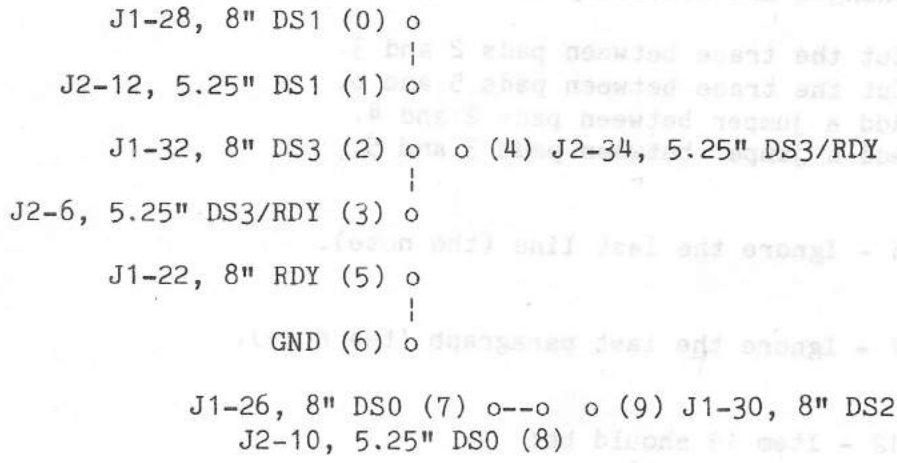
The diagram of the adaptor board should be ignored. See Figure 1 for the new p.c. adaptor board with its options.

Add item 5 as follows:

5. If there is a requirement to operate both 8" and 5.25" drives at the same time, consideration must be give as to the amount of pre-write compensation that is needed by each size drive. If the 5.25" drives require an amount that is other than twice what is required on the 8" drives a modification to the switching circuit is needed. Contact Teletek for further details.

Figure 1

Eight inch to five and a quarter inch drive p.c. adaptor board:



J1 is the 50 pin connector and J2 is the 34 pin connector.

Options:

1) Pad 0 to 1 - 8" drive select 1 connected to 5.25" drive select 1.

2) Pad 1 to 2 - 8" drive select 2 connected to 5.25" drive select 1.

3) Pad 2 to 3 - 8" drive select 3 connected to 5.25" pin 6. Normally this pin is drive select 3 for 5.25" drives but is the READY signal for Micropolis drives.

4) Pad 2 to 4 - 8" drive select 3 connected to 5.25" pin 34. This pin is drive select 3 for Micropolis drives or READY for Pertec drives.

5) Pad 5 to 3 - 8" READY connected to 5.25" pin 6. This pin is the READY pin for Micropolis drives.

6) Pad 5 to 6 - 8" READY connected to GROUND. Since most 5.25" drives do not provide a READY signal it is neccessary to ground this line.

7) Pad 7 to 8 - 8" drive select 0 connected to 5.25" drive select 0.

8) Pad 9 to 8 - 8" drive select 2 connected to 5.25" drive select 0.

C. Page 5 - Add the following note:

Note: Since the Micropolis drive does not follow the ANSI standard several changes are neccessay on the p.c. adaptor board as follows:

1. Cut the trace between pads 2 and 3.
2. Cut the trace between pads 5 and 6.
3. Add a jumper between pads 2 and 4.
4. Add a jumper between pads 3 and 5.

D. Page 6 - Ignore the last line (the note).

E. Page 7 - Ignore the last paragraph (the note).

F. Page 12 - Item 13 should be:

13. 2S Jumper

G. Page 13 - The title should read "Shugart 800/801 Disk Drive" not "Shugart 800/851 Disk Drive". The drive configuration for the Shugart 850/851 disk drive is included as a separate page at the end of appendix C.

H. Page 14 - Replace the second to the last paragraph with the following paragraph:

The Siemens drives need to be modified if more than 1 drive will be in the system. On the drive p.c. board, cut the trace going to pin 9 of IC 6C. Add a jumper between pins 9 and 12 of IC 6C. This change accommodates the NEC controller.

II. Appendix D

A. Page 1 - Add to the MSC DMB-6400 the following sentence:

Replace the delay line at U64 (lower right hand corner) with a 14 pin dip header. Jumper pins 1 and 12 on the header.

B. Page 4 - Add the following for Teletek's memory board:

Required Configuration:

1. Area A - Jumper SM1 to the center post
2. Area B - Jumper SM180 to the center post
3. Area C - Jumper PDBIN to the center post
4. Area D - Jumper Ø (PHI) to the center post
5. Area E - Jumper SM1 to the center post
6. Area F - Jumper PDBIN to the center post
7. The other jumper options depend on the desired bank options, refer to the memory user manual.

III. Appendix E

The following changes are necessary for release 3.2 of the FDC-I monitor:

A. Page 8 - Add the following paragraph at the end of the page:

BP5A RTIVT This routine will return the address of the interrupt vector table in the register pair HL. The user should use this routine to store his interrupt vectors. Locations E0-FF are reserved for the monitor.

B. Page 14 - Delete the first paragraph. Motor control is taken care of automatically in the monitor.

C. Page 14 - The column labeled "System RAM" should read as follows:

Version	System RAM
a	FE00-FFFF
b	EE00-EFFF**
c	F600-F7FF

D. Page 15 - The first two sentences should read as follows:

The FDC-I monitor uses 512 bytes for the interrupt vector table, stack, flags, pointers and other data for the I/O facilities of the monitor.

Errata for FDC-I User Manual, Revision 5
9-28-81

I. Appendix E

The following changes are necessary for release 4.0 of the FDC-I monitor:

A. Page 1 - Paragraph 2, line 4 should read:

When I/O has been assigned, "FDC-I r.rv Monitor" will printed on the output device (where r.r is the release number and v is the version of the monitor).

B. Page 1 - Paragraph 3, line 2 should read:

The user should hit the CR key until "FDC-I r.rv Monitor" appears.

C. Page 1 - Add the following note after paragraph 4:

NOTE: on releases previous to release 4.0 input on SIO A and B was on an interrupt basis with a 1 character buffer. Input on SIO A is now on an interrupt basis, with a 64 character buffer implemented. This means that the user may "type ahead" and not lose the characters he has typed. Input on SIO B is now on a polled basis, allowing easier software development for "modem" programs. Also, SIO B is no longer a valid input device in the IOBYTE. If all the user software has been written so that it only uses the monitor jump vectors these changes will be transparent. If the software did not use the jump vectors, he was forewarned in the "Monitor System RAM" section.

D. Page 2 - Paragraph 1, the last two lines should be replaced with the following:

Several editing features are available. The command buffer is first filled with spaces. Entering a key stores the character in the command buffer and echoes it unless it is one of the following keys. The Back Space key will back up the cursor one place. The DELete key will insert a space at the current cursor position. The space bar echoes the character in the command buffer at the cursor position. A control C will print "^X" and cancel the command line.

E. Page 2 - Paragraph 2 should read:

Control C (^C) has a special function. In the commands marked with <^X>, entering ^C will terminate the command prematurely. Hitting ^C will cause "^X" to print and the command input mode to be entered.

F. Page 2 - Paragraph 3 should read:

If the monitor does find a command the user entered it will echo the command followed by a question mark (?).

G. Pages 3 through 4 - All the "<ESC>"s should be replace with "<^X>". Also "Complete" is not longer printed when a command is terminated.

H. Page 3 - The "IO" command has been deleted. Add the following two commands:

I_H1<CR>	Input from port H1: inputs froms port H1 and prints the value. Hitting the space bar will input again, any other key will terminate the command.
O_H1_H2<CR>	Output to port H1: outputs H2 to port H1. Hitting the space bar will output again, any other key will terminate the command.

I. Page 4 - Paragraph 2: In the IOBYTE assignment, SIO B is no longer a valid input device. Bit 6 is now reserved.

J. Page 13 - Paragraph 4, line 3 should read:

The location will be in a conditional assembly:

```
if   eight ;inch drives
ld   a,-1
else ;five and a quarter inch drives
ld   a,0
endif.
```

K. Page 13 - Paragraph 5, line 3 should read:

The location will be in a conditional assembly:

```
if   auto
nop             ;fall thru to boot
nop
else
jr   wtio
endif.
```

L. Page 13 - Paragraph 6, lines 3 and 4 should read:

To enable this option the user has to program a FE hex in the RECAL and SEEK routines. The locations will be in a conditional assembly:

```
if   persci
and  0feh       ;reset bit 0
else
and  0ffh
endif.
```

M. Page 14 - Paragraph 1, line 1 should read:

The next option will permit the monitor to control the DC motor on Persci or Tandon drives.

N. Page 14 - Paragraph 1, the last two lines should read:

To enable this option, the user must program a NOP in the MOTOR routine. The location will be in a conditional assembly:

```
if   persci or tandon
nop             ;ignore drive size
else
ret  nz
endif.
```

O. Page 15 - The last line of item 6 of the Monitor Initialization should read:

SIO A input is set on an interrupt basis and its output on a polled basis. SIO B input and output is set on a polled basis.

P. The monitor listing should be the listing for release 4.0.

Table of Contents

Specifications

Central processor: Z80A CPU - 2 or 4 MHz operation.

Timer: Z80A CTC - 4 channels, 2 used for serial ports, 2 used for real-time clock

Serial: Z80A SIO - 2 RS-232, independent operation. Speeds from 45 to 9600 baud.

Parallel: Z80A PIO - 1 bidirectional port with 4 handshake lines, 3 independent input or output lines.

Floppy disk controller: NEC uPD 765 single or double density operation, mini or maxi drives, ANSI standard 50 pin connector, IBM compatible format.

Disk data transfer rates:

Single density (FM)	5-1/4" - 125k bits/sec	8" - 250k bits/sec
Double density (MFM)	5-1/4" - 250k bits/sec	8" - 500k bits/sec

EPROM/RAM: 2716 (Intel-type), Mostek 4118 RAM, up to 8k bytes total.

EPROM programmer: Off-board power supply is required for programming power: +25.5 ±.7 volts @ 50 mA.

S-100 bus signals:

A0-A15	PHLDA	RFSH
CDSBL	PHOLD	
CLOCK	PINT	SHLTA
DATA IN	PINTE*	SINP
DATA OUT	POC	SINTA
MWRITE	PRDY	SM1
NMI	PRESET	SMEMR
Ø	PSYNC	SOUT
PSTVAL	PWAIT**	SWO
PDBIN	PWR	XRDY

Note: FDC-I does not provide 8080-type I/O addressing; only the lower 8 address lines contain the I/O address.

Dimensions: 5.05" x 10.0", excluding edge connector.

Power requirements: +8v @ 1.5 amp, +16v @ 50mA, -16v @ 50mA.

Workmanship conforms to the requirements of MIL-STD-454.

Forced air cooling is required.

*PINTE is not a IEEE S-100 bus signal but is provided on pin 28 as an Interrupt Enable Out signal.
**PWAIT is not a IEEE S-100 bus signal but is provided on pin 27 for dynamic ram boards which need to know if the processor is in a wait state.

Basic Description

The FDC-I is a microcomputer on a board. It incorporates most of the features required in a small computing system. On board is a Z80A CPU which operates at 4 MHz for high-speed efficient processing of information. The Z80A provides the capability to support many sophisticated applications. The interrupt structure of the Z80A is particularly important for systems which do multiple tasks concurrently. The FDC-I utilizes the structured interrupt system of the Z80A in most of its I/O capabilities. The CPU adds one wait state to on-board memory functions so that standard 450 ns 2716's can be used with the system.

Also on-board is a counter-timer chip which provides software-settable clocks for both serial ports and a real time clock. The real time clock is used by the monitor to provide timekeeping functions. It normally functions under interrupt control requiring a minimum of overhead. This real time clock can be used by software for any time-related functions, such as time dating of files, a stop watch or timing loops for external operations.

The Z80A PIO provides two parallel ports. One of these two ports is bidirectional with 8 data and 4 handshake lines. Normally this port is configured as a keyboard input, but because it is under software control, it can be reconfigured by the user to be a latching output or a truly bidirectional port. The second parallel port has 3 data lines available which can be set independently to be input or output lines. The remaining lines of this port are used for on-board functions within the FDC-I. If those functions are not needed, additional lines are made available on the second port.

Providing two independent serial ports, the Z80A SIO provides RS232C compatible serial ports which can be operated under interrupt control. Both serial ports include full handshaking for connection to external devices as a printer, CRT terminal, or MODEM.

Using the NEC 765 floppy disk controller IC, FDC-I provides single and double density data storage on both mini and maxi floppy disk drives providing capabilities which minimize the overhead burden on the CPU and software. Some of these capabilities are: single and double density data transfer under software control; performance of simultaneous seek operations on all drives connected to the system; IBM compatible formatting for ease of information exchange with controllers using similar operating system software; compatibility with both single and double sided drives; ANSI standard 50 pin disk drive connector; automatic reading of sequential sectors on a diskette; automatic reading of both tracks of a two-sided diskette; automatic error-checking detected via CRC; under software control, possible selection of sector size to be 128, 256, 512, or 1,024 bytes. The floppy disk control section of FDC-I also incorporates a phase-locked oscillator (PLO) which is used to stabilize the separated information and clock for precise data recovery.

With a jumper to the PIO B, software can control the size of floppy disk drive running in the system. Thus, mini and maxi drives can be intermixed with appropriate software.

The on-board memory of FDC-I can provide up to 8k bytes of storage in either EPROM/ROM/RAM. With appropriate RAM IC's, EPROM, ROM or RAM can be mixed in any combination up to the limit of the board. The standard FDC-I is set up for 6k bytes of EPROM and 2k bytes of RAM. Other combinations are possible and require a change of one of the on-board bipolar PROMs, U-25, which maps the memory space of FDC-I. One 2716 EPROM with a monitor program, and one Mostek MK 4118 (1k byte) RAM are supplied on board.

A reset-jump circuit on FDC-I makes the CPU jump to the monitor software on board whenever the system reset button is activated. This is useful for systems which do not have a front panel. For systems with a front panel, reset-jump will override the functions of the front panel. Also, incorporated as part of the reset-jump circuit, a power-on-clear function is included which automatically generates a reset when power is first applied.

All of the hardware and monitor routines necessary to program and verify Intel-type 2716's are provided on board. An external power supply (+25.5±0.7 volts at 50 mA) is required to program these EPROMs.

As part of the standard FDC-I, a 2k monitor is provided which has routines for initializing the LSI circuits of the FDC-I, programming Intel-type 2716's, assigning input and output devices, loading and examining memory, moving and verifying memory, reading and writing to floppy disk (both single and double density), and transferring execution. The monitor also provides routines necessary to interface to the floppy disk drive which means the operating system need only make a simple call to perform disk operations. Included in the monitor is a boot routine which will read the first sector of track 0 of drive 0 in the system and then execute that sector. Normally, the first sector of track 0 of an operating system provides the cold start loader for loading the operating system into memory. This standard software package makes the task of system integration much easier.

FDC-I provides most of the functions required in a small microcomputer system. Just add a 64k RAM board and any essential peripherals for a complete system.

Visual Inspection

Upon receipt of FDC-I, check the shipping package for signs of abuse which may indicate possible damage. Check the board physically to look for any parts which may have been damaged during shipping. Note the presence of all normally supplied parts; that is, all female connectors for the peripheral connections at the top of the board, and the floppy disk cable and its two edge connectors. If any diskettes were shipped with FDC-I, check the diskettes for signs of damage which might be any bending or signs of a sharp object placed against the diskettes. Diskettes are quite fragile and any warping of the surface of the diskette will render it inoperative. Notify Teletek of any discrepancies and call Teletek first if there is shipping damage, as a claim will probably be made with the shipping company.

Installation

FDC-I is ready for immediate use upon receipt. It only requires that the peripherals which will be used with it be connected to the appropriate female connector which will then plug into the connectors along the top of the board. For the particular connections required, see the adjoining section entitled "Peripheral Connections".

FDC-I need only be plugged into a standard S-100 bus for power and it will be functional, able to utilize the peripherals connected to it with the memory on board. The FDC-I needs to be in a well ventilated area due to the high density of IC's on board. Ideally, the board should be mounted vertically in a stream of air which will be moving across the face of the board. Whatever the mounting position, forced-air cooling is essential. Bring peripheral cables neatly away from the board with enough slack to prevent any tension being applied to the cable, as this may cause the cable to separate from its crimp connection causing intermittent problems.

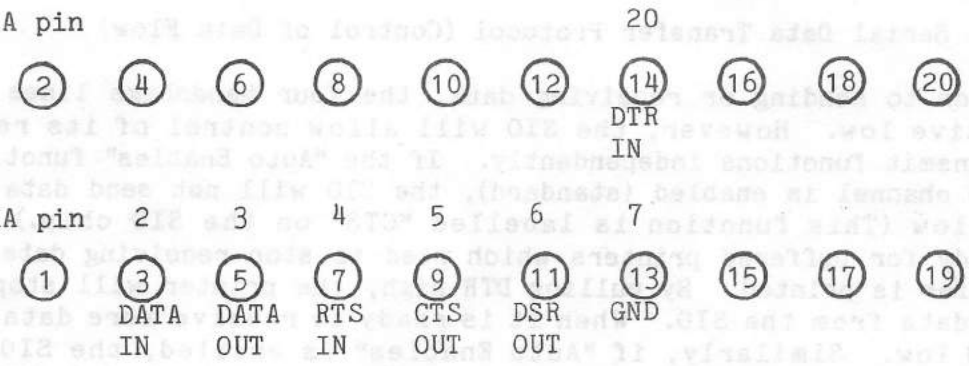
For serial console devices, SIO A is the primary port. With the standard monitor software, SIO A can determine the baud rate of a carriage return and thus set the appropriate speed automatically after a reset. The serial speed must be a standard value between 110 and 9600. Up to 6 carriage returns may be necessary for a 110 baud speed. Also, SIO A requires the handshake lines of the RS-232-C interface before it will function. See "Serial Ports" for further information.

Some versions of FDC-I do not have RAM on-board. If no RAM is on-board, there must be system memory immediately below the location of the monitor. See "EPROM/RAM Options" for further details on available configurations of the memory space allocation. There is no need to worry about memory conflicts with 64K memory boards, because FDC-I will automatically ignore memory conflicts with its on-board memory.

Peripheral Connections

Serial Ports

J-1, SIO A and J-2, SIO-B



These are the connections going into channels A and B of the SIO chip. In this configuration, each channel appears as a data communication device, and will connect to a terminal or a printer.

IN and OUT refer to data direction with respect to the FDC-I. Data from an external device is IN to FDC-I, and data to an external device is OUT.

CTS (Clear To Send) and DSR (Data Set Ready) are outputs to the external device and are at a positive voltage levels when the SIO channel is ready to function. RTS (Request To Send) and DTR (Data Terminal Ready) are inputs which must be at a positive voltage level for the SIO channel to function if the Auto Enables option is activated through software. This option is normally enabled in the standard FDC-I monitor. See the source listing for the machine language pertinent to this function.

Either channel can be crimp-connected to a 25 pin RS-232 connector by aligning pin 1 of the cable from the FDC-I connector with pin 1 of the 25 pin RS-232 connector. In this configuration, the channel connects directly to a terminal or printer. To connect to a MODEM, the signals must be connected as follows:

FDC-I SIO-A Pin #	EIA pin #	Direction	Function
5	2	OUT	Data to MODEM
3	3	IN	Data to FDC-I
11	4	OUT	RTS (Request To Send)
14	5	IN	CTS (Clear To Send)
7	6	IN	DSR (Data Set Ready)
13	7	--	Signal Ground
9	20	OUT	DTR (Data Terminal Ready)

IN refers to data sent to FDC-I, and OUT refers to data sent to the MODEM.

Note: If the terminal or printer does not provide RTS and DTR, pins 4, 5, and 20 on the terminal side of the RS-232 male connector must be jumpered together. This ensures that the required handshake signals to the SIO port are provided. If the AUTO ENABLES feature of the SIO is not enabled this is not required. In the standard software provided, AUTO ENABLES is enabled.

EIA Serial Data Transfer Protocol (Control of Data Flow)

Prior to sending or receiving data, the four handshake lines should be active low. However, the SIO will allow control of its receive and transmit functions independently. If the "Auto Enables" function of the SIO channel is enabled (standard), the SIO will not send data until DTR is low (This function is labelled "CTS" on the SIO chip.). This is handy for buffered printers which need to stop receiving data until the buffer is printed. By pulling DTR high, the printer will stop the flow of data from the SIO. When it is ready to receive more data, it pulls DTR low. Similarly, if "Auto Enables" is enabled, the SIO will not accept information until RTS is low (This function is labelled "DCD" on the SIO chip.). This is primarily used with a communications link where,if signal conditions deteriorate, the data may be garbled.

In summary, the handshake lines provide a convenient means of controlling the flow of information in a serial channel. If any line goes high, transfer ceases.

RS-232-C Voltage Levels

A logic high (a binary ONE), or marking condition, is any voltage less than -3 volts to a minimum of -25 volts. A logic low (a binary ZERO), or spacing condition, is any voltage greater than +3 volts to a maximum of +25 volts. Any level between -3 and +3 volts is undefined. This is called the transition region. The maximum transition time between bit cells is four per cent of the basic clock period. The maximum voltage rate of change (slew rate) is 30 volts/uSec. Thus the maximum RS-232-C transmission speed, based on voltage swings of -12 to +12 volts, is 50,000 baud.

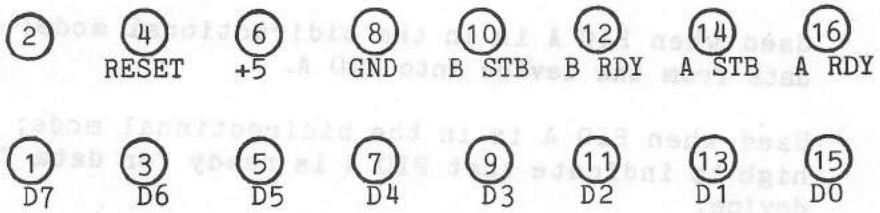
Serial Data Timing

Prior to transmitting data, the signal line is held high, or marking. It goes low or spacing to indicate the start of a character. The bits representing the character are then sent Least Significant Bit first, then a parity bit (if used), and finally 2 stop bits. The stop bits indicate the end of the character and are always logic ONEs. The standard FDC-I is set up for 8 data bits, no parity, and 2 stop bits.

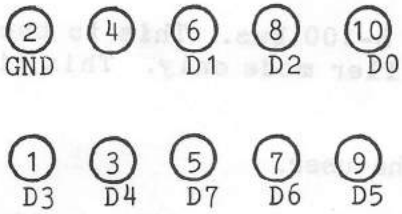
The value of each character bit is held for the entire length of each bit cell. The length in time of each bit cell is the basic clock period, equal to the reciprocal of the baud rate. Thus for 9600 baud, each bit cell is 104 uSec long (.0001041 Sec=1/9600).

Parallel Ports

J-3, PIO A



J-4, PIO B



These are the connections into the PIO chip. The PIO chip has two parallel ports, A and B. As configured, PIO A may be used as an input, output, bidirectional or control port with four handshake lines. PIO B is the same except that it does not have bidirectional capabilties or handshake lines.

The signals are:

D0 - D7 — 8 data lines

A STB — Strobe input pulse from a device. Depending on the mode of operation, it means:
1. Output mode: Positive edge of this strobe is issued by the device to acknowledge the receipt of data made available by PIO A.
2. Input mode: The strobe is issued by the device to load data from the device into PIO A.
3. Bidirectional mode: Same as 1, except output data is present only while A STB is low.
4. Control mode: The strobe is inhibited internally.

A RDY — Ready output to a device. Depending on the mode of operation, it means:
1. Output mode: Indicates that the data bus is stable for transfer to the device.
2. Input mode: When active, it indicates that PIO A is ready to accept data from the device.
3. Bidirectional mode: Same as 1.
4. Control mode: Always in a low state.

RESET — The active-low reset line on the FDC-I. This can be used to reset a Hard-Disk, such as the IMI 7710, connected to PIO A.

B STB — Used when PIO A is in the bidirectional mode; strobes data from the device into PIO A.

B RDY — Used when PIO A is in the bidirectional mode; it goes high to indicate that PIO A is ready for data from the device.

The monitor supplied by Teletek allows PIO A to be set up as an input port (for a keyboard), or an output port (for a parallel printer). PIO B is set up in the control mode as follows:

D7 is PHLDA from the S-100 bus. This is the DMA acknowledge line, used in the controller mode only. This line is not available for the user.

D6 is available for the user.

D5 is CDSBL from the S-100 bus. This is the Control Disable line, used in the controller mode only. In the CPU mode must alway be at a low logic level. This line is not available for the user.

D4 is free for the user; it can be connected through a jumper to one of the vectored interrupts on the S-100 bus if needed for handshaking to other devices in the system.

D3 is the RQ line; this can be a request line to the FDC-I.

D2 is the MC line; this line is used to control the dc motor on floppy drives. If not used, this line is free fot the user.

D1 is the PHLD from the S-100 bus. This is the DMA request line, used in the controller mode only. In the CPU mode must alway be at a low logic level. This line is not available for the user.

D0 is free for the user.

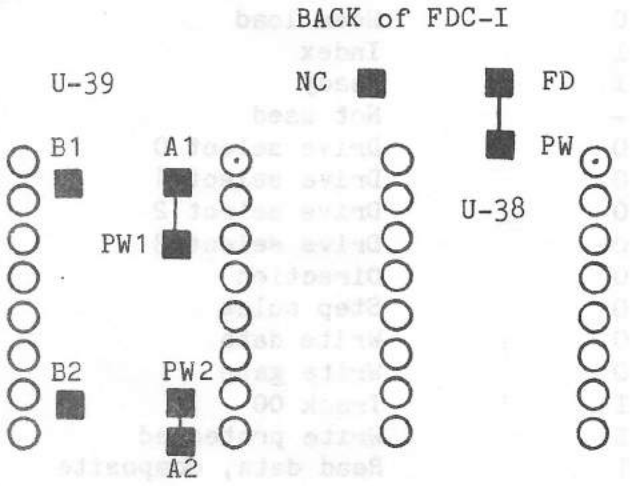
Floppy Disk Drive

Ground Pin #	Signal Pin #	Input - I Output - O	Description
1	2	O	Above track 43
3	4	-	Not used
5	6	-	Not used
7	8	O	Above track 43
9	10	I	Dual sided
11	12	-	Not used
13	14	O	Head 1
15	16	-	Not used
17	18	O	Head load
19	20	I	Index
21	22	I	Ready
23	24	-	Not used
25	26	O	Drive select 0
27	28	O	Drive select 1
29	30	O	Drive select 2
31	32	O	Drive select 3
33	34	O	Direction
35	36	O	Step pulse
37	38	O	Write data
39	40	O	Write gate
41	42	I	Track 00
43	44	I	Write protected
45	46	I	Read data, composite
47	48	-	Not used
49	50	O	Motor control (optional)

Input/Output are referenced to FDC-I. Input is a signal from the disk drive to FDC-I, and output is a signal to the disk drive.

Pre-Write Compensation

To help compensate for the shifting of data bits during the read process of the floppy disk drive, the write data is compensated. This is particularly critical for double-density operation. As seen in the previous disk drive section, different drives require different amounts of pre-write compensation. The symptoms of too much or not enough pre-write compensation are as follows: 1. Too much pre-write compensation shows up as read errors (usually CRC) in the outer tracks (0-42); 2. Not enough pre-write compensation shows up as read errors in the inner tracks (43-76). Jumper pads are provided on the BACK of FDC-I to adjust the amount of compensation.



Connection	Result
NC to FD	No Pre-Write Compensation
FD to PW	Standard. Provides Pre-Write Compensation.
B1 to PW1 B2 to PW2	Provides 125 ns compensation for 8" drives, and 250 ns for 5" drives.
A1 to PW1 A2 toPW2	Standard. Provides 250 ns compensation for 8" drives, and 500 ns for 5" drives.

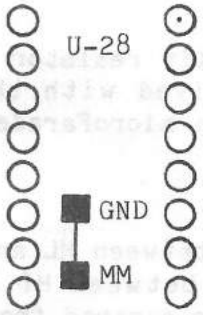
Mini/Maxi Floppy Selection

Through the use of one control line, the FDC-I can switch between mini (5 1/4") and maxi (8") floppy disk drives. This means that if this control line is connected to one of the unused data lines of PIO B, software control of the drive size is achieved. Otherwise, the control line is wired for either mini or maxi drives. When the line is low, the floppy controller is set for maxi drives. When the line is high, the controller is set for mini drives.

The control line is brought to a pad on the BACK side of the p.c. board. On the standard FDC-I, this pad has a trace connecting it to ground, setting the floppy controller for maxi drives. If mini drives are desired, cut the trace between GND and MM. No other jumper is needed as the line has a pull-up resistor connected to +5 volts.

If software is available to take advantage of bothe mini and maxi drives in the same system, do the following: cut the trace between GND and MM. Add a jumper between MM and an unused data line from PIO B. When a "1" is output to that bit of PIO B, the floppy controller will be set for mini drives. When a "0" is output to the PIO B bit, the controller will be set for maxi drives.

BACK of FDC-I



1. For maxi drives (standard), MM is connected to GND.

2. For mini drives, the trace between GND and MM is cut.

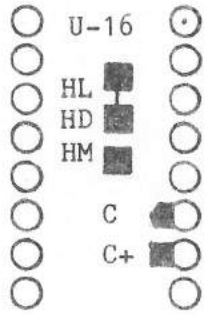
Extended Head Load

The uPD-765 floppy disk controller has a maximum head unload time of 240 mSec. In some applications this will cause an undue amount of head loading and unloading. To increase this head unload time, and reduce the number of head load actions, a 74LS123 (U-16) monostable can be wired into the head drive circuit. With the addition of a 6 volt capacitor, the head unload time is extended. This increases the life of the media and the heads where there would normally be a great deal of head load activity. The following table gives the effective head load time for several different capacitor values:

Capacitor (uF)	Head Load Time (sec)
10	0.5
30	1.4
50	2.3
70	3.2
90	4.1
110	5.0
130	5.9
150	6.8
170	7.7
190	8.6
210	9.5
230	10.4
250	11.3

The time values are approximate (since normally resistor values are ± 10% and capacitor values ± 20%) and are arrived with the following equation: HLT = (45 * C)/(1E03), where C is in microFarads.

BACK of FDC-I



1. Cut the trace between HL and HD.
2. Solder a wire between HD and HM. The solder mask must be scraped from the pads to allow soldering.
3. Solder the desired 6 volt capacitor to "C" and "C+". Be sure the "+" lead of the capacitor connects to "C+".

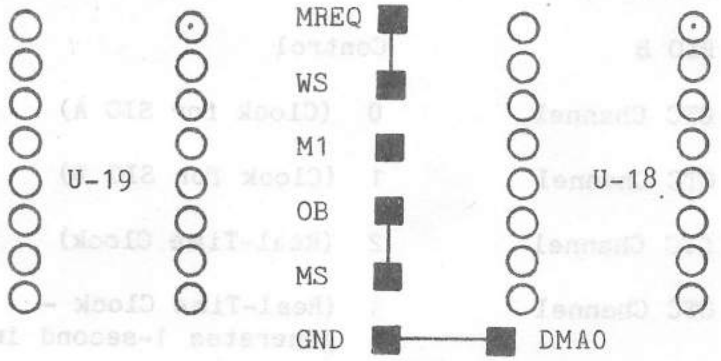
Port Assignments

Address	Function	
00H	SIO A	Data
01	SIO A	Control
02	SIO B	Data
03	SIO B	Control
04	PIO A	Data
05	PIO A	Control
06	PIO B	Data
07	PIO B	Control
08	CTC Channel	0 (Clock for SIO A)
09	CTC Channel	1 (Clock for SIO B)
0A	CTC Channel	2 (Real-Time Clock)
0B	CTC Channel	3 (Real-Time Clock - generates 1-second interrupts)
0C, 0E	FDC Status	
0D, 0F	FDC Data	
10-13	FDC DMA Acknowledge	
14-17	FDC Terminal Count	
18-1F	FDC Wait for Data	

Wait-State Options

The wait-state generator on-board can be set to provide wait states for all memory accesses, or for accesses to on-board memory only. Thus, if fast system memory is used, but a standard 450 ns 2716 for the monitor program, the wait states can be jumpered to occur only when the 2716 is accessed. Also, wait states can be totally eliminated, occur on M1 cycles only, or for all memory request cycles. Normally, to allow programming 2716's, the memory request state is chosen because of the set-up time requirements of the 2716 and programmer. If programming is not desired, but a 450 ns 2716 is used, the M1 wait-state should be enabled. The following drawing shows the location of the appropriate jumper pads on the BACK side of the FDC-I and their function.

BACK of FDC-I

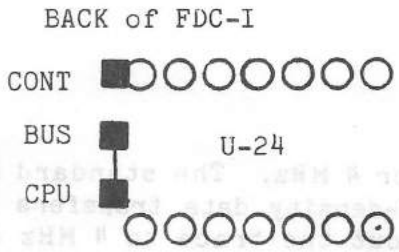


1. WS selects either M1 or MREQ (any memory access, including M1 cycles) wait states. The standard FDC-I has a trace between WS and MREQ to allow EPROM programming. To change to M1 only, cut this trace and solder a wire between WS and M1. To eliminate all wait states, cut the trace to MREQ and solder a wire between WS and pin 14 of U-18.
Note: the solder mask may have to be scraped off the pad to solder to it.

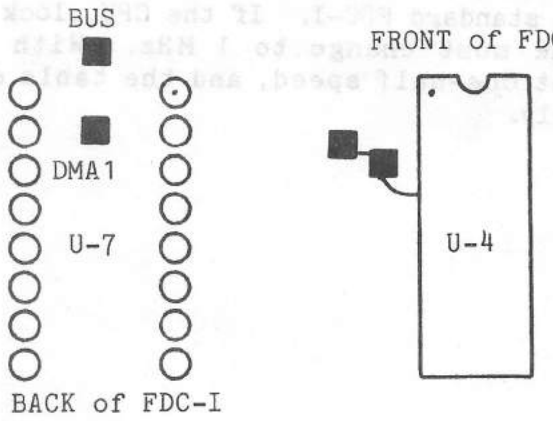
2. MS selects the memory space in which wait states occur. OB generates wait states only for on-board memory accesses. Connecting MS to GND generates wait states for all system memory accesses. The standard FDC-I has a trace between MS and OB for wait states only during on-board memory accesses. To change to all system memory wait states, cut the trace between OB and MS, and solder a wire between MS and GND.

CPU/Intelligent Controller

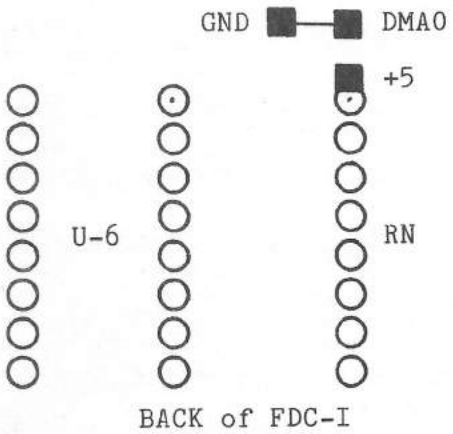
The FDC-I can be strapped as the CPU for a system or as an intelligent peripheral controller. When in the controller mode, the FDC-I transfers information to the system via the DMA control protocol of the S-100 bus standard. The differences between the CPU and the Controller require a change in direction of some of the control lines to the bus. The following jumpers must be changed to convert an FDC-I from the CPU (standard FDC-I) to an intelligent controller:



1. The BUS signal controls the tri-state bus drivers. For the controller mode, cut the trace to CPU and solder a wire between BUS and CONT.



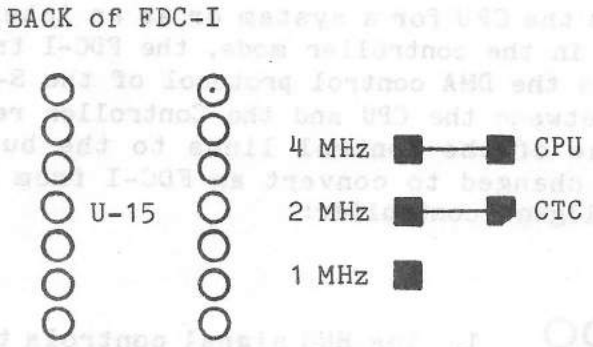
2. The control signals, PWR, PWAIT, PDBIN, PINTE, and PSYNC, are controlled by CDSBL when the FDC-I is in a CPU mode and by the internal BUS signal when a controller mode. To make this change, the trace from pin 4 of U-4 to CDSBL must be cut. Solder a wire from BUS to DMA1.



3. The DMA synchronization clock, 01, and the bus clock must be disabled in the controller mode. To do this, cut the trace to GND, and solder a wire between DMA0 and +5, on pin 1 of the resistor network.

Note: the standard monitor for the FDC-I does not incorporate the routines necessary to operation as an intelligent peripheral controller. Those routines are being developed and will be made available as guidelines for software development.

Clock Jumpers



1. The CPU clock can be set to 2 or 4 MHz. The standard FDC-I is set for 4 MHz because 8" floppy double-density data transfers require this speed. To set the CPU to 2 MHz, cut the trace to 4 MHz and solder a wire to 2 MHz.

2. The CTC channels used to generate the serial port clocks have their inputs connected to 2 MHz in the standard FDC-I. If the CPU clock speed is changed to 2 MHz, this line must change to 1 MHz. With these changes, the CTC will operate at one-half speed, and the table of CTC timing will change correspondingly.

CTC Timing for Serial Ports

Baud Rate	SIO Clock(x16)	CTC Mode	Prescaler	Count	Hex	SS No.*
110	1,760	Timer	16-250kHz	142	8EH	7
150	2,400	Timer	16	104	68H	6
300	4,800	Timer	16	52	34H	5
600	9,600	Counter	--	208	D0H	4
1,200	19,200	Counter	--	104	68H	3
2,400	38,400	Counter	--	52	34H	2
4,800	76,800	Counter	--	26	1AH	1
9,600	153,600	Counter	--	13	0DH	0

CTC Set-Up

1. Timer Mode:
   a. Output 07 to appropriate CTC channel
   b. Output appropriate time count byte

2. Counter Mode:
   a. Output 47H to appropriate CTC channel
   b. Output appropriate time count byte

*The SS number is used with the SS command of the monitor to set the baud rate.

EPROM/RAM Options

The Memory Map PROM, U-25

U-25, a bipolar PROM, "maps" the four memory sockets on-board FDC-I. It selects the appropriate socket for a desired memory address and disables the input bus drivers. Also, it provides the correct chip enable inversion when programming Intel-type 2716 EPROMs. U-25 is a 32X8 PROM. Thus it has 5 address lines (A0-A4) and 8 outputs (D0-D7). The function of each is as follows:

- A0 is connected to AB11 of the address bus.
- A1 is connected to AB12 of the address bus.
- A2 connects to the reset flip-flop, U-24, which will select the first EPROM (U-34) after a reset operation to allow a jump to the monitor.
- A3 connects to the write signal from the CPU. When active, this determines that a write operation is in process. This could be a write to the on-board RAM or a program operation into an EPROM.
- A4 connects to the memory decoder, U-23. This line is active when the 8K block of memory from E000 to FFFF for the standard FDC-I is accessed.

The outputs of U-25 are as follows:

- D0 connects to the chip select line of U-34.
- D1 connects to the chip select line of U-35.
- D2 connects to the chip select line of U-36.
- D3 connects to the chip select line of U-37.
- D4,D5 connect to U-13, the program monostable. Two lines are used to reduce the chance of stray noise causing an erroneous program operation.
- D6 is active high whenever an on-board memory access is in progress. This signal connects to U-22 to disable the data-in buffer to prevent a conflict with the on-board memroy.

The chip select of a RAM IC is active low for either a read or write operation. However, the Intel-type 2716 chip-select line is active low for a read operation, active high for a program operation into the 2716, or low if another 2716 is being programmed. Thus the PROM is an effective means of decoding these many conditions.

Note: any part equivalent to the Signetics 82S123 PROM may be programmed for a particular memory arrangement and inserted into the U-25 position.

The following table gives some memory arrangements which are popular and have become available options for the FDC-I:

Option	Description
M1*	3 programmable EPROMs (U-34,35,36), from E000 to F7FF, and 1 RAM (U-37), from F800 to FFFF. The FDC-I occupies 8K of memory space.
M2	2 programmable EPROMs (U-34,35), from F000 to FFFF, and no other memory on-board. The FDC-I occupies 4K of memory space.
M3	1 non-programmable EPROM (U-34), from F000 to F7FF and no other memory on-board. The FDC-I occupies 2K of memory space.
M4	2 programmable EPROMs (U-34,35), from E000 to EFFF, and 2 RAMS (U-36,37), from F000 to FFFF. The FDC-I occupies 8K of memory space.
M5**	1 non-programmable EPROM (U-34), from F800 to FFFF, and no other memory on-board. The FDC-I occupies 2K of memory space.
M6**	1 non-programmable EPROM (U-34), from F800 to FFFF, and 1 RAM (U-37), from F000 to F7FF. The FDC-I occupies 4K of memory space.

*This is the standard option.
**This option provides the greatest amount of contiguous RAM space.

Note: for those options which do not have RAM on-board FDC-I, some space in system RAM must be set aside for stack and buffer areas. The standard monitors for these options use the 128 bytes just under the location of the monitor EPROM. For example, the M-5 option monitor starts at F800 and uses F780 to F7FF for stack and buffer space. This also means there must be RAM in the system in this area.

Memory Conflicts

Because the U-25 PROM knows when on-board memory accesses occur there is no need to worry about external memory conflicts. No deselection of a 64K RAM board is necessary, no matter which option of memory space is used with FDC-I.

Memory Map for M1 PROM

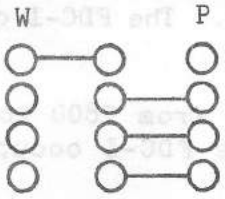
The memory map following this section illustrates the logic levels into and out of the U-25 PROM. It can be seen that different operations result in different chip select signals to the EPROMs. This also implies that if a RAM is substituted for an EPROM, the map must change to accomodate the different chip-select protocol.

Jumper changes to accommodate RAM or EPROM

Due to differences between EPROM and RAM ICs, and differences between 1K and 2K RAMs, jumper pads on FDC-I allow the necessary connections to support each device. Basically the EPROM requires a programming voltage instead of a write strobe, while the 2K RAM requires a connection to address line AB10 instead of +5 volts. Position U-34 is permanently wired as an EPROM location. On the standard FDC-I, positions U-35 and U-36 are wired as EPROMs and position U-37 is wired for a 1K RAM, the MOSTEK MK 4118.

To change an EPROM location to a RAM location, cut the trace to "P" and solder a wire to "W". To accomodate the 1K RAM, cut the trace to AB10 and solder a wire to +5 volts (a 2K RAM does not require this last change).

FRONT of FDC-I, right side



U-37 RAM
U-36 EPROM
U-35 EPROM
U-34 EPROM

RAM/EPROM jumpers
To change a particular position, cut the trace and connect the center donut to P for an EPROM, or W for a RAM.

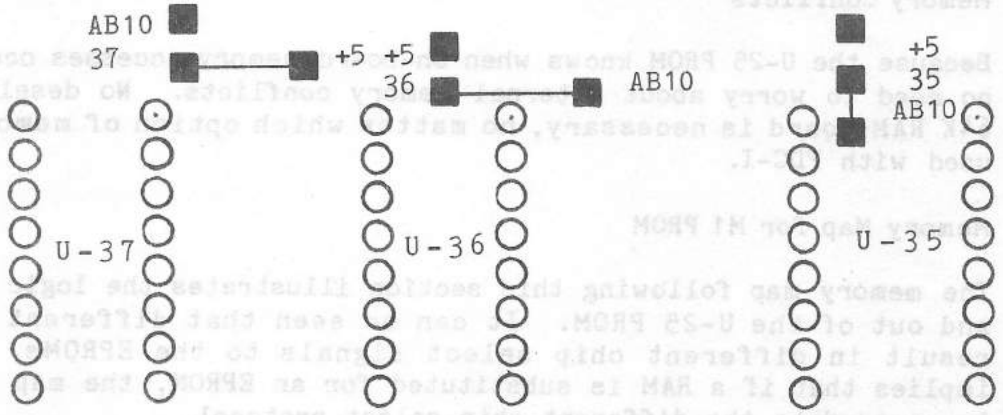
(Standard board traces shown)

1K/2K RAM address-line jumpers

Position U-37 is set for a 1K RAM, MOSTEK MK 4118. To change to a 2K RAM, or EPROM, cut the trace from "37" to "+5", and connect "37" to "AB10". Positions U-35 and U-36 don't need address-line changes for 2K RAMs. To accomodate the 1K MOSTEK device, cut the trace to "AB10" and connect to "+5".

To change a RAM position to EPROM, remember to connect AB10 and change from "W" to "P".

BACK of FDC-I



M-1 Memory Control PROM map. Standard FDC-I configuration: 3 EPROM, 1 RAM

PROM Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
A0 AB-11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A1 AB-12	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
A2 $\overline{JMP}$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
A3 $\overline{WR}$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
A4 $\overline{BS}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PROM Outputs																																
D0 ROM0 U-34	1	1	1	1	1	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
D1 ROM1 U-35	1	1	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D2 ROM2 U-36	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D3 RAM U-37	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D4 PGM	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5 PGM	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6 MEM	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
DATA	0F	0F	0F	0F	79	7A	7C	47	4E	4E	4E	4E	4E	4D	4B	47	0F	0F	0F	0F	0F	0F	0F	0F	4E	4E	4E	4E	0F	0F	0F	0F
Operation	Inactive				Write				Read/Jump				Read				Inactive								Jump				Inactive			

Explanation of abbreviations

AB = on-board addresses; $\overline{JMP}$ is the jump signal from U-24; $\overline{WR}$ is the write signal from the CPU; $\overline{BS}$ is the board-select signal from U-23; PGM are signals that trigger the program monostable, U-13; MEM is the signal that indicates an on-board memory access.

Status PROM

U-12 is a bipolar PROM programmed to generate the S-100 status signals from the equivalent operation of the Z-80A CPU. These signals are then used by system peripherals to coordinate their activities. The "S-2 Status PROM Map" following this section lists the outputs of U-12 for each combination of inputs. Some input combinations will never occur, but are listed to provide complete program information. The abbrebviations used and their meaning are explained below.

The address lines of U-12 connect to the CPU status lines:

A0	RD	This is the read strobe and is active when data is sent to the CPU. During an interrupt acknowledge cycle, this is not active.
A1	M1	This line is active during the first read operation of a new instruction (fetch) and during an interrupt acknowledge operation.
A2	MREQ	Whenever the CPU accesses memory this line is active. Also during a refresh operation this line indicates to the memory controller that a stable refresh address is available from the Z-80 CPU.
A3	IORQ	Whenever Input/Output ports are accessed this line is active. When IORQ and M1 are active together, this signals an interrupt acknowledge operation is in process. See "Theory of Operation" section "Interrupts" for further details.
A4	RFSH	This signal indicates to a dynamic memory controller that a refresh operation can occur. When RFSH and MREQ are both active, a stable refresh address is available from the CPU address lines.

The outputs of U-12 are:

D0	PSYNC	This output triggers a monostable, U-13, whose pulse output indicates to the peripherals on the bus that stable status and address information is available. PSYNC is generated for every operation.
D1	RFSH	Because all dynamic memory boards on the market have a built-in refresh address counter, this signal is active as long as RFSH from the CPU is active. Memory controllers can execute a refresh operation during this time.

The following are all S-100 status signals:

D2 SWO	This signal is active low for every write operation.
D3 SINTA	This signal indicates that the CPU is responding to an interrupt request. While this signal is active, priority of conflicting devices is determined and in Mode 2 a vector address is placed on the data lines by the highest priority device.
D4 SMEMR	Whenever the CPU is reading from memory, this signal is active.
D5 SINP	This signal indicates that the CPU is inputting data from an I/O port.
D6 SOUT	This signal indicates that the CPU is outputting data to an I/O port.
D7 SM1	This signal indicates the first read operation of a new instruction.

Status PROM Map

The map following this section illustrates the outputs for each combination of inputs. A 0 indicates a low logic level, and a 1 indicates a high logic level. The "Data" line represents the program burned into the PROM in hexadecimal notation. This program reproduces the "S-100 Standard" status exactly.

S-2 Status PROM

March 1980

Operation	Address	A0 RD	A1 M1	A2 MREQ	A3 IORQ	A4 RFSH	Outputs D0 PSYNC	D1 RFSH	D2 SWO	D3 SINTA	D4 SMEMR	D5 SINP	D6 SOUT	D7 SM1	Data
	00	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	01	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	02	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	03	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	04	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	05	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	06	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	07	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	08	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	09	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	0A	0	1	0	1	0	0	1	1	0	0	0	0	0	06
Refresh	0B	0	1	0	1	0	0	0	1	0	0	0	0	0	04
	0C	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	0D	0	1	0	1	0	0	1	1	0	0	0	0	0	06
	0E	0	1	0	1	0	0	1	1	0	0	0	0	0	06
Refresh	0F	0	1	0	1	0	0	0	1	0	0	0	0	0	04
	10	0	1	0	1	1	0	1	1	0	0	0	0	0	06
	11	0	1	0	1	1	0	1	1	0	0	0	0	0	06
	12	0	1	0	1	1	0	1	1	0	0	0	0	0	06
	13	0	1	0	1	1	0	1	1	0	0	0	0	0	06
	14	0	1	0	1	1	0	1	1	0	0	0	0	0	06
SINTA	15	1	0	1	0	1	0	1	1	1	0	0	0	0	8F
SINP	16	1	0	1	0	1	0	1	1	0	0	1	0	0	27
SOUT	17	1	0	1	0	1	0	1	0	0	0	0	1	0	43
Fetch	18	1	0	1	0	1	0	1	1	0	0	0	0	0	97
Fetch	19	1	0	1	0	1	0	1	1	0	0	0	0	0	97
Read	1A	1	0	1	0	1	0	1	1	0	0	0	0	0	17
Write	1B	0	0	1	0	1	0	1	0	0	0	0	0	0	03
	1C	0	0	0	0	1	1	1	1	0	0	0	0	0	06
Fetch	1D	1	0	0	0	1	1	1	1	0	0	0	0	0	87
	1E	0	1	0	0	1	1	1	1	0	0	0	0	0	06
	1F	0	0	0	0	1	1	1	1	0	0	0	0	0	06

Theory of Operation

Central Processor Operations

The heart of FDC-I is a 4 MHz Z80A. It provides the intelligence to operate the on-board peripherals and to provide the information interchange to the S-100 bus. Connections to the bus are made through tri-state buffers and control logic to provide the correct timing signals and status signals to operate other peripherals within the microcomputer. Data to and from the bus is buffered in octal tri-state buffers. The data input buffer is enabled and disabled by U-22, a bipolar PROM, which decodes the status signals from the CPU and the memory and peripheral select gates to determine whether the next operation will be data from on- or off-board the FDC-I. The data-out drivers are enabled by the on-board bus enable signal which also controls the address buffers. Because of U-22, conflicts between external peripherals and memory are avoided. Thus memory in the system can conflict with the memory space of FDC-I without problem. This means that deselection of external memory is not necessary. Status to the bus is controlled by U-12, another PROM which is programmed to decode the control signals of the Z80A and provides the necessary status signals to the bus at appropriate times in the CPU operation. This is a tri-state Device and is enabled whenever FDC-I talks to the bus.

Because the Z80 does not provide a synchronization signal as did the 8080, U-12, the status decoder PROM, has an output which triggers one-half of U-13, a dual monostable. This monostable output provides a pulse equivalent in timing to what an 8080 processor would provide as PSYNC. For those systems which require a memory write signal, an OR gate provides the necessary write signal whenever the memory request and the write line of the Z80 are active. The address lines from the Z80 are buffered in two octal tri-state bus drivers which are enabled whenever FDC-I accesses the S-100 bus.

Reset Jump Circuit

The reset-jump and power-on-clear jump circuitry is enabled whenever PRESET, pin 75 on the S-100 bus, is active. When PRESET is active, two NAND gates cross-connected as a set-reset flip-flop are enabled which cause a signal to be generated on FDC-I which activates the internal EPROM in position P0. This EPROM has been programmed with a jump instruction to the memory on board FDC-I. This absolute jump to the EPROM address must be programmed into the first 3 locations of the EPROM.

When the processor jumps to the memory address at which FDC-I is sitting, this reset jump circuitry is reset and returns to its idle condition.

Memory Decoding

U-23, a 3-line to 8-line decoder, is connected to the Z80 status signals RFSH and MREQ and determines when the Z80 is accessing memory. The three select lines of U-23 are connected to the three high order address lines of the Z80. Thus, U-23 will decode blocks of memory, each block being 8k bytes in size, from 0 - E000. One of the select lines from U-23 is connected to U-25, a bipolar PROM. U-25 is connected to address lines 11 and 12 of the Z80 and is used to select 2k byte blocks of memory space within the 8k block selected by U-23. U-25 is connected to the write signal of the Z80 and to the jump signal from the reset-jump circuit. U-25 is programmed such that the chip select lines to the EPROM and RAM on board FDC-I are enabled properly, depending on the operation; in a normal read mode, the select line, when active, will go low; however, when programming an EPROM, the select line is normally held low on the EPROM that is not being programmed and pulsed high for the EPROM that will be programmed. Thus, the need for a PROM to provide the complex logic necessary to determine the type of operation and the state of the output signals corresponding to that operation. The select line to the RAM in the FDC-I is active low during read or write and inactive high otherwise. Thus, U-23 must be programmed according to the amount of EPROM and the amount of RAM which will be on FDC-I and is thus unique for each different variation. The standard PROM supplied assumes that there are three EPROMs and one RAM on board FDC-I. U-25 also provides an output which activates one-half of U-13 to provide a programming power pulse to the selected EPROM. This output is only active for those EPROM slots on board when they are selected in a write mode. For information about the program contained in U-25, see the section entitled "Memory Address PROM".

EPROM Programming

The necessary power for programming EPROMs is supplied by providing +25.5 $\pm$ 0.7 volts to the two pins adjacent to U-51. The + pin is for the 25.5 volts and the - pin for the ground connection. This voltage is applied whenever EPROM programming is desired. Software then writes the desired programmed byte to the appropriate EPROM. PROM U-25 provides the appropriate chip select level to the EPROM and U-13 is triggered by U-25 and provides a pulse on the output which halts the processor holding the data and address lines constant and energizes the transistor network which provides the programming voltage to the EPROMs. This task, in order to retain dynamic memory, must occur for a short enough period of time to prevent data from dissipating in a dynamic RAM. Therefore, the pulse is repeated 256 times until the proper programming time has been attained. The recommended programming time for each word in a 2716 is 50 milliseconds. Therefore, the programming time per pulse is approximately 200 microseconds. With 2716's any byte may be programmed; the entire EPROM need not be programmed at one time. Selected areas can be programmed at one time and later, additional areas programmed. The normal state of an erased EPROM is all high in the memory cells. When programmed, the cells become active low.

I/O Select Logic

U-20 and U-33 provide the selection at appropriate addresses for the on-board I/O ports. U-20 is enabled when IORQ is active low and M1 is inactive high. The three high order port address lines, that is A7, A6, A5, select one of eight outputs on U-20. One of those outputs is connected to U-33, a 3-line to 8-line decoder, to enable that IC when the appropriate I/O address is present at the processor address lines. U-33 further decodes the address lines into eight separate chip select lines. Six of those chip select lines are used on board. The first four chip select lines select the SIO, the PIO, the CTC and the FDC data port. The fifth chip select line is used to reset the floppy controller IC by giving it a terminal count, and the sixth line is used to halt the CPU when waiting for floppy disk data transfer. No wait states are generated other than the normal Z80-inserted wait state into the I/O request.

Clock Generation

The on-board 16 MHz crystal-controlled oscillator provides the basic timing reference for all board timing. This 16 MHz clock is divided by U-15 to form those frequencies essential to operation of the Z80A processor, the system bus clock and the clock frequencies required for floppy control operations. The 4 MHz clock from U-15 is further buffered by three sections of U-29 which have two 510 ohm pull-up resistors on the output to provide the levels necessary to drive the Z80A processor. The 2 MHz system bus clock line is driven via a tri-state buffer which is enabled whenever FDC-I needs to access the system bus. The Z80 CTC provides an on-board clock for the two SIO serial ports as well as a real time clock for operating system requirements. The triggers of the first three sections of the CTC, which is composed of four separate counter timers, are connected to the 2 MHz line from U-15. This provides a lower input frequency which yields a greater range of lower frequencies from the CTC timers. The fourth section of the CTC is normally used as a one-second interrupt clock to provide operating system timing marks. Jumpers adjacent to U-15 provide user selection of frequencies essential to system operation. The Z80A clock frequency can be changed, as can the 2MHz bus clock.

SIO and the Serial Ports

The Z80 SIO is used to generate two entirely independent serial ports. Both serial ports incorporate all the handshaking lines required by an RS232 data interconnection device. As configured by the standard software, SIO A input is interrupt driven. SIO B and the SIO A output operate in a polled-status mode. Each channel of the SIO is driven by an independent section of the CTC. This means that baud rates for the two channels can be independently selected. In fact, the baud rates may range anywhere from 45 baud up to 9600 baud. These frequencies are determined during initialization of the CTC channel which is performed by the monitor software or through the use of the SS command in the standard monitor. The data lines to and from the SIO channels are buffered by RS-232-C level translators. These buffers are also inherently protected from short circuits on the external lines.

Miscellaneous Control Functions

FDC-I has a power-on-clear timer which is enabled whenever power is restored after a period of being off. This line is output to the bus as a power-on-clear signal to the rest of the system and also activates the PRESET line of the bus. This action causes FDC-I to automatically do a reset-jump to the internal monitor on-board FDC-I.

When FDC-I is configured to act as a central processor in a microcomputer system, other peripherals may request use of the bus via Direct Memory Access. The Bus Request and Bus Acknowledge signals of the Z-80A CPU are brought to the bus signals PHLD and PHLDA. When a peripheral such as a hard-disk controller needs to transfer information directly to memory, it pulls PHLD low. The Z-80A CPU will release the bus when it finishes its present operation. At that time it will acknowledge the request. This releases the on-board bus signal tri-stating all but the control signals, and raises PHLDA high. The peripheral will then place its control signals on the bus and pull CDSBL low, tri-stating the FDC-I control signals. When the peripheral is finished, the process is reversed and the CPU begins operation at the point where it was interrupted.

Automatic wait circuitry causes a wait state to be generated for every memory request of the Z80A processor. This has two advantages. One, it allows 450 nanosecond random access memory to be used, even though the processor is running at a 4 MHz speed and two, it allows some of the internal functions a greater amount of time to process such as the EPROM programmer section. This wait state can be jumpered to provide a wait state only on M1 cycles or alternatively, may be jumpered to provide no wait states at all. The memory wait state is generated by U-19, a dual J-K flip-flop. One half of U-19 has its K input connected to either the M1 or the MREQ line of the Z80A. Because the clock input of U-19 is connected to the processor clock, this information will be clocked into U-19 on the positive transition of the processor clock. The Q output of this half of U-19 will then go low which will cause the second half of U-19 to go low on the next positive processor clock cycle. This process will then clear the first half of U-19 and the wait state will have been

generated. Additional wait states are required by the floppy controller during data transfers to and from the floppy disk drive. Additionally, the EPROM programmer will generate a wait state when it is active. External devices holding the wait line low on the bus will indefinitely halt the processor until that line is released.

Floppy Disk Controller Operation

The central element of the floppy disk control portion of FDC-I is the NEC uPD-765 floppy disk control chip. This IC provides all the support necessary to control a flexible disk drive. The uPD-765 provides the IBM compatible formatting and encoding of data in single and double density formats to the disk drive and provides the decoding essential for reading both single and double density information. This IC has the capability of simultaneously seeking on four drives because it time multiplexes accesses to each drive within the system. The support circuitry on FDC-I for the uPD-765 includes demultiplexing circuitry because the NEC IC uses several of its pins for multiple functions. U-43 decodes the two unit select lines of the 765 to provide the four unique select lines which will drive up to four disk drives. U-45, which is an octal tri-state buffer, is separated into two halves of four buffers. One half is controlled by the read-write/seek line from the uPD-765. In the seek mode, the direction, the two side and the step function from the uPD-765 are enabled. In the read-write mode, the write protect, the low current and the fault line as well as the fault reset line are enabled. Because the disk drives are terminated in resistor networks, about 25 milliamps of drive are required per line. Therefore, all signals to the drive are buffered by tri-state buffers capable of sinking 25 mA. All lines from the disk drive are terminated in 220/330 ohm resistive networks and buffered via low power Schottky Schmidt trigger input buffers. The read data from the disk drive must be phase-locked to a stable signal. This is accomplished by U-15, 17, 52, 53, and 54. The data from the drive are phase detected and a voltage-controlled oscillator composed of two sections of U-54 is locked to the data frequency. U-28 is a 2-line to 1 encoder. This is required because of the different frequencies required by single density and double density operation. For example, in double density operation, the read window to the uPD-765 must be 500 kHz for an 8" drive. In single density operation, that frequency must be halved to 250 kHz. Correspondingly, during a write operation, the write clock to the 765 must be 1 MHz in double density and 1/2 MHz in single density when driving an 8-inch disk drive.

Because data transfer to the floppy disk drive is faster than the processor could normally handle an I/O request in a polled status mode, the Z80 executes a block I/O move instruction. Because the timing of the floppy disk drive may not be the same as the I/O instruction timing, the processor is put into a wait state until the floppy disk control IC requires this data. This wait function is generated as follows: The CPU performs an I/O operation to port 18H. This enables one half of U-40, a 2-line to 4-line decoder. The A and B inputs to U-40 connect to the DRQ and inverted INT outputs of U-41 respectively. Until U-41 is ready to transfer data or interrupt, the DRQ and INT outputs are low. This causes the "2" output of U-40 to be low, halting the CPU. When U-

41 is ready, it outputs a 1 to DRQ (DMA request). This causes the "2" output to return to its inactive state, which is high. The CPU is released and executes its next instruction, one loop through a block I/O instruction. Then the CPU once again executes an I/O operation to port 18H and waits until U-41 is again ready. This type of operation is required because too much time would be required to first check the status of U-41, then execute a data transfer. If U-41 encounters an error, its interrupt (INT) goes high, releasing the CPU wait.

Parallel Port

The parallel port consists primarily of the Z80 PIO. Port A is used as an 8 bit input, output or bidirectional port. The four handshaking lines of the PIO are used with port A. Normally, port A is configured as an input for such parallel items as a keyboard. Under software control, port A can be configured as an output or as a bidirectional port where input data and output data as well as direction are controlled by the four handshaking lines.

Port B of the PIO is used in a bit mode to provide various inputs and outputs while running FDC-I. The lines of port B are used to control the tri-state buffers which connect to the S-100 bus as well as to provide independent lines for control of the vector interrupt where FDC-I is used as an intelligent controller and for handshaking purposes with other devices in the system which may want to talk to FDC-I. In this mode, the port line acts as a wake-up line to get attention.

Interrupts

The FDC-I CPU is configured in interrupt mode 2. In this mode, a requesting peripheral generates an interrupt and when that interrupt is acknowledged, the CPU expects the peripheral device to place an 8 bit address vector on the data lines. The CPU then adds this 8 bit vector with another 8 bit register internal to the CPU to form a 16 bit absolute memory address. This address points to a 2 byte location in memory which contains the absolute address of the desired subroutine to service the interrupt. In the case of the Z80, SIO, PIO and CTC, the necessary interrupt vectors are loaded to internal registers during initialization. For the case of the floppy controller IC, the interrupt vector is simply composed of that vector formed by the pull-up resistors on the data lines, an FF. Thus, the interrupt table for the floppy controller must begin on a boundary of FF. The Z80 peripheral IC's normally begin on an even memory location because bit 0 is always a 0 during their interrupt response. When a device external to the CPU requests an interrupt, the external device must provide an interrupt vector on the data bus when interrupt acknowledge status line goes active high. The interrupt enable function is controlled by the SEEK/RW line of the uPD-765. Thus, when a floppy disk data transfer is in progress, interrupts other than the floppy controller are disabled. The Z80 peripheral IC's are series connected to provide priority interrupts. The last peripheral in the chain, namely the PIO, provides an interrupt enable signal to the S-100 bus. When this line is high, interrupts are enabled for external requests. When this line is low, external devices

must be prevented from generating a response to an interrupt acknowledge signal. The vector that external devices place on the bus, when combined with the internal high order vector of the CPU, must point to a location in memory which provides the absolute address of the subroutine used for servicing that particular interrupt. In the standard Teletek monitor provided on board, there are areas in the interrupt table which are availble for the user. If more area is required, then the monitor program must be changed or the interrupt table moved to another PROM. To do so would simply require a change in the vector stored in the interrupt register of the CPU to point to the new interrupt table location.

Disk Data Encoding

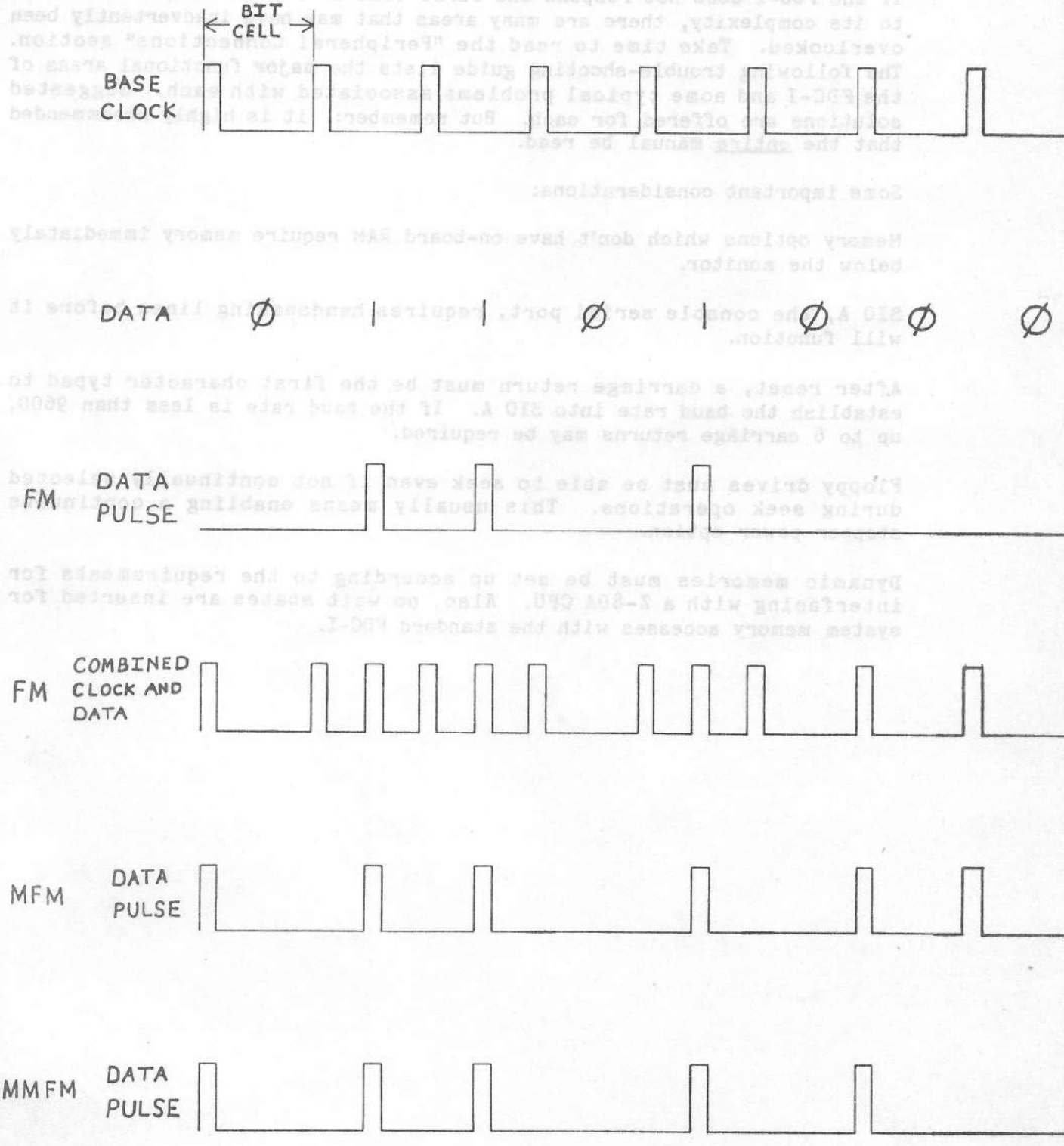
Physically, double density disk drives do not differ significantly from their single density counterparts. Improvements in double density record and playback heads and changes in mechanics often provide less expensive and more durable drives. These changes are minor compared to the differences in reading and writing functions. Figure 1 reviews encoding methods used in single and double density. The standard recording formats are FM for frequency modulation, MFM for modified frequency modulation (double density), and MMFM for modified modified frequency modulation which is a refinement of MFM. Line 1 of Figure 1 indicates the basic clock frequency which designates the bit cell in which information will be passed. The next line illustrates a sample of information. The next line shows the pulses which generate that information in a single- density FM format. Notice that information actually sent to and received from the drive is a combination of the basic clock frequency and data pulses. Refer to the next line which is MFM.

Here only the data pulses will be sent to the drive and their orientation within the bit cell determines the value of that particular data pulse, be it a 1 or a 0. Every 0 is represented by a data pulse that coincides with the basic clock frequency. Every 1 is represented by a pulse that occurs midway between two clock pulses. Thus, when the data pulse occurs in the middle of a bit cell, it is a 1; when it occurs in the beginning of a bit cell, it is a 0. Look at the next line which represents MMFM.

This is a slight refinement of MFM and in this instance, the data pulses once again represent 1's and 0's via their placement within the bit cell. However, the rules change slightly. If the preceding data pulse was a 0 and the present data is a 0, then the data pulse will occur. If the last data pulse was a 1 and the present data is a 0, the present data pulse does not appear. If the last data pulse was a 1 and the present data is a 1, that data pulse appears. Every time there is a 1, a data pulse will appear in the middle of a bit cell. But whether or not a 0 data pulse occurs depends on the preceding data. Note that the density of data pulses for MFM is almost exactly one-half the density of data pulses for FM. Thus, for the same density of pulses on the diskette, MFM will record twice as much information as FM. MMFM has slightly less dense data pulses than MFM, but its complexity of encoding and decoding outweighs the slight advantage it might enjoy due to slightly less density.

The basic clock frequency for FM encoding is 250 KHz for an 8-inch diskette. When we delete the clock and leave only the data pulses in MFM, that clock rate changes to 500 KHz. The MFM data transfer rate is twice as fast as FM. The density and the speed are both doubled which means that twice as much information can be stored in the same physical space and manipulated twice as fast.

FIGURE 1 FLOPPY DISK DATA ENCODING



In Case of Trouble

If the FDC-I does not respond the first time it's connected, relax. Due to its complexity, there are many areas that may have inadvertently been overlooked. Take time to read the "Peripheral Connections" section. The following trouble-shooting guide lists the major functional areas of the FDC-I and some typical problems associated with each. Suggested solutions are offered for each. But remember: it is highly recommended that the entire manual be read.

Some important considerations:

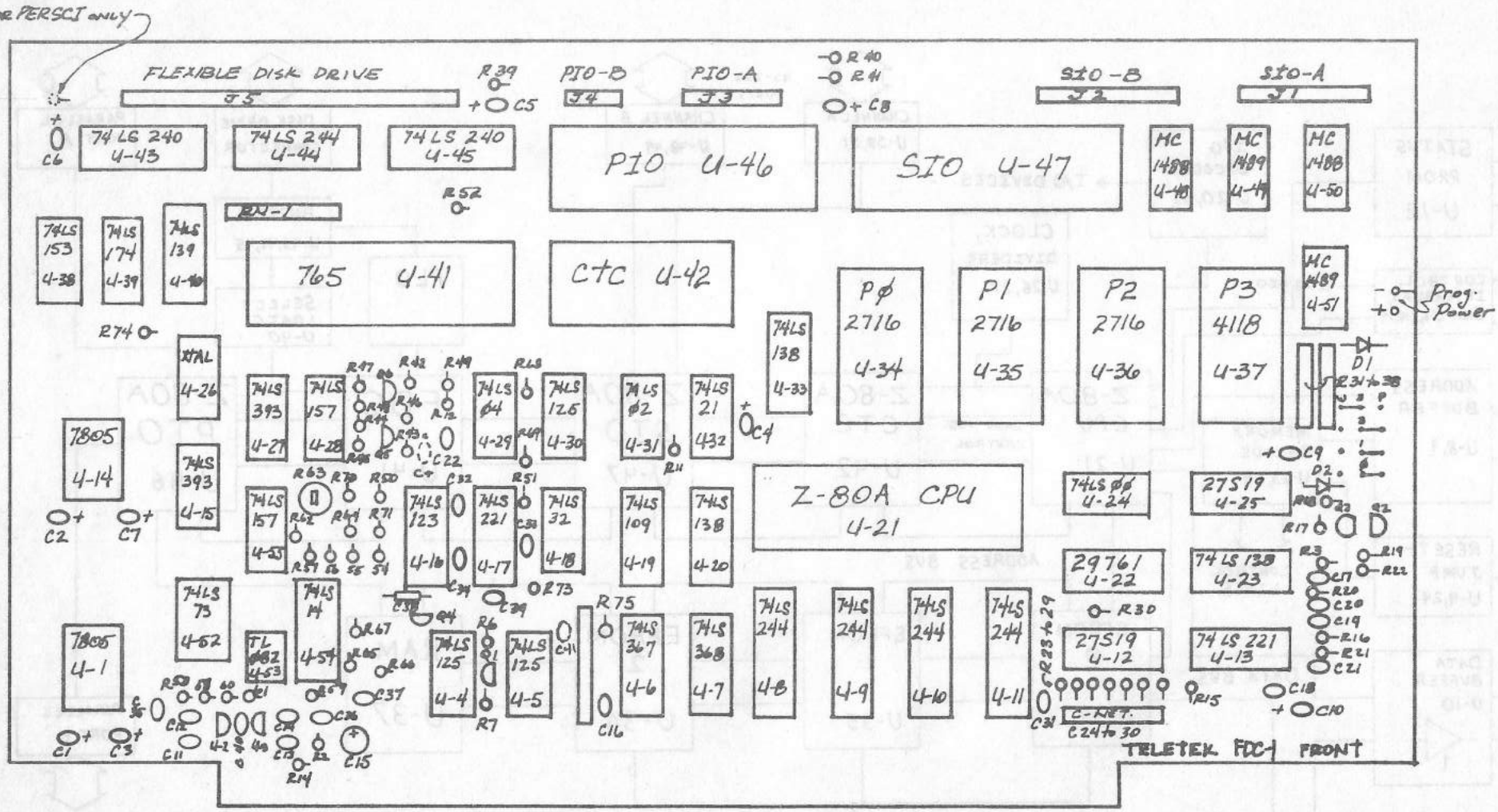
Memory options which don't have on-board RAM require memory immediately below the monitor.

SIO A, the console serial port, requires handshaking lines before it will function.

After reset, a carriage return must be the first character typed to establish the baud rate into SIO A. If the baud rate is less than 9600, up to 6 carriage returns may be required.

Floppy drives must be able to seek even if not continually selected during seek operations. This usually means enabling a continuous stepper power option.

Dynamic memories must be set up according to the requirements for interfacing with a Z-80A CPU. Also, no wait states are inserted for system memory accesses with the standard FDC-I.



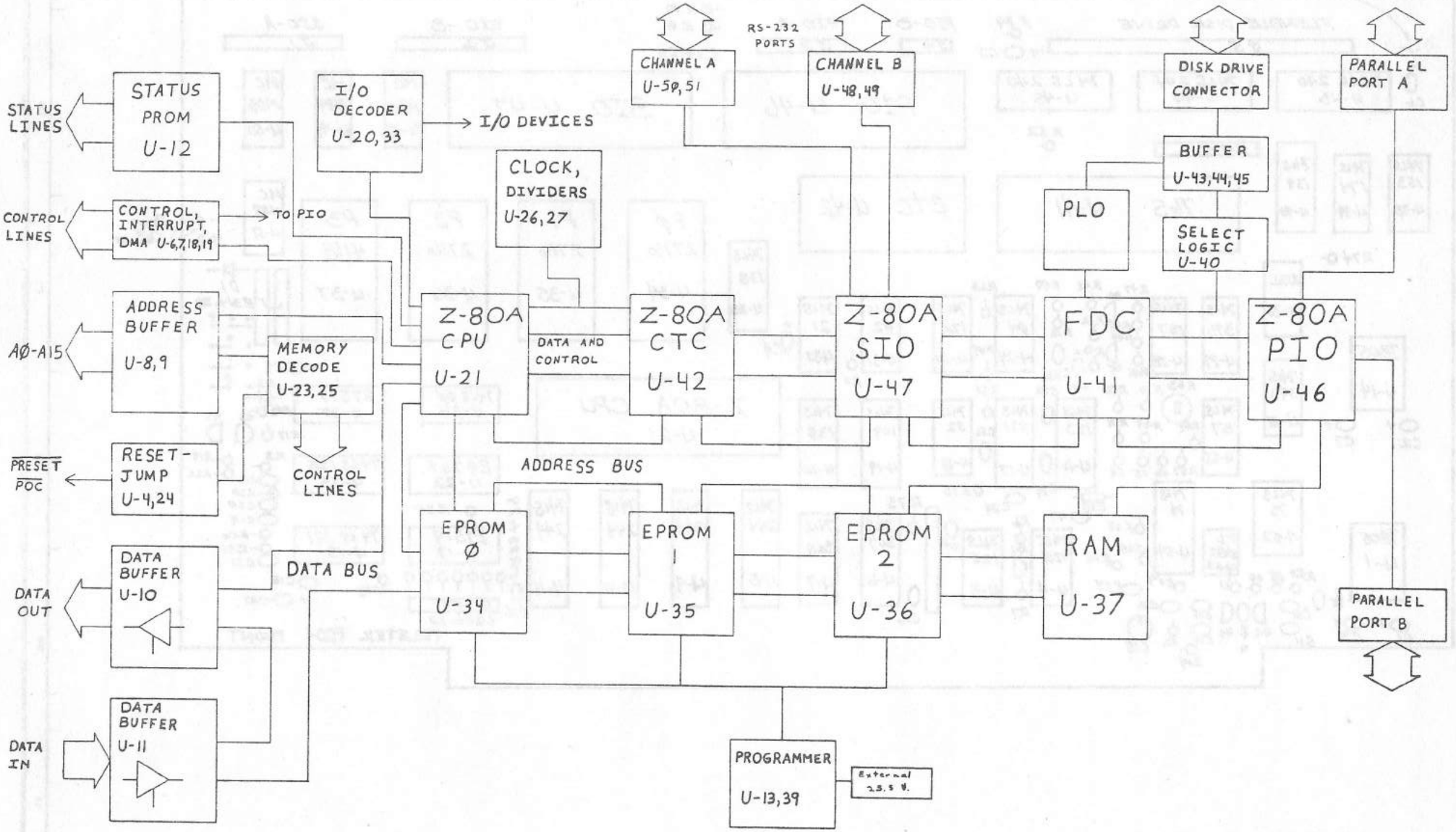
FDC-I User Manual, Appendix A

8 7 6 5 4 3 2 1

TELETEK FDC-I BOARD LAYOUT

August 7, 1980 KB

BLOCK DIAGRAM



FDC-I User Manual, Appendix B

TELETEK FDC-I
AUG. 7, 1980

FDC-I User Manual, Appendix C

Contents

Disk Drive Interfacing

In controlling a disk drive from FDC-I, proper connections must be made to the disk drive in order for it to be operational. The drive options must be configured as outlined in the appropriate manufacturer's section following this introduction. Particularly important is the fact that the uPD-765 continuously polls all drives in the system to keep track of their status. With some drives this will interfere with their seek function (positioning of the head). Thus, most drives will have a stepper motor enable option, or simultanious seek option, that powers the stepper motor continuously, rather than just when the drive is selected. If the drive won't read initially, check for this option.

Drive interfacing deals with the proper connection of functional signals and the satisfying of electrical and mechanical requirements.

To help ease the shock of transition from the interchanging of various disk drives to other host controllers, a standard known as ANSI was developed which standardized the means of intercommunication between disk drive and host controller by specifying power requirements and voltage levels, edge connector and cable specifications, and specific pin numbers of the connector to particular functional signals.

ANSI Standards

Functional signals assigned to specific pin numbers of the connector are shown below for a 5.25-inch disk drive and an 8-inch disk drive.

ANSI Standard for 5.25 Inch Drive

Signal Pin No.	Ground Pin No.	Signal
2	1	Not assigned (Head load)
4	3	In use control
6	5	Drive select 3 (Ready)
8	7	Index/sector
10	9	Drive select 0
12	11	Drive select 1
14	13	Drive select 2
16	15	Motor on
18	17	Direction select
20	19	Step
22	21	Composite write data
24	23	Write gate
26	25	Track 0
28	27	Write protected
30	29	Composite read data
32	31	Side one select
34	33	Disk change (Drive select 3)

ANSI Standard for 8-Inch Drive

Signal Pin No.	Ground Pin No.	Signal
2	1	Head current switch
4	3	Not assigned
6	5	Not assigned
8	7	Drive busy
10	9	Two-sided
12	11	Disk change
14	13	Side one select
16	15	In use control
18	17	Head load
20	19	Index
22	21	Drive ready
24	23	Sector
26	25	Drive select 0
28	27	Drive select 1
30	29	Drive select 2
32	31	Drive select 3
34	33	Direction select
36	35	Step
38	37	Composite write data
40	39	Write gate
42	41	Track 0
44	43	Write protected
46	45	Composite read data
48	47	Separated read data
50	49	Separated read clock

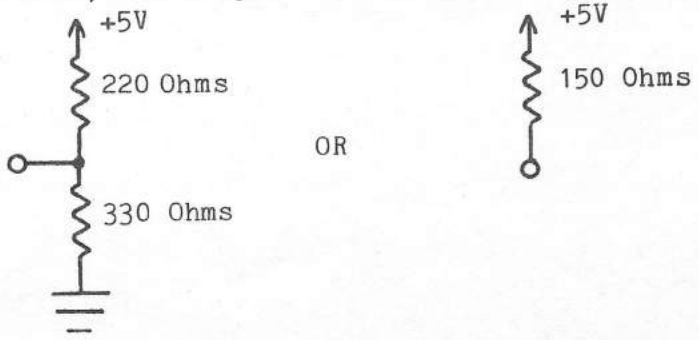
Electrical

1. Multi Drop Bus: Multiple drives may be connected to the same host controller as shown in Figure 1. Only one drive is logically connected to the interface at a time.

2. Voltage Levels (as measured at the driver)

Logical true	Active low	+0V to +0.4V
Logical false	Active high	+2.4V to +5.5V

3. Termination: Signal lines shall be terminated by one of the two resistive networks illustrated below, whether the termination occurs at the drive or the host, but only at the terminal point of a signal.



4. Signal Drivers: The signal drivers should have open collector output stages capable of sinking a minimum of 40mA at logical true (low) level, with maximum voltage of 0.4V as measured at the driver output.

5. Signal Receivers: The signal receivers should not unduly load the multi drop bus and should not require more than 40uA current from the driver at input high (2.4V) nor supply more than 1.6mA to a current sink at input low (0.4V) level.

Interconnecting Cable

Conductor Size

Copper
AWG #30 or larger for solid conductor
AWG #28 or larger for stranded conductor

Non-copper
Sufficient size as to yield a dc resistance not to exceed 110 Ohms per 1000 ft. per conductor.

Stray capacitance
Capacitance between one wire in a cable and all others in the cable with all others connected to ground shall not exceed 40pF/ft. and the value shall be reasonably uniform over the length of the cable.

Mutual pair capacitance
Capacitance between one wire of the pair to the other shall not exceed 20pF/ft. and the value should be reasonably uniform over the length of the cable.

Use of Mini-Floppy Drives

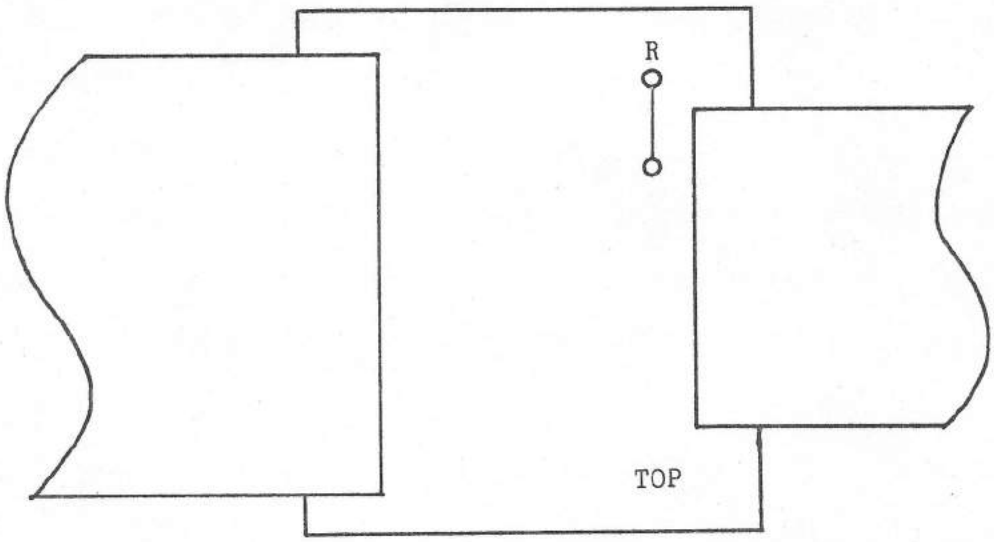
The use of any mini-floppy drive requires the following:

1. Changing the clock speed to the floppy disk controller chip. See the section entitled "Mini/Maxi Floppy Selection" in the manual for more details.

2. Enabling several options in the FDC-I monitor. See the section entitled "Monitor Options" in Appendix E for more details.

3. An adapter p.c. board and 34 pin edge connectors which plugs into the 50 pin connector on the FDC-I. This adapter provides all the signal line connections required by the mini-floppy drives. Contact the factory for delivery information.

4. Some mini-floppy drives do not provide a "Ready" signal. On these drives it is neccessary to connect the "R" jumper on the adapter board as shown in the following diagram:



Micropolis 1015 Disk Drive

Required drive configuration:

1. DS1-4 Select appropiate drive address
2. HDLD Enable this option

Install the termination network only in the last drive in the daisy chain.

Each drive requires 12 volts at 1.3A and 5 volts at 0.5A.

Required Pre-write Compensation: 250 ns.

Shugart SA-400 Disk Drive

Required drive configuration:

1.	HL	Jumper
2.	DS-1,2,3	Select appropiate drive address
3.	MX	Open
4.	MH	Open

Install the termination network only in the last drive in the daisy chain.

Each drive requires 12 volts at 1.8A and 5 volts at 0.7A.

Required Pre-write Compensation: none. (If pre-write compensation is wanted use 250 ns.)

Note: requires the "R" jumper connected on the adapter board.

MPI B-51/52 Disk Drive

Required drive configuration:

1.	T1	Jumper
2.	T2-T4	Select appropiate drive address
3.	T5	Open
4.	T6	Open
5.	T7	Open

Install the termination network only in the last drive in the daisy chain.

Each drive requires 12 volts at 1.5A and 5 volts at 0.7A.

Required Pre-write Compensation: none.

Note: The adapter board from Teletek does not provide Drive Select 3 on pin 6 of the 34 pin connector. Thus this line must be disconnected going into the drive. Also, the "R" jumper has to be connected on the adapter board.

Caldisk 143M Disk Drive

Required drive configuration:

1.	DS A	Closed (DS = Dip Switch)
2.	DS B	Open
3.	DS C	Open
4.	DS D	Open
5.	JPR1	Open
6.	JPR2	Open
7.	JPR3	Open
8.	JPR4	Jumper
9.	JPR5	Open
10.	JPR6	Open
11.	JPR7	Jumper
12.	JPR8-11	Select appropiate drive address
13.	JPR12	Jumper
14.	JPR13	Open
15.	JPR14	Open
16.	JPR15	Open
17.	JPR16	Open

Install the termination network only in the last drive in the daisy chain.

Each drive requires 24 volts at 1.5A and 5 volts at 1.0A.

Required Pre-write Compensation: none.

Innotronics 410/420 Disk Drive

Required drive configuration*:

1.	EH	Trace intact
2.	AH	Open
3.	TH	Trace intact
4.	NT	Open
5.	TE	Trace intact
6.	NT	Open
7.	LM	Trace intact
8.	TM	Open
9.	WP	Trace intact (2 places)
10.	NP	Open (2 places)
11.	T4	Jumper
12.	T3, T5, T7	Jumpered on last drive in system
13.	S0-3	Select appropriate drive address

To use the Model 420 (Hard Sector) Disk Drive as a Soft Sector Disk Drive, the following link positions must be set:

1.	IB	Jumper
2.	HS	Open
3.	RD	Jumper
4.	SD	Open (2 places)
5.	VV	Jumper to accept -5 volt supply, otherwise open to accept -7 to -12 volts.

Install the termination network only in the last drive in the daisy chain.

Each drive requires +5 volts at 0.8A, -5 volts at .08A, and +24 volts. The current rating for the 24 volts supply depends on whether the drives will seek individually or simultaneously. If CP/M or a similar DOS is used, the total current is 1.4A. This is because Innotronics applies power to the stepper motor only when the drive is seeking. If software is used that can simultaneously seek on all drives in the system, each individual drive will require 1.4A.

Required Pre-write Compensation: 125 ns.

*The model 410 disk drives are shipped fully compatible with the FDC-I.

PerSci 277 Disk Drive System

The PerSci 277 disk drive system is a system composed of two 8-inch drive units requiring a 50 pin connector to the host system used. Pin assignments to functional signals of the 50 pin connector and the changes required to interface to FDC-I are shown below.

FDC-I	Signal Pin No.	Ground Pin No.	Signal
2	2	1	Unassigned
4	4	3	Drive select 2 right
6	6	5	Ready 1
Cut	8	7	Index 1
Cut	10	9	Seek complete
12	12	11	Restore
Cut	14	13	Remote eject 0
18	16	15	Direct headload
Cut	18	17	Drive select 2 left
20	20	19	Index 0
22	22	21	Ready 0
50	24	23	Spindle motor enable
26	26	25	Drive select 1 left
28	28	27	Drive select 1 right
Cut	30	29	Write protect 1
Cut	32	31	Remote eject 1
34	34	33	Direction select
36	36	35	Step
38	38	37	Write data
40	40	39	Write gate
42	42	41	Track 0
44	44	43	Write protect 0
46	46	45	Read data
48	48	47	Separated data
Cut	50	49	Separated clock

Required drive configuration:

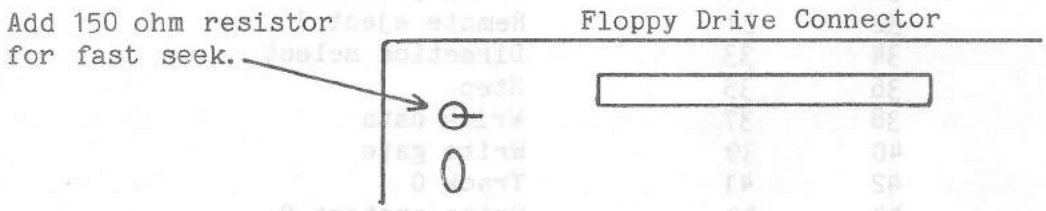
1. Address DIP, U-11: connect pin 4 to 11, and pin 2 to 13.
2. A-B — Raw read data
3. D-BL — Select gate enabled
4. E,F,G — Open
5. J-Z — Enable L=0, and R=1
6. K-L — Wire-OR the Write Protect signals
7. M,N,P — Open
8. R,S,T — Open
9. U-V — Wire-OR the Index signals
10. W-X — Enable Index 0
11. AB-AC — Enable Index 0
12. AD-AE — Enable Index 1
13. AH-AJ — Enable Index 1
14. AM-AL — For spindle motor control
    or
    AM-AN — Spindle motor runs continuously

15. AP-AR — Wire-OR the Ready signals
16. AS-AT — Remote eject, connects L and R together
17. AU,AV,AW — Open
18. BA-BB — Enable Index 1
19. BD-BE — Seek complete enable
20. BF,BH,BJ — Open
21. BK-BM — Enable Index 0

Each drive requires 24 volts at 1.3A, 5 volts at 2.2A, -5 volts at 0.2A and for the spindle power, 7 - 10 volts 2.0A.

Because the PerSci has two physical drives connected to one head positioner, the FDC-I monitor software must be made to seek and recalibrate only drive 0. Otherwise the uPD-765 will seek on both drives 0 and 1, and position past the correct track. Also, if the motor control option is used, the monitor must have the motor control option enabled. See Appendix E for more details.

If fast seek is required (seek rate less than the standard 10 ms step), the seek complete line must be connected to PIO B on one of its spare lines, and a 150 ohm 1/4 watt resistor connected to +5 volts. Contact the factory for further information relating to software requirements.



Required Pre-write Compensation: 250 ns.

Qume DT-8 Disk Drive

Required drive configuration:

1.	A	Jumper
2.	B	Open
3.	X	Jumper
4.	Z	Jumper
5.	HL	Open
6.	R	Jumper
7.	I	Jumper
8.	RI	Trace intact
9.	RR	Trace intact
10.	C	Jumper
11.	D	Open
12.	DC	Open
13.	2S	Open
14.	DS	Open
15.	Y	Open
16.	DL	Open
17.	WP	Trace intact
18.	NP	Open
19.	S2	Trace intact
20.	S1, S3	Open
21.	DS1-4	Select appropriate drive address
22.	B1, 2, 3, 4	Open

Install 2 resistor terminator modules into the last drive in the daisy chain.

Each drive requires 24 volts at 0.9A and 5 volts at 1.1A.

Required Pre-write Compensation: none.

Shugart 800/851 Disk Drive

Required drive configuration:

1.	X	Jumper
2.	DC	Open
3.	D	Open
4.	C	Jumper
5.	I	Trace intact
6.	R	Trace intact
7.	S	Trace intact
8.	DS1-4	Select appropriate drive address
9.	T1, 3, 4, 5, 6	Jumper on last drive in system
10.	T2	Jumper
11.	HL	Open
12.	DS	Open
13.	RI	Trace intact
14.	RR	Trace intact
15.	Y	Open
16.	Z	Jumper
17.	800	Jumper
18.	801	Open
19.	A	Jumper
20.	B	Open

Each drive requires 24 volts at 1.7A, +5 volts at 1.0A, and -5 volts at 0.07A. Note: Many power supplies for floppy drives do not have the required current capability for 2 or more Shugart drives.

Required Pre-write compensation: 250 ns.

Siemens FDD 100-8D Disk Drive

Required drive configuration:

1.	RAD SEL 0-3	Select appropriate drive address
2.	RAD STEP	Jumper pads labelled "2"
3.	"36"	Jumper
4.	A	Open
5.	"34"	Jumper
6.	B	Open
7.	RR	Jumper
8.	"22"	Jumper
9.	RI	Jumper
10.	C	Open
11.	"20"	Jumper
12.	"24"	Jumper
13.	L	Jumper
14.	J	Open
15.	K	Open
16.	"18"	Jumper
17.	M	Open
18.	SS	Jumper
19.	HS	Open
20.	S	Jumper
21.	U	Jumper
22.	R	Open
23.	H	Open (for Activity Indicator)
24.	"16"	Open
25.	E	Jumper
26.	V	Open
27.	"12"	Jumper
28.	G	Open (cut trace)
29.	H	Open (for Phase Option)
30.	F	Jumper

Install the terminator resistor pack in the last drive of the daisy chain.

Each drive requires 24 volts at 1.6A, and +5 volts at 1.0A.

The Siemens drives need to be modified if more than 1 drive will be in the system. On the drive p.c. board, locate IC 4D, a 7402. Cut trace coming from pin 4. Next, locate IC 4E (normally there should not be an IC in this location). Install a jumper wire from pin 12 of IC 4E to the "RI" pads on the p.c. board. This change accommodates the NEC controller.

Required Pre-write Compensation: 250 ns.

Remex 2000/4000 Disk Drive

Required drive configuration:

1.	2S	Jumper
2.	DC	Open
3.	C	Jumper
4.	D	Open
5.	DS1-4	Select appropriate drive address
6.	1B, 2B, 3B, 4B	Open
7.	S1, 2, 3	S2
8.	TS-FS	Don't care
9.	4000/4001	4000
10.	DL	Jumper
11.	S	Jumper
12.	R	Jumper
13.	I	Jumper
14.	X	Jumper
15.	B	Open
16.	A	Jumper
17.	HL	Open
18.	Z	Jumper
19.	DS	Open
20.	Y	Open
21.	RI	Traces intact
22.	RR	Traces intact

The last drive in the daisy chain must have the resistor termination pack installed in location 7A.

Each drive requires 24 volts at 0.6A, +5 volts at 1.0A, and -5 volts at 0.05A.

Required Pre-write Compensation: 250 ns.

MFE 500/700 Disk Drive

Required drive configuration:

1.	SE1,SE2	Open
2.	SE3	Open
3.	L-1	Jumper
4.	L-2	Open
5.	L-3	Open
6.	DL-0	Don't care
7.	DS-1 thru DS-4	Select appropiate drive address
8.	HL3, HL5	Open
9.	HL1, HL2, HL4	Jumper
10.	RR	Jumper
11.	RIS	Jumper
12.	J-4	Jumper
13.	J-6	Jumper
14.	J-7	Jumper
15.	DL0, DL1	Trace intact
16.	DL2, DL3	Open
17.	PRU	Trace intact
18.	PRL	Open
19.	J-5	Jumper
20.	LC2, PS6	Jumper
21.	PS2, LC6	Open
22.	SS1, SS2	Jumper
23.	SS3, SS4	Open
24.	WP1	Jumper
25.	WP2	Open

Only the last drive in the daisy chain should have the termination circuit (Z-15) installed.

Each drive requires 24 volts at 1.4A, +5 volts at 1.2A, and -5 volts at 0.025A.

Required Pre-write Compensation: 250 ns.

FDC-I/II Additional Drives

Control Data 9406-2/3 Disk Drive

Required configuration:

1.	RR	Jumper
2.	RI	Jumper
3.	R	Jumper
4.	2S	Jumper
5.	HS	Open
6.	SS	Jumper
7.	DC	Open
8.	WP	Jumper
9.	NP	Open
10.	D	Open
11.	DD	Jumper
12.	DL	Jumper
13.	A	Jumper
14.	B	Open
15.	X	Jumper
16.	C	Jumper
17.	Z	Jumper
18.	Y	Open
19.	S1	Open
20.	S2	Jumper
21.	S3	Open
22.	E	Open
23.	DR	Jumper
24.	TS	Jumper
25.	FS	Open
26.	NS	Jumper
27.	OS	Open
28.	HO	Jumper
29.	IU	Open
30.	I	Jumper
31.	S	Jumper
32.	Switch 1	Select appropriate drive address

Install the termination network RM3 in the last drive only.

Each drive requires +24 volts at 1.4A and +5 volts at 0.8A.

Required Pre-write Compensation: 250 ns.

NEC FD1160 Disk Drive

Required configuration:

1.	P51, DLD	Jumper (DLS Open)
2.	P52, HLS	Jumper (HLD Open)
3.	P53, N	Jumper (RDR Open)
4.	P54, P55	Select appropriate drive address
5.	P56, E	Jumper (IFU Open)
6.	P57, C	Jumper (PWD Open)
7.	P58, PRI	Jumper (PRS Open)
8.	P59, N	Jumper (DLH Open)
9.	P60, FPL	Jumper (J7 Open)

Install the termination networks RN-1 and RN-2 in the last drive only.

Each drive requires +24 volts at 0.9A, +5 volts at 1.5A and -5 volts at 0.07A.

Required Pre-write Compensation: none.

Qume DataTrak 5 Disk Drive

Required configuration:

1.	DS0-3	Select appropriate drive address
2.	HS	Jumper
3.	MX	Open
4.	HM	Open
5.	P-M	Jumper
6.	P-S	Open
7.	A	Open
8.	B1	Open
9.	B3	Open
10.	HL	Open

Install the termination network U2B in the last drive only.

Each drive requires +12 volts at . A and +5 volts at . A.

Required Pre-write Compensation: none.

FDC-I/II Additional Drives

Pertec FD250 Disk Drive

Required configuration:

1.	Switch 1	Select appropriate drive address
2.	DP	Open
3.	DH	Open
4.	IS	Jumper
5.	DC	Open
6.	HL	Jumper
7.	DL	Open
8.	IB	Open
9.	HB	Jumper

Install the termination network U2 in the last drive only.

Each drive requires +12 volts at 1.6A and +5 volts at 0.8A.

Required Pre-write Compensation: none.

Note: Since the DataTrak 5 does provide a READY signal the following change is neccessary on the p.c. adaptor board:

1. Cut the trace between pads 5 and 6.
2. Add a jumper between pads 4 and 5.

Pertec FD650/651 Disk Drive

Required configuration:

1.	J101, 1-16	Open
2.	J101, 2-15	Open
3.	J101, 3-14	Open
4.	J101, 4-13	Open
5.	J101, 5-12	Jumper
6.	J101, 6-11	Open
7.	J101, 7-10	Open
8.	J101, 8-9	Jumper
9.	S1	Open
10.	4B, 3B, 2B, 1B	Open
11.	2S	Jumper
12.	IWBSY	Open
13.	IHCS	Open
14.	ID	Open
15.	RI	Jumper
16.	RR	Jumper
17.	SA	Jumper
18.	SS	Open
19.	SH	Open
20.	X	Open
21.	DDS	Open
22.	DD	Jumper
23.	S3	Open
24.	Switch 1	Select appropriate drive address
25.	650	Open
26.	651	Jumper
27.	DL	Open
28.	DSSEP	Jumper
29.	SSD	Open
30.	DSDO	Jumper
31.	HCS	Open
32.	T43	Jumper
33.	WP	Jumper
34.	NP	Open
35.	Ø	Open
36.	WPTD	Open
37.	Z	Jumper
38.	PNL	Open
39.	BDL	Open
40.	S	Jumper
41.	R	Jumper
42.	I	Jumper
43.	D	Open
44.	S2	Jumper
45.	DC	Open

Install the termination networks U1 and U2 in the last drive only. Each drive requires +24V at . A, +5 volts at . A and -5 volts at . A.

Required Pre-write Compensation: none.

FDC-I/II Additional Drives

Shugart 850/851 Disk Drive

Required configuration:

1.	X	Shunt intact
2.	DC	Open
3.	D	Open
4.	C	Jumper
5.	I	Shunt intact
6.	R	Shunt intact
7.	S	Shunt intact
8.	DS1-4	Select appropriate drive address
9.	HL	Shunt open
10.	DS	Open
11.	RI	Trace intact
12.	RR	Trace intact
13.	Y	Open
14.	Z	Shunt intact
15.	850	Jumper
16.	851	Open
17.	A	Shunt intact
18.	B	Shunt open
19.	1B, 2B, 3B, 4B	Open
20.	2S	Jumper
21.	WP	Trace intact
22.	NP	Open
23.	S1	Open
24.	S2	Jumper
25.	S3	Open
26.	DL	Jumper
27.	M	Jumper
28.	TS	Open
29.	FS	Jumper
30.	IW	Jumper
31.	RS	Jumper
32.	RM	Open
33.	HLL	Open
34.	IT	Jumper
35.	HI	Open
36.	F	Open
37.	AF	Jumper
38.	NF	Open

Install the termination network IC2F in the last drive only.

Each drive requires +24 volts at 1.0A and +5 volts at 1.1A.

Required Pre-write Compensation: 250 ns.

FDC-I/II Additional Drives

Tandon TM100 Disk Drive

Required configuration:

1.	MX	Open
2.	HS	Jumper
3.	HM	Open
3.	NDS0-NDS3	Select appropriate drive address

Install the termination network 2F in the last drive only.

Each drive requires +12 volts at 0.9A and +5 volts at 0.6A.

Required Pre-write Compensation: none.

FDC-I User Manual, Appendix D

Contents

Memory Board Set-up

MSC DM-6400

Required configuration:

Header #1, the 18-pin DIP

2 - 1	Jumper
4 - 5	Jumper
6 - 5	Jumper
9 - 11	Jumper
13 - 12	Jumper
15 - 14	Jumper
17 - 18	Jumper

Header #2, the 16-pip DIP

1 - 2	Jumper
3 - 11	Jumper
6 - 7	Jumper
9 - 8	Jumper

MSC DMB-6400

Required configuration:

Header #1

4 - 9	Jumper
1 - 16	Jumper
11 - 13	Jumper
8 - 10	Jumper
7 - 13	Jumper

Area "B", jumper the Pull Up to Phantom

Area "C", cut the trace from "5" to "6" near the edge connector on the BACK of the p.c. board. Add a jumper from "4" to "5". This accomodates the floppy wait cycles.

S-1	all down, bank select on reset.
S-2	address= 64K
S-3	both "on", phantom pin, 01 or 02.
S-4	phantom all
S-5	bank select port address.

Cromemco 16KZ

The 16KZ memory needs modification to enable refresh during the extended I/O operations that occur in FDC-I. These modifications are not permanent in nature and the board can easily be changed back to its original configuration.

An unused portion of U-31, a 74LS08, is used to provide the extra gating needed for this modification. The first step in the modification is to pull U-14 out of its socket and bend pin 2 out so that when the IC is reinserted, pin 2 will not enter the socket. Attach a small wire (30 gauge wire-wrap is fine) to pin 2 of U-14 and route it through any convenient hole in the p.c. board. Solder the other end of this wire to pin 11 of U-31.

Connect pin 12 of U-31 to pin 6 of U-63.

Connect pin 13 of U-31 to pin 2 of U-63.

With this change, the 16KZ will initiate refresh operations during the extended input/output operations of FDC-I, thus preserving memory.

Chrislin CI-S100

Required configuration:

Next to U55	1 - 2	Open
	3 - 4	Jumper
Above U63	1 - 2	Jumper
	2 - 3	Open
Above U38	1 - 2	Open
	2 - 3	Jumper

For disabled bank select option - all 64K enabled on reset:

Next to U54	1 - 2	Open
	2 - 3	Jumper
Next to U58	1 - 2	Jumper
	3 - 4	Jumper
	1 - 3	Open
	4 - 5	Open

Central Data 64K RAM

Required Configuration:

Header IC-15		Header IC-16	
1 - 16	Open	1 - 16	Not used
2 - 15	Jumper	2 - 15	Jumper
3 - 14	Open	3 - 14	Open
4 - 13	Open	4 - 13	Jumper
5 - 12	Open	5 - 12	Open
6 - 11	Open	6 - 11	Jumper
7 - 10	Open	7 - 10	Open
8 - 9	Jumper	8 - 9	Not used

In order to provide asynchronous refresh during a system wait cycle the Central Data board must be modified in the following manner: The trace coming from pin 72 (PRDY) on the S-100 bus must be cut. In its place an inverted PWAIT (pin 27 on the S-100 bus) signal must be provided to pin 13 of IC-2. There are several available inverters on the Central Data board (IC-7,17) or if desired a chip may be added at IC-5.

FDC-I User Manual, Appendix E

Contents

Introduction

Teletek's FDC-I board comes with a 2K monitor. The monitor has several commands that allow the user to access memory, assign input and output devices, transfer execution, program 2716 EPROMs (requires an external power supply), get floppy drive status and read or write on floppy disks. The monitor also provides all of the routines for the I/O facilities of the FDC-I board.

Upon initialization, the monitor waits for a carriage return (CR) from either the parallel (PIO A) or serial (SIO A) port. If it gets a CR from the parallel port, it assigns input from the parallel port and output to a user output routine. If the CR is from the serial port, I/O will be assigned to the serial port. When I/O has been assigned, a Form Feed and "FDC-I r.r Monitor (v)" will be printed on the output device (where r.r is the release number and v is the version of the monitor).

On the serial port (SIO A) the monitor will automatically determine the baud rate of the device, but the device must be at one of the following baud rates: 9600, 4800, 2400, 1200, 600, 300, 150, or 110. The user should hit the CR key until "FDC-I r.r Monitor (v)" appears. The maximum number of carriage returns is 6 for a 110 baud rate device. Note: this occurs only after a reset and only if I/O has not yet been assigned.

The baud rate on the second serial port (SIO B) is initialized at 300 baud but the user may change this with the "SS" command or through the "STSSP" routine.

Monitor Commands

When the monitor is ready to accept a command it prints a ">_" as a prompter on the output device. The user can now enter a command. All commands are of the following format: "CD_H1_H2_H3<CR>", where CD is a one or two letter command chosen from below, _ is a space, H1, H2, and H3 are one to four digit hex numbers and <CR> is a carriage return. Not all the commands require the three hex numbers, some require one, two or even none. Only the last four digits are used and preceding zeros may be omitted. The DELete or the Back Space (CNTL/H) keys may be used to correct errors. The ESCape key will cancel the command line. The monitor accepts both upper and lower case letters.

The ESCape key has a special function. In the commands marked with <ESC>, the ESCape key will terminate the command prematurely. Hitting the ESCape key will cause "Escape" and the prompter to be printed. This action also takes place when the user is entering a command and he hits ESCape.

If the monitor encounters an error it will print one of the following error messages:

C Err! Command error - the monitor did not recognize the command.
H Err! Hex error - the user entered an invalid hex digit.
V Err! Verify error - the monitor found a verification error.

Errors for the floppy read/write commands are handled as follows: After completion of the command, the memory pointer is printed along with ST0, ST1, ST2, C, H, R and N, the status registers of the NEC uPD765. If bits 7 and 6 of ST0 are zero then no errors occurred. If bits 7 and 6 are not zero then an error occurred. The user should refer to the NEC uPD765 manual to interpret the error code in ST0, ST1 and ST2. C, H, R and N are updated as explained in the uPD765 manual. These commands are marked <RW>.

These are the commands:

BT<CR> — BooT: boot up the disk operating system. Reads in the first sector on track 0 on drive 0 at location 0000 and transfers to 0000 if no errors occurred. It tries until no errors occur. <ESC>

DA_H1<CR> — Dump ASCII: dumps memory in ASCII. First H1 is printed followed by the contents of the next eight bytes (including H1) in hex, then the ASCII character of each of the eight bytes. If the character is non-printing a period (.) is printed instead. If the user hits the space bar, the process is repeated with the next eight bytes. Any other key will cause "Complete" to be printed and the command to terminate.

DS_H1<CR> — Drive Status: prints the status (ST3) of floppy disk drive H1. See the uPD765 manual for the explanation of ST3.

E_H1<CR> Enter memory: enters data into memory starting at H1. First the memory address and its contents are displayed. If new data is desired, enter the two hex numbers. After the new data is entered into memory it is read back and displayed, followed by the next memory address and its contents. Hitting the space bar will bring up the next memory location without changing the contents while hitting the Back Space key will bring up the previous memory location. The above process will be repeated until the RETURN key is hit; "Complete" will then be printed and the command terminated.

F_H1_H2_H3<CR> Fill memory: fills the memory block from H1 to H2 with H3, prints "Complete" and terminates.

G_H1<CR> Go: transfers control to H1 by executing a CALL instruction. If the routine ends with a RET instruction and the stack pointer points where it did upon entry to the routine, a return to the monitor will be made.

M_H1_H2_H3<CR> Move memory: moves the memory block from H1 through H2 to H3 and then verifies the move. Prints "Complete" when done. <ESC>

IO_H1<CR> Assign I/O devices: stores H1 in the IOBYTE in the System RAM. The IOBYTE is defined as follows:

Bit	Device	Function
7	Reserved	Input
6	SIO B	"
5	SIO A	"
4	PIO A	"
3	User Routine	Output
2	SIO B	"
1	SIO A	"
0	PIO A	"

Setting the bit enables the device function. More than one bit may be set for either the input or output function; i.e., if bits 2 and 1 are set, then output will appear on the devices connected to SIO B and SIO A.

P_H1_H2_H3<CR> Program: programs the memory block from H1 to H2 in the 2716 EPROM located at H3. Before programming the user must apply 25.5 volts to the pins on the right hand side of the FDC-I board. After "Complete" is printed the 25.5 volts should be removed. Any bytes that did not program correctly will be listed. <ESC>

RC_H1<CR> ReCalibrate: recalibrates drive H1. Prints ST0 if any errors occurred.

RD_H1_H2_H3<CR> Read Double density: reads MFM data from head and drive H1 (bit 2 is the head and bits 1 and 0 are the drive), track H2 and sector H3. The user is asked "Addr & no. of sctr's". Enter "H4_H5<CR>" where H4 is the address where the data will go and H5 is the number of sectors to read. After execution, the next memory location is printed along with ST0, ST1, ST2, C, H, R and N. <RW>

RS_H1_H2_H3<CR> Read Single density: same as RD except it reads FM data. <RW>

SD_H1<CR> Set Drive size: sets the floppy disk drive size for subsequent operations. If H1 is non-zero then 8 inch drives are set; if H1 is zero then 5.25 inch drives are set. Note: In order for this command to work correctly, bit 0 of PIO B must be jumpered to the MM pad as explained in the "Mini/Maxi Floppy Selection" section of this manual.

SS_H1_H2<CR> Set Serial speed: sets the baud rate of SIO H1 (H1 must be "A" or "B") according to H2, chosen as follows:

H2	Speed
0	9600
1	4800
2	2400
3	1200
4	600
5	300
6	150
7	110

V_H1_H2_H3<CR> Verify: verifies the memory block from H1 to H2 against that at H3. It will print any verification errors along with the addresses. It prints only 13 verification errors and then waits for any key to continue. Prints "Complete" when the verification process is over. <ESC>

WD_H1_H2_H3<CR> Write Double density: writes MFM data on head and drive H1, track H2 and starting at sector H3. It asks the user "Addr". Enter "H4_H5<CR>" where H4 and H5 specify the memory block to be written on the disk. After execution, the next memory location is printed along with ST0, ST1, ST2, C, H, R and N. <RW>

WS_H1_H2_H3<CR> Write Single density: same as WD except it writes FM data. <RW>

Interfacing with the Monitor

At the beginning of the monitor are several jump vectors which ease interfacing other programs with the FDC-I. These jump vectors provide the necessary routines to use the FDC-I I/O facilities. The user's program just calls the appropriate jump vector. Data is handled in the accumulator (register A) in all the routines except where noted (none of the routines use the Z-80 alternate registers). In the following discussion, "Z" refers to the zero flag and "BP" refers to the Base Page address of the monitor (BP varies with the different monitor versions).

These are the jump vectors:

BP00 INIT — This routine initializes all the hardware on the FDC-I board and the System RAM, assigns the input and output devices, prints "FDC-I r.r Monitor (v)" and enters into the command input mode. It does not return to the calling program.

BP03 WARM — This routine is the warm entry point into the monitor. It prints "FDC-I r.r Monitor (v)" and enters into the command input mode. It does not return to the calling program.

BP06 INPM — This routine returns with data from the assigned input device with bit 7 reset.

BP09 INPUT — This routine returns with data from the assigned input device.

BP0C SBIN — This routine returns with data from the second serial port (SIO B).

BP0F OUTCH — This routine outputs data to the assigned output device(s).

BP12 SBOUT — This routine outputs data to the second serial port (SIO B).

BP15 POUT — This routine outputs data to the parallel port (PIO A). Before calling this routine the parallel port must be configured as an output port by calling the routine ASPA with A=0, otherwise this routine will not return. Also, this routine assumes that a "Centronics" type device is connected to the port.

BP18 GSTAT — This routine gets the status of the assigned input device. It returns with Z set and A=0 if no data is available, otherwise Z is reset and A=0FFH.

BP1B STATIN — This routine is the same as GSTAT except that it returns with the data (bit 7 reset) if it is available.

BP1E STATB This routine is the same as GSTAT except that it gets the status of the second serial port (SIO B).

BP21 STMODE This routine sets the mode for subsequent floppy disk operations. Register A should have the mode as follows:

Bit 7 set for multi-track
" 6 " " MFM (double density)
" 5 " " skip deleted data address mark
Bits 4-0 don't care.

Refer to the uPD765 manual for more details. This routine uses all the registers.

BP24 STHDR This routine sets the head and drive for subsequent disk operations. Register A should have the head and drive as follows:

Bits 7-3 don't care
Bit 2 with the head (0 for side 0, 1 for second side)
Bits 1,0 with the drive (0-3).

BP27 STSECT This routine sets the sector for subsequent disk operations.

BP2A RECAL This routine recalibrates (moves head to track 0) the drive set by STHDR. If no errors occurred Z is set and A=0, otherwise Z is reset and A is non-zero. It sets the track to zero.

BP2D SEEK This routine sets and seeks the track contained in A, on the drive set by STHDR. If no errors occurred Z is set and A=0, otherwise Z is reset and A is non-zero.

BP30 READ This routine reads from the floppy disk the number of sectors in register E into memory specified by the register pair HL. The mode, drive, head, track and starting sector must already have been set. If no errors occurred Z is set and A=0, otherwise Z is reset and A is non-zero. This routine uses all the registers.

BP33 WRITE This routine writes on the floppy disk the number of sectors in register E from memory specified by the register pair HL. The mode, drive, head, track and starting sector must already have been set. If no errors occurred Z is set and A=0, otherwise Z is reset and A is non-zero. This routine uses all the registers.

BP36 RTST0 This routine returns the address of ST0 in the register pair HL. ST0 and the next 6 bytes in the System RAM (ST1, ST2, RC, RH, RR and RN) contain the error code after a disk operation. See the uPD765 manual to interpret the error.

BP39 RTSEC This routine returns the address of SEC in the register pair HL. The real-time clock is kept at SEC and the next five bytes (MIN, HOUR, YEAR, DAY and MONTH) . SEC is updated every second, MIN is updated every minute, HOUR is updated every hour and the date is updated every 24 hours. HL-1 points to USTM which is a user time-out flag which is decremented every second if it is non-zero. All the values are kept in binary.

BP3C ASIO This routine assigns the I/O devices. It stores A in the IOBYTE. The IOBYTE is defined as follows:

Bit	Device	Function
7	Reserved	Input
6	SIO B	"
5	SIO A	"
4	PIO A	"
3	User Routine	Output
2	SIO B	"
1	SIO A	"
0	PIO A	"

Setting the bit enables the device function. More than one bit may be set for either the input or output function; i.e., if bits 2 and 1 are set, then output will appear on the devices connected to SIO B and SIO A.

BP3F ASPA This routine assigns the function of the parallel port (PIO A). If A is not zero the port is configured as an input port. If A=0 the port is configured as an output port. The parallel port must be configured as an output port before calling the routine POUT. When the port is configured as an output port the assumption is made that the device connected to the parallel port will be a "Centronics" type device. This routine uses all the registers.

BP42 SRTDF This routine stores A in the Time Display Flag (TDF) in the Sytem RAM. If A is non-zero then a user time routine (USTIME) is called every second by the real-time clock interrupt routine, otherwise if A is zero USTIME is not called.

BP45 STSSP This routine sets the speed on the serial ports. Register C should be 0A hex for SIO A or 0B hex for SIO B. Register pair DE should be chosen as follows for the desired baud rate:

DE	Baud Rate
0000	9600
0001	4800
0002	2400
0003	1200
0004	600
0005	300
0006	150
0007	110

This routine uses all the registers.

BP48 USR This is an optional user routine. It is not implemented but is left for the user. USR is provided so that a user may have a fixed jump vector to any routine he may desire to implement.

BP4B READID This routine will read the first ID it can on the floppy drive set by STHDR and in the mode set by STMODE. If no errors occurred Z is set and A=0, otherwise Z is reset and A is non-zero. See the uPD765 manual for more details on the READ ID command.

BP4E DRVST This routine will return the status (ST3) of the drive set by STHDR. See the uPD765 manual for more details on the SENSE DRIVE STATUS command.

BP51 RTDRDY This routine will return the address of DRDY0 in the register pair HL. Four drive ready flags (DRDY0, DRDY1, DRDY2 and DRDY3) exist in the System RAM with the ready status of each of the four drives. If DRDYx is non-zero then drive x is ready; if DRDYx is zero then the drive is not ready.

BP54 SPEC This routine will specify the Step Rate Time (SRT), Head Unload Time (HUT) and the Head Load Time (HLT). Call SPEC with register D= SRT/HUT and register E= HLT(bits 7-1). See the uPD765 manual for more details on the SPECIFY command. Note: the monitor always sets the uPD765 in the DMA mode.

BP57 STSIZE This routine will set the size of the floppy disk drive for subsequent read/write operations. If register A is non-zero then 8 inch drives are set; if A is zero then 5.25 inch drives are set. This routine also uses the HL register pair. Note: In order for this command to work correctly, bit 0 of PIO B must be jumpered to the MM pad as explained in the "Mini/Maxi Floppy Selection" section of this manual.

Interfacing with the Floppy Disk Routines

The FDC-I monitor provides all the neccessary routines to access a floppy disk. These routines permit the user to read or write on floppy disks in an IBM compatible format, either in single or double density. It is recommended that the user become familiar with the NEC uPD765 manual and how the chip works.

The BT command of the monitor boots up a disk operating system such as CP/M. This command first sets up the single density mode, recalibrates drive 0, reads in the first sector on track 0 to memory location 0, and then transfers to location 0 if no errors occurred. If an error does occur, it will retry until successful.

The routine STMODE sets the mode for read/write operations. It should be called before changing modes or errors will occurr; i.e., when changing from single to double density or vice versa. If the BT command is used, the mode is set for single density after the initial boot operation.

The routine STHDR sets the head and drive for disk operations. It should be used when changing heads and/or drives.

The routine STSECT sets the starting sector for read/write operations. It should be called when changing sectors.

The routine RECAL will recalibrate the drive set by STHDR. It also sets the track register to zero.

The routine SEEK will seek the track contained in register A on the drive set by STHDR. It also sets the track register for subsequent read/write operations.

The routine READ will read from the floppy disk in the mode set by STMODE, starting at the sector set by STSECT, on the track set by SEEK and on the head and drive set by STHDR. It will read the number of sectors specified in E into the memory location specified by HL. READ uses all the registers.

The routine WRITE will write to the floppy disk in the mode set by STMODE, starting at the sector set by STSECT, on the track set by SEEK and on the head and drive set by STHDR. It will write the number of sectors specified in E from the memory location specified by HL. WRITE uses all the registers.

The routine READID will execute the READ ID command as explained in the uPD765 manual. It tries to read the first ID on the drive set by STHDR and in the mode set by STMODE. This routine can be used to determine the density of a diskette. See the uPD765 manual for more details on the READ ID command.

The routine DRVST will return the status (ST3) of the drive set by STHDR. The status will tell if the drive is ready, write protected or double-sided. See the uPD765 manual for more details on the SENSE DRIVE STATUS command.

The routine RTDRDY will return the address of DRDY0 in the register pair HL. DRDY0 and the next three locations DRDY1, DRDY2 and DRDY3 in the System RAM contain the ready status of each of the four drives. If DRDYx is non-zero then drive x is ready otherwise if DRDYx is zero the drive is not ready. This routine can be used to check the ready status of a drive before any floppy operation is performed.

The following two routines are provided for flexibility. The routine SPEC will specify the step rate, head unload time and the head load time. Enter the routine with D= SRT/HUT and E= HLT. See the uPD765 manual for more details on the SPECIFY command. Use this routine to change the drive timing constants. Note: the monitor will always set the uPD765 in the DMA mode regardless of what bit 0 of register E is.

The routine STSIZE will set the size for read/write operations. Enter with A non-zero for 8 inch drives or with A equal zero for 5.25 drives. Use this routine to switch between mini and maxi drives. Note: This routine assumes that bit 0 of PIO B is jumpered to the MM pad on the back of FDC-I.

Errors for the floppy disk routines are handled as follows: If the disk operation was successful then the zero flag is set and A=0. If an error occurred then the zero flag is reset and A is non-zero. The error code is stored at ST0 and the next 6 bytes (ST1, ST2 RC, RH, RR and RN) in the System RAM. The routine RTST0 will return the address of ST0 in HL. Refer to the uPD765 manual to interpret the error code.

User Floppy Disk Routines

The user is free to write his own floppy disk routines to implement floppy disk operations that are not in the monitor; i.e., a SCAN EQUAL routine. If the user does write any routines he should follow the general outline of the routines that are in the monitor (see RECAL, SEEK, READ, WRITE or READID). A routine like FDDRW will be needed to send the different command bytes to the NEC uPD765 chip. The actual disk routine must be callable and end with an "endless" loop instead of the normal return instruction. This requirement is mandatory because of the way that floppy disk interrupts are handled in the monitor. Even though the routine ends in an "endless" loop, it will return to the calling program. Further, if no errors occurred, the zero flag will be set and register A will be zero; if an error did occur, then the zero flag is reset and register A will be non-zero. Again, the user should refer to the routines that are in the monitor and follow their general outlines.

Real-Time Clock

The FDC-I board has an on-board real-time 24 hour clock and date calender. The time and date are kept at SEC and the next 5 bytes (MIN, HOUR, YEAR, DAY and MONTH) in the System RAM. The CPU is interrupted every second and the time updated. The date is also updated if necessary but leap years and turns of centuries are not accounted for. All the values are kept in binary.

The user may access the clock in the System RAM through the routine RTSEC. This routine returns the address of SEC in the HL register pair.

A user time-out flag is also provided. Whenever USTM in the System RAM is non-zero, it is decremented every second until it is zero. USTM may be accessed by calling RTSEC and decrementing register pair HL once.

Upon reset the time is not initialized. It is up to the user's software to initialize the time and date. Until the time and date are initialized they should be considered invalid.

Sample BASIC Program

The following program written in Microsoft BASIC will initialize the date and time:

```
100  REM  The FDC-I standard monitor is assumed.
110  DEFINT I, M, N
120  REM
130  REM  Define USR0 at FC00 hex:
140  M = &HFC00
150  DEF USR0 = M
160  REM
170  REM  The following "POKE" sequence sets up the
180  REM  following assembly language subroutine:
190  REM       EX   DE,HL          ;Save integer location
200  REM       CALL 0E039H         ;Get addr of SEC
210  REM       EX   DE,HL
220  REM       LD   (HL),E         ;Put addr of SEC at
230  REM       INC  HL             ;integer location
240  REM       LD   (HL),D
250  REM       RET
260  FOR I = 0 TO 8
270       READ O: POKE M+I,O
280  NEXT
290  REM
300  REM  Put address of SEC in integer M:
310  M = USR0(0)
320  REM
330  REM  Now input and set up the date and time:
340  FOR I = 5 TO 0 STEP -1
350       READ P$: PRINT P$;: INPUT N: POKE M+I,N
360  NEXT
370  SYSTEM:   REM Return to CP/M
380  REM
390  REM  Assembly language subroutine code:
400  DATA &HEB, &HCD, &H39, &HE0, &HEB, &H73, &H23, &H72, &HC9
410  REM
420  REM  Data for getting date and time:
430  DATA "Month", "Day", "Year", "Hour", "Minutes", "Seconds"
440  END
```

Programming 2716 EPROMs

The FDC-I can program 2716 EPROMs with the "P" command. The user must apply 25.5 volts to the two pins on the right side of the FDC-I board before attempting to program. The "+" pin is for the 25.5 volts and the "-" pin is ground. After programming the 25.5 volts should be removed or some erroneous data may be accidentally programmed into one of the on-board EPROMs. The 2716 EPROM programming specifications permit between 1 and 2048 bytes to be programmed at one time.

Parallel Port (PIO A)

The parallel port can be configured as an input or output port. Normally it is configured as an input port but it can be changed to an output port with the ASPA routine. The port should be configured as an output port before assigning output to PIO A or calling the POUT routine, otherwise the monitor will hang up. When the port is configured as an output port the assumption is mad at the device connected to the parallel port will be a "Centronics" type device. Therefore the (Not) Data Strobe from the device should be connected to bit 7 on the port and the (Not) Acknowledge from the device connected to the (Not) Strobe on the port.

Monitor Options

Several options are provided for the user to implement if he so desires. To enable these options the user must program one or two NOPs (00) and maybe an external routine address in the appropriate locations in the monitor EPROM as explained in the following discussion. See the monitor listing for the exact locations.

The first option will set the drive size to 5.25 inch drives. To enable this option, change the "LD A,-1" to a "LD A,0". The location will be marked by the statement "IF MINI". Note: remember, to read mini drives requires a change in the clock speed to the uPD765.

The second option will cause the monitor to do a BT command after initialization without entering into the command input mode. From the monitor listing the user can determine where the two NOP'S should be programmed. The location will be marked by the statement "IF CLDBT". Note: if this option is enable, the user must assign I/O with the ASIO routine in the code that is first executed at 0000.

The next option is for users who have PerSci drives. Since on the PerSci drive one stepper motor controls "two" drives it is necessary to recalibrate and seek on only one drive. To enable this option the user has to program two NOPs followed by a "SUB A" in the RECAL and SEEK routines. The locations will be marked by the statement "IF PERSCI". Note: PerSci users may also want to enable the motor control option.

The next option will permit the monitor to control the DC motor on 5.25 inch or PerSci drives. The motor control line is bit 2 out of the second parallel port (PIO B). Once enabled, the monitor will control the DC motor. Before any read or write operation the motor will be turned on if it is off. After a two second delay (to give time for the motor to get up to speed) the operation will take place. If the motor is already on, the operation will take place without any delay. If the next read or write operation does not take place within the next thirty seconds the motor will be turned off. The timing for the delays is provided by the real-time clock. To enable this option, the user must program a NOP at the beginning of the MOTOR routine. It will be marked by the statement "IF MINI OR PERSCI".

A user output routine is also an option. This routine will be called whenever OUTCH is called and bit 3 of the IOBYTE is set. To enable this option a NOP and the address of the user routine has to be programmed at USOUT. If this option is enabled, the data to output will be passed in register A. No registers must be changed.

The final option is a user time routine. This routine may be used to print the time or any function the user desires. It will be called every time the real-time clock interrupts and the TDF flag in the System RAM is set. To enable this option a NOP and the address of the user routine has to be programmed at USTIME. When this routine is called, register pair HL will point to SEC in the System RAM. The user may freely use registers AF and HL but must save any other registers that he uses. Also, only 16 bytes of stack are available. Remember, the time and date are kept in binary values.

Monitor Versions

The FDC-I monitor comes in several versions depending on the memory map of the FDC-I board. All the monitors are identical except in their base page address and the location of the System RAM. In other words, all the monitors were assembled from the same source code, just the "ORG" statement was changed for the different versions. All the I/O routines are fixed relative to the base page address of the monitor. The following table gives the base page address of the monitor, on-board memory and System RAM location for the different versions:

Version	Base Page	On-board Memory	System RAM
a*	E0	E000-FFFF	FF80-FFFF
b	F0	F000-FFFF (or F000-F7FF)	EF80-EFFF**
c	F8	F800-FFFF	F780-F7FF** ***

*Standard version of the monitor.
**This RAM is not on the FDC-I board and has to be provided externally by the user. The monitor will not come up without this RAM.
***The M6 memory option does not require the external memory.

Monitor System RAM

The FDC-I monitor needs 128 bytes for its System RAM (in addition to the stack which is located before the System RAM). This RAM is used to store flags, pointers and other data for the I/O facilities of the monitor. Jump vectors at the beginning of the monitor are provided to access the different locations in the System RAM. It is highly recommended that the user use these vectors and not address the System RAM directly. The System RAM is likely to change with different releases of the monitor and addressing the System RAM directly would cause errors should the System RAM change. Thus to avoid errors and headaches, use the jump vectors.

Monitor Initialization

The following is a list of the initialization procedure of the monitor upon reset or power-on:

1. The first parallel port (PIO A) is set up as an input port with interrupts enabled.
2. The second parallel port (PIO B) is set up in the control mode with no interrupts. Bits 7 and 3 are inputs while the rest of the bits are outputs.
3. Channel 0 of the CTC is set up to provide 9600 baud to SIO A.
4. Channel 1 of the CTC is set up to provide 300 baud to SIO B.
5. Channel 2 and 3 of the CTC are set up to provide the one second interrupt for the real-time clock.
6. Both serial ports (SIO A & B) are set up to transmit/receive 8 data bits, 2 stops bit and no parity. The handshake lines are set to the appropriate levels and "Auto Enables" is selected. "Auto Enables" means that if the appropriate handshake line is not at the correct level, the serial port will not transmit/receive data; i.e., the port will only transmit if /CTS is asserted. The ports are set to interrupt on receive data but to transmit in a polled mode.
7. The IOBYTE and Time Display Flag are reset.
8. The user time-out flag is set for 30 seconds.
9. The ready flags of all four floppy drives are (re)set.
10. The floppy drive size is set to 8 inch and the following timing constants are set: SRT= 10 msec, HUT= 240 msec and HLT= 36 msec.

The following items are not altered or initialized in any way: the real-time clock values (SEC, MIN, HOUR, YEAR, DAY or MONTH) or the parameters for floppy operations (mode, drive, head, track or sector).

Memory-mapped Video Driver

Enclosed is a source listing for a memory-mapped video driver. This driver is for a 1K video board such as Processor Technology's VDM, addressed at CC00 hex. It has 15 lines of 65 characters and reconizes several control characters. The top line is used to display the real-time clock in the following format "mm/dd/yy hh:mm:ss", where mm is the month, dd is the day, yy is the year, hh is the hour, mm is the minutes, and ss the seconds. Normally it is used with USOUT and USTIME enabled and with a keyboard connected to PIO A.

Sample FORMAT Program

A source listing for a sample formatting program is provided. This program will format diskettes in single or double density and on one or both sides of the diskette. It is set up for 8 inch drives but is easily adaptable for 5.25 inch drives. It runs under CP/M but assumes that the base page address of the monitor is located at 40H.

CP/M 1.4 Software

Two source listings are provided with the CP/M 1.4 Cold Start Loader and BIOS. These two programs permit the CP/M 1.4 operating system to run with the FDC-I board. Teletek also has the Cold Start Loader and BIOS for CP/M 2.2. This CP/M 2.2 BIOS will permit users to intermix single and double density diskettes. For more information on these contact Teletek.

In Case of Trouble

If you have any trouble in bringing up the monitor, reading/writing on floppy disks or any other problem and you call Teletek, please have your system near you. It usually saves time if you can readily try some commands or other action while in contact with Teletek. Also, please remember to read all the manual.

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001
Video Utility Package for FDC-I Monitor

                        0002 ;Revision date: 4-7-80
                        0003 ;
                        0004 ;Written by Aram Perez, 3-13-80
                        0005 ;
                        0006 ;This package contains a video screen driver and a routine to
                        0007 ;print the time.  The location of the video screen is VSL and
                        0008 ;a 1K memory mapped video is expected.  The first seventeen
                        0009 ;locations (VSL - VSL+16) are used for the date and time, i.e.
                        0010 ;03/04/80 01:36:22.  The rest of the top line is not used at
                        0011 ;all.  The video driver uses the rest of the screen providing
                        0012 ;15 lines of 64 characters.  The video driver reconizes the
                        0013 ;following special codes:  1) Back Space (BS, 08) move back 1
                        0014 ;location.  2) Line Feed (LF, 0A) move down 1 line.  3) Home
                        0015 ;(HM, 0B) move cursor to home position (left hand margin, 2
                        0016 ;line from top).  4) Form Feed (FF, 0C) clear the screen.
                        0017 ;5) Carriage Return (CR, 0D) move cursor to left hand margin.
                        0018 ;6) Rub Out (or Delete) (RB, 7F) same as BS.  7) Skip Over
                        0019 ;(SO, 0E) skip over 1 location (non-destructive).  8) Clear
                        0020 ;Top Line (SI, 0F) clear CC00-CC3F.  8) Tab (HT, 09), tab to
                        0021 ;every 8th location.
                        0022 ;
(CC00)                  0023 VSL:    EQU    0CC00H          ;Video screen location
(CC10)                  0024 DDL:    EQU    VSL+16          ;Date/time display location
(CC40)                  0025 TSCR:   EQU    0CC40H          ;Top of screen (home position)
(0040)                  0026 LINE:   EQU    64              ;No. of char's/line
(000F)                  0027 NLN:    EQU    15              ;No. of lines for driver
(03BF)                  0028 SCR:    EQU    NLN*LINE-1      ;Size of screen-1
(CFFF)                  0029 BSCR:   EQU    TSCR+SCR        ;Bottom of screen
(0008)                  0030 BS:     EQU    8               ;Back Space
(0009)                  0031 HT:     EQU    9               ;Horizontal Tab
(000A)                  0032 LF:     EQU    10              ;Line Feed
(000B)                  0033 HM:     EQU    11              ;Home
(000C)                  0034 FF:     EQU    12              ;Form Feed
(000D)                  0035 CR:     EQU    13              ;Carriage Return
(000E)                  0036 SO:     EQU    14              ;Skip Over
(000F)                  0037 SI:     EQU    15              ;Clear top line
(007F)                  0038 RB:     EQU    127             ;Rub Out
                        0039 ;
(FFFF)                  0040 N$I:    EQU    0FFFFH          ;Not implemented
                        0041 ;
                        0042 ;RAM pointers in FDC-I Monitor System RAM.
(FF9A)                  0043 VPTR:   EQU    0FF80H+26       ;Video pointer
                        0044 ;
0000                    0045         ORG    0E800H
E800  C312E8            0046         JP     VSD             ;Video screen driver
E803  C3B0E8            0047         JP     PRTIME          ;Print time routine
E806  C3FFFF            0048         JP     N$I             ;Save for future expansion
E809  C3FFFF            0049         JP     N$I
E80C  C3FFFF            0050         JP     N$I
E80F  C3FFFF            0051         JP     N$I
                        0052 ;
```

CROMEMCO CDOS Z80 ASSEMBLER version 02.15 PAGE 0002
Video Utility Package for FDC-I Monitor

```
                0053 ;VSD is the video screen driver.
E812  F5        0054 VSD:   PUSH    AF
E813  C5        0055        PUSH    BC
E814  D5        0056        PUSH    DE
E815  E5        0057        PUSH    HL
E816  E67F      0058        AND     7FH
E818  FE0C      0059        CP      FF
E81A  284C      0060        JR      Z,FFD
E81C  FE0F      0061        CP      SI
E81E  2857      0062        JR      Z,CTLN
E820  2A9AFF    0063        LD      HL,(VPTR)       ;Get VSD pointer
E823  CBBE      0064        RES     7,(HL)          ;Reverse background
E825  FE0B      0065        CP      HM
E827  285D      0066        JR      Z,HOME
E829  FE0D      0067        CP      CR
E82B  285E      0068        JR      Z,CRET
E82D  FE0A      0069        CP      LF
E82F  2860      0070        JR      Z,LFD
E831  FE09      0071        CP      HT
E833  2862      0072        JR      Z,TAB
E835  FE08      0073        CP      BS
E837  2869      0074        JR      Z,BSP
E839  FE7F      0075        CP      RB
E83B  2865      0076        JR      Z,BSP
E83D  FE0E      0077        CP      SO
E83F  2801      0078        JR      Z,VD1
E841  77        0079        LD      (HL),A          ;Put char on screen
E842  23        0080 VD1:   INC     HL
E843  7C        0081 VD2:   LD      A,H
E844  FED0      0082        CP      [BSCR+1]/256    ;End of screen?
E846  3816      0083        JR      C,VD3           ;No
E848  7D        0084        LD      A,L
E849  2180CC    0085        LD      HL,TSCR+LINE    ;Scroll screen
E84C  1140CC    0086        LD      DE,TSCR
E84F  018003    0087        LD      BC,[NLN-1]*LINE
E852  EDB0      0088        LDIR
E854  012040    0089        LD      BC,4020H
E857  2B        0090 CBLN:  DEC     HL              ;Clear bottom line
E858  71        0091        LD      (HL),C
E859  10FC      0092        DJNZ    CBLN
E85B  C6C0      0093        ADD     0C0H
E85D  6F        0094        LD      L,A             ;Restore L
E85E  CBFE      0095 VD3:   SET     7,(HL)          ;Reverse background
E860  229AFF    0096        LD      (VPTR),HL       ;Save pointer
E863  E1        0097 VD4:   POP     HL
E864  D1        0098        POP     DE
E865  C1        0099        POP     BC
E866  F1        0100        POP     AF
E867  C9        0101        RET
E868  21FFCF    0102 FFD:   LD      HL,BSCR         ;Form feed, clear screen
E86B  3620      0103        LD      (HL),' '
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0003
Video Utility Package for FDC-I Monitor

E86D  11FECF      0104          LD      DE,BSCR-1
E870  01BF03      0105          LD      BC,SCR
E873  EDB8        0106          LDDR
E875  18E7        0107          JR      VD3
E877  2100CC      0108 CTLN:    LD      HL,VSL          ;Clear top line of screen
E87A  3620        0109          LD      (HL),' '
E87C  1101CC      0110          LD      DE,VSL+1
E87F  013F00      0111          LD      BC,LINE-1
E882  EDB0        0112          LDIR
E884  18DD        0113          JR      VD4
E886  2140CC      0114 HOME:    LD      HL,TSCR         ;Home cursor to top of screen
E889  18D3        0115          JR      VD3
E88B  7D          0116 CRET:    LD      A,L             ;Carriage return
E88C  E6C0        0117          AND     0C0H
E88E  6F          0118          LD      L,A
E88F  18CD        0119          JR      VD3
E891  014000      0120 LFD:     LD      BC,LINE
E894  09          0121          ADD     HL,BC
E895  18AC        0122          JR      VD2
E897  7D          0123 TAB:     LD      A,L             ;Horizontal Tab
E898  E6F8        0124          AND     0F8H
E89A  C608        0125          ADD     8
E89C  6F          0126          LD      L,A
E89D  30A4        0127          JR      NC,VD2
E89F  24          0128          INC     H
E8A0  18A1        0129          JR      VD2
E8A2  1140CC      0130 BSP:     LD      DE,TSCR         ;Back Space
E8A5  EB          0131          EX      DE,HL
E8A6  ED52        0132          SBC     HL,DE
E8A8  EB          0133          EX      DE,HL
E8A9  28B3        0134          JR      Z,VD3
E8AB  2B          0135          DEC     HL
E8AC  3620        0136          LD      (HL),' '
E8AE  18AE        0137          JR      VD3
                  0138 ;
                  0139 ;PRTIME prints the date and time.  It uses no subroutines or
                  0140 ;relative jumps to make execution as fast as possible.
E8B0  C5          0141 PRTIME:  PUSH    BC
E8B1  D5          0142          PUSH    DE
E8B2  1110CC      0143          LD      DE,DDL
E8B5  EB          0144          EX      DE,HL
E8B6  1A          0145          LD      A,(DE)          ;Get seconds, convert
E8B7  01003A      0146          LD      BC,':'*256+0    ;to decimal and put
E8BA  D60A        0147 PRT0:    SUB     10              ;on screen
E8BC  DAC3E8      0148          JP      C,PRT1
E8BF  0C          0149          INC     C
E8C0  C3BAE8      0150          JP      PRT0
E8C3  C63A        0151 PRT1:    ADD     30H+10
E8C5  77          0152          LD      (HL),A
E8C6  2B          0153          DEC     HL
E8C7  3E30        0154          LD      A,30H
```

```
E8C9  81          0155          ADD     C
E8CA  77          0156          LD      (HL),A
E8CB  2B          0157          DEC     HL
E8CC  70          0158          LD      (HL),B
E8CD  2B          0159          DEC     HL
E8CE  13          0160          INC     DE
E8CF  1A          0161          LD      A,(DE)          ;Get minutes, convert
E8D0  0E00        0162          LD      C,0             ;to decimal and put
E8D2  D60A        0163 PRT2:    SUB     10              ;on screen
E8D4  DADBE8      0164          JP      C,PRT3
E8D7  0C          0165          INC     C
E8D8  C3D2E8      0166          JP      PRT2
E8DB  C63A        0167 PRT3:    ADD     30H+10
E8DD  77          0168          LD      (HL),A
E8DE  2B          0169          DEC     HL
E8DF  3E30        0170          LD      A,30H
E8E1  81          0171          ADD     C
E8E2  77          0172          LD      (HL),A
E8E3  2B          0173          DEC     HL
E8E4  70          0174          LD      (HL),B
E8E5  2B          0175          DEC     HL
E8E6  13          0176          INC     DE
E8E7  1A          0177          LD      A,(DE)          ;Get hours, convert
E8E8  010020      0178          LD      BC,' '*256+0    ;to decimal and put
E8EB  D60A        0179 PRT4:    SUB     10              ;on screen
E8ED  DAF4E8      0180          JP      C,PRT5
E8F0  0C          0181          INC     C
E8F1  C3EBE8      0182          JP      PRT4
E8F4  C63A        0183 PRT5:    ADD     30H+10
E8F6  77          0184          LD      (HL),A
E8F7  2B          0185          DEC     HL
E8F8  3E30        0186          LD      A,30H
E8FA  81          0187          ADD     C
E8FB  77          0188          LD      (HL),A
E8FC  2B          0189          DEC     HL
E8FD  70          0190          LD      (HL),B
E8FE  2B          0191          DEC     HL
E8FF  13          0192          INC     DE
E900  1A          0193          LD      A,(DE)          ;Get year, convert
E901  01002F      0194          LD      BC,'/'*256+0    ;to decimal and put
E904  D60A        0195 PRT6:    SUB     10              ;on screen
E906  DA0DE9      0196          JP      C,PRT7
E909  0C          0197          INC     C
E90A  C304E9      0198          JP      PRT6
E90D  C63A        0199 PRT7:    ADD     30H+10
E90F  77          0200          LD      (HL),A
E910  2B          0201          DEC     HL
E911  3E30        0202          LD      A,30H
E913  81          0203          ADD     C
E914  77          0204          LD      (HL),A
E915  2B          0205          DEC     HL
```

```
E916  70          0206          LD      (HL),B
E917  2B          0207          DEC     HL
E918  13          0208          INC     DE
E919  1A          0209          LD      A,(DE)      ;Get day, convert
E91A  0E00        0210          LD      C,0         ;to decimal and put
E91C  D60A        0211 PRT8:    SUB     10          ;on screen
E91E  DA25E9      0212          JP      C,PRT9
E921  0C          0213          INC     C
E922  C31CE9      0214          JP      PRT8
E925  C63A        0215 PRT9:    ADD     30H+10
E927  77          0216          LD      (HL),A
E928  2B          0217          DEC     HL
E929  3E30        0218          LD      A,30H
E92B  81          0219          ADD     C
E92C  77          0220          LD      (HL),A
E92D  2B          0221          DEC     HL
E92E  70          0222          LD      (HL),B
E92F  2B          0223          DEC     HL
E930  13          0224          INC     DE
E931  1A          0225          LD      A,(DE)      ;Get month, convert
E932  0E00        0226          LD      C,0         ;to decimal and put
E934  D60A        0227 PRT10:   SUB     10          ;on screen
E936  DA3DE9      0228          JP      C,PRT11
E939  0C          0229          INC     C
E93A  C334E9      0230          JP      PRT10
E93D  C63A        0231 PRT11:   ADD     30H+10
E93F  77          0232          LD      (HL),A
E940  2B          0233          DEC     HL
E941  3E30        0234          LD      A,30H
E943  81          0235          ADD     C
E944  77          0236          LD      (HL),A
E945  D1          0237          POP     DE
E946  C1          0238          POP     BC
E947  C9          0239          RET
                  0240 ;


Errors            0
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001
Format Program for FDC-I

                        0002 ;FORMAT is a program that formats 8 inch diskettes in a
                        0003 ;IBM compatible form using Teletek's FDC-I.  It can format
                        0004 ;in either single or double density formats.
                        0005 ;It runs on any CP/M system.
                        0006 ;
                        0007 ;Copyright (C) 1979, Teletek Enterprises, Inc.
                        0008 ;
                        0009 ;Revision date:  4-7-80  Release 2.0
                        0010 ;For release 3 of the FDC-I monitor, DO NOT use with release 2.
                        0011 ;
                        0012 ;Written by Aram Perez
                        0013 ;
     (0002)             0014 PCH:     EQU     2               ;CP/M print char
     (0009)             0015 PRINT:   EQU     9               ; "     "   string
     (000A)             0016 RDBUFF:  EQU     10              ; "   read buffer
0000                    0017          ORG     100H
     (0100)             0018 STAK:    EQU     $               ;Set stack at 100H
0100 C36101             0019          JP      START
0103 436F7079           0020          DB      'Copyright (C) 1979, Teletek '
     72696768
     74202843
     29203139
     37392C20
     54656C65
     74656B20
                        0021 ;The following routines allow direct access to the same
                        0022 ;routines in the FDC-I monitor.  For these routines to work
                        0023 ;the address for the FDC-I monitor must be put at 40H by
                        0024 ;the BIOS, otherwise disaster!!!!
011F C32100             0025 STMODE:  JP      11*3            ;Set read/write mode
0122 C32400             0026 STHDR:   JP      12*3            ;Set head/drive
0125 C32700             0027 STSECT:  JP      13*3            ;Set sector
0128 C32A00             0028 RECAL:   JP      14*3            ;Recalibrate
012B C32D00             0029 SEEK:    JP      15*3            ;Set/seek track
012E C33000             0030 READ:    JP      16*3            ;Read from disk
0131 C33300             0031 WRITE:   JP      17*3            ;Write to disk
0134 C33600             0032 RTST0:   JP      18*3            ;Return addr of ST0
0137 C34E00             0033 DRVST:   JP      26*3            ;Get drive status
                        0034 ;
013A 0E09               0035 PSTR:    LD      C,PRINT         ;Print string
013C C30500             0036          JP      5
013F CD3A01             0037 INSTR:   CALL    PSTR            ;Inputs a string
0142 114307             0038          LD      DE,BUFF
0145 3E02               0039          LD      A,2
0147 12                 0040          LD      (DE),A
0148 0E0A               0041          LD      C,RDBUFF
014A CD0500             0042          CALL    5
014D 3E0A               0043          LD      A,10
014F CD5B01             0044          CALL    0CH
0152 3A4407             0045          LD      A,(BUFF+1)
0155 B7                 0046          OR      A
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0002
Format Program for FDC-I

0156  C8          0047          RET     Z
0157  3A4507      0048          LD      A,(BUFF+2)
015A  C9          0049          RET
015B  0E02        0050 OCH:     LD      C,PCH           ;Print char
015D  5F          0051          LD      E,A
015E  C30500      0052          JP      5
                  0053 ;
0161  310001      0054 START:   LD      SP,STAK
0164  212101      0055          LD      HL,STMODE+2
0167  110300      0056          LD      DE,3
016A  0609        0057          LD      B,9
016C  3A4000      0058          LD      A,(40H)         ;Get addr of FDC-I monitor
016F  77          0059 STRT:    LD      (HL),A
0170  19          0060          ADD     HL,DE           ;Set up correct jmp addr
0171  10FC        0061          DJNZ    STRT
0173  118D03      0062          LD      DE,FMT
0176  CD3A01      0063          CALL    PSTR
0179  DD213A07    0064          LD      IX,CMD
017D  DD36031A    0065          LD      (IX+3),26       ;SC, no. of sectors/track
0181  DD3605E5    0066          LD      (IX+5),0E5H     ;D, filler byte
0185  214907      0067          LD      HL,VBUFF
0188  114A07      0068          LD      DE,VBUFF+1
018B  01FF00      0069          LD      BC,255
018E  EDB0        0070          LDIR
0190  11CD03      0071 CHDR:    LD      DE,EDRV         ;Input drive
0193  CD3F01      0072          CALL    INSTR
0196  E65F        0073          AND     5FH             ;Convert to upper case
0198  32E304      0074          LD      (DRIVE),A
019B  D641        0075          SUB     'A'
019D  323B07      0076          LD      (DRV),A         ;Save drive
01A0  FE00        0077          CP      0
01A2  38EC        0078          JR      C,CHDR
01A4  FE04        0079          CP      4
01A6  30E8        0080          JR      NC,CHDR
01A8  11EE03      0081 INSD:    LD      DE,ESD          ;Input side (head)
01AB  CD3F01      0082          CALL    INSTR
01AE  32EB04      0083          LD      (SIDE),A
01B1  D630        0084          SUB     30H
01B3  324207      0085          LD      (HEAD),A
01B6  FE00        0086          CP      0
01B8  38EE        0087          JR      C,INSD
01BA  FE02        0088          CP      2
01BC  30EA        0089          JR      NC,INSD
01BE  87          0090          ADD     A
01BF  87          0091          ADD     A
01C0  DDB601      0092          OR      (IX+1)
01C3  323B07      0093          LD      (DRV),A
01C6  CD2201      0094          CALL    STHDR           ;Set head/drive
01C9  310001      0095 POPT:    LD      SP,STAK         ;Set stack
01CC  110804      0096          LD      DE,OPT
01CF  CD3F01      0097          CALL    INSTR           ;Input an option
```

CROMEMCO CDOS Z80 ASSEMBLER version 02.15 PAGE 0003
Format Program for FDC-I

```
01D2  32D804      0098          LD      (OPTION),A      ;Get & save option
01D5  D630        0099          SUB     '0'
01D7  3D          0100          DEC     A
01D8  CA0000      0101          JP      Z,0             ;Return to CP/M
01DB  3D          0102          DEC     A
01DC  28B2        0103          JR      Z,CHDR          ;Change head/drive
01DE  3D          0104          DEC     A
01DF  28C7        0105          JR      Z,INSD
01E1  3D          0106          DEC     A
01E2  2808        0107          JR      Z,FS            ;Single density
01E4  3D          0108          DEC     A
01E5  281F        0109          JR      Z,FD            ;Double density
01E7  3D          0110          DEC     A
01E8  2836        0111          JR      Z,FDS           ;Double density, trk 0 single
01EA  18DD        0112          JR      POPT
01EC  DD36000D    0113 FS:      LD      (IX+0),0DH      ;Format single density
01F0  DD360200    0114          LD      (IX+2),0        ;N
01F4  DD36041B    0115          LD      (IX+4),1BH      ;GPL
01F8  DD36064D    0116          LD      (IX+6),77       ;Last track to format+1
01FC  CD4F02      0117 FS1:     CALL    FORMAT
01FF  38FB        0118          JR      C,FS1
0201  CD6C03      0119          CALL    DONE
0204  18F6        0120          JR      FS1
0206  DD36004D    0121 FD:      LD      (IX+0),4DH      ;Format double density
020A  DD360201    0122          LD      (IX+2),1        ;N
020E  DD360436    0123          LD      (IX+4),36H      ;GPL
0212  DD36064D    0124          LD      (IX+6),77       ;Last track to format+1
0216  CD4F02      0125 FD1:     CALL    FORMAT
0219  38FB        0126          JR      C,FD1
021B  CD6C03      0127          CALL    DONE
021E  18F6        0128          JR      FD1
0220  DD36000D    0129 FDS:     LD      (IX+0),0DH      ;Format double density, with
0224  DD360200    0130          LD      (IX+2),0        ;N      1st trk single density
0228  DD36041B    0131          LD      (IX+4),1BH      ;GPL
022C  DD360601    0132          LD      (IX+6),1        ;Last track to format+1
0230  CD4F02      0133 FDS1:    CALL    FORMAT
0233  38FB        0134          JR      C,FDS1          ;Jp if error
0235  DD36004D    0135          LD      (IX+0),4DH      ;Format double density
0239  DD360201    0136          LD      (IX+2),1        ;N
023D  DD360436    0137          LD      (IX+4),36H      ;GPL
0241  DD36064D    0138          LD      (IX+6),77       ;Last track to format+1
0245  CD9302      0139          CALL    FRM4
0248  38D6        0140          JR      C,FDS
024A  CD6C03      0141          CALL    DONE
024D  18D1        0142          JR      FDS
024F  11BA04      0143 FORMAT:  LD      DE,RDY          ;First wait till user ready
0252  CD3F01      0144          CALL    INSTR
0255  F620        0145          OR      20H             ;Convert to lower case
0257  FE79        0146          CP      'y'
0259  C2C901      0147          JP      NZ,POPT         ;User is not ready
025C  CD3701      0148          CALL    DRVST           ;First check status
```

```
025F  CB6F      0149          BIT   5,A       ;Drive ready?
0261  2008      0150          JR    NZ,FRM0   ;Yes
0263  11F504    0151          LD    DE,DNR
0266  CD3A01    0152          CALL  PSTR
0269  18E4      0153          JR    FORMAT
026B  CB77      0154 FRM0:    BIT   6,A       ;Is disk write protected?
026D  2808      0155          JR    Z,FRM1    ;No, continue
026F  112605    0156          LD    DE,PROT
0272  CD3A01    0157          CALL  PSTR
0275  18D8      0158          JR    FORMAT
0277  0605      0159 FRM1:    LD    B,5
0279  CD2801    0160 FRM2:    CALL  RECAL     ;Now recalibrate
027C  2815      0161          JR    Z,FRM4    ;Jp if no errors
027E  1009      0162          DJNZ  FRM3
0280  11B705    0163          LD    DE,UTR    ;Unable to recalibrate
0283  CD3A01    0164          CALL  PSTR
0286  C3C901    0165          JP    POPT
0289  C5        0166 FRM3:    PUSH  BC
028A  119105    0167          LD    DE,RCER   ;Recalibrate error
028D  CD3A01    0168          CALL  PSTR
0290  C1        0169          POP   BC
0291  18E6      0170          JR    FRM2      ;Try again
0293  324107    0171 FRM4:    LD    (TRK),A   ;Save track
0296  0605      0172          LD    B,5
0298  CD2B01    0173 FRM5:    CALL  SEEK      ;Seek track
029B  2812      0174          JR    Z,FRM7
029D  1006      0175          DJNZ  FRM6
029F  110106    0176          LD    DE,UTS    ;Unable to seek
02A2  C33A01    0177          JP    PSTR
02A5  C5        0178 FRM6:    PUSH  BC
02A6  11E205    0179          LD    DE,SKER   ;Seek error, try again
02A9  CD3A01    0180          CALL  PSTR
02AC  C1        0181          POP   BC
02AD  18E9      0182          JR    FRM5
02AF  0602      0183 FRM7     LD    B,2       ;Two retrys max
02B1  C5        0184 FRM8:    PUSH  BC
02B2  CD3A03    0185          CALL  ACTFRM    ;Do actual format
02B5  C1        0186          POP   BC
02B6  2820      0187          JR    Z,FRM10
02B8  1014      0188          DJNZ  FRM9
02BA  112B06    0189          LD    DE,FMER0  ;Can't format
02BD  CD3A01    0190          CALL  PSTR
02C0  3A4107    0191          LD    A,(TRK)
02C3  CD7203    0192          CALL  PDEC      ;Print track number
02C6  115D06    0193          LD    DE,FMER1
02C9  CD3A01    0194          CALL  PSTR
02CC  37        0195          SCF             ;Set error flag
02CD  C9        0196          RET
02CE  C5        0197 FRM9:    PUSH  BC
02CF  117406    0198          LD    DE,FMER2  ;Format error, try again
02D2  CD3A01    0199          CALL  PSTR
```

CROMEMCO CDOS Z80 ASSEMBLER version 02.15 PAGE 0005
Format Program for FDC-I

```
02D5  C1        0200          POP     BC
02D6  18D9      0201          JR      FRM8
02D8  3A3A07    0202 FRM10:   LD      A,(CMD)
02DB  CD1F01    0203          CALL    STMODE      ;Set mode
02DE  3E08      0204          LD      A,8
02E0  CD2501    0205          CALL    STSECT
02E3  214907    0206          LD      HL,VBUFF
02E6  1E01      0207          LD      E,1
02E8  CD3101    0208          CALL    WRITE       ;Write 1 sector
02EB  3E01      0209          LD      A,1
02ED  CD2501    0210          CALL    STSECT
02F0  0602      0211          LD      B,2         ;Try reading 2 times
02F2  C5        0212 FRM11:   PUSH    BC
02F3  214907    0213          LD      HL,VBUFF
02F6  1E1A      0214          LD      E,26        ;Read all 26 sectors
02F8  CD2E01    0215          CALL    READ        ;& verify if formated
02FB  C1        0216          POP     BC
02FC  200B      0217          JR      NZ,FRM12    ;Jp if no errors
02FE  3A4107    0218          LD      A,(TRK)
0301  3C        0219          INC     A
0302  DDBE06    0220          CP      (IX+6)
0305  C29302    0221          JP      NZ,FRM4
0308  C9        0222          RET
0309  1025      0223 FRM12:   DJNZ    FRM13
030B  11BB06    0224          LD      DE,BAD      ;Bad sector
030E  CD3A01    0225          CALL    PSTR
0311  CD3401    0226          CALL    RTST0
0314  110500    0227          LD      DE,5
0317  19        0228          ADD     HL,DE
0318  7E        0229          LD      A,(HL)
0319  CD7203    0230          CALL    PDEC        ;Print bad sector no
031C  11D706    0231          LD      DE,BAD1
031F  CD3A01    0232          CALL    PSTR
0322  3A4107    0233          LD      A,(TRK)
0325  CD7203    0234          CALL    PDEC        ;Print track
0328  11E206    0235          LD      DE,BAD2
032B  CD3A01    0236          CALL    PSTR
032E  37        0237          SCF                 ;Set error flag
032F  C9        0238          RET
0330  C5        0239 FRM13:   PUSH    BC
0331  119C06    0240          LD      DE,RDER
0334  CD3A01    0241          CALL    PSTR
0337  C1        0242          POP     BC
0338  18B8      0243          JR      FRM11
033A  0606      0244 ACTFRM:  LD      B,6         ;Actual formatting routine
033C  213A07    0245          LD      HL,CMD
033F  DB0C      0246 ACT0:    IN      A,(12)
0341  07        0247          RLCA                ;Wait for RQM
0342  30FB      0248          JR      NC,ACT0
0344  7E        0249          LD      A,(HL)      ;Send command, head/drive,
0345  D30D      0250          OUT     (13),A      ;N, SC, GPL and D
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0006
Format Program for FDC-I

0347  23          0251          INC     HL
0348  10F5        0252          DJNZ    ACT0
034A  DD5602      0253          LD      D,(IX+2)        ;Get N
034D  2A4107      0254          LD      HL,(TRK)        ;Get trk (L) & head (H)
0350  01011A      0255          LD      BC,26*256+1
0353  7D          0256 ACT1:    LD      A,L             ;Send C
0354  D318        0257          OUT     (24),A
0356  D310        0258          OUT     (16),A
0358  7C          0259          LD      A,H             ;Send H
0359  D318        0260          OUT     (24),A
035B  D310        0261          OUT     (16),A
035D  79          0262          LD      A,C             ;Send R
035E  D318        0263          OUT     (24),A
0360  D310        0264          OUT     (16),A
0362  7A          0265          LD      A,D             ;Send N
0363  D318        0266          OUT     (24),A
0365  D310        0267          OUT     (16),A
0367  0C          0268          INC     C
0368  10E9        0269          DJNZ    ACT1
036A  18FE        0270 WAIT:    JR      WAIT            ;Wait till done formatting
                  0271 ;
036C  110D07      0272 DONE:    LD      DE,FCOM         ;Function complete
036F  C33A01      0273          JP      PSTR
0372  0E00        0274 PDEC:    LD      C,0             ;Print A in decimal
0374  D60A        0275 PD0:     SUB     10
0376  3803        0276          JR      C,PD1
0378  0C          0277          INC     C
0379  18F9        0278          JR      PD0
037B  C63A        0279 PD1:     ADD     '0'+10
037D  47          0280          LD      B,A
037E  79          0281          LD      A,C
037F  B7          0282          OR      A
0380  2807        0283          JR      Z,PD2           ;Don't print leading zero
0382  C630        0284          ADD     '0'
0384  C5          0285          PUSH    BC
0385  CD5B01      0286          CALL    OCH
0388  C1          0287          POP     BC
0389  78          0288 PD2:     LD      A,B
038A  C35B01      0289          JP      OCH
038D  0D0A5465    0290 FMT:     DB      13,10,'Teletek''','s FDC-I Format '
      6C657465
      6B277320
      4644432D
      4920466F
      726D6174
      20
03A6  50726F67    0291          DB      'Program [Release 2.0]'
      72616D20
      5B52656C
      65617365
      20322E30
```

CROMEMCO CDOS Z80 ASSEMBLER version 02.15 PAGE 0007
Format Program for FDC-I

```
                5D
03BB  0D0A4350  0292          DB      13,10,'CP/M Version.',13,10,'$'
      2F4D2056
      65727369
      6F6E2E0D
      0A24
03CD  456E7465  0293 EDRV:    DB      'Enter the drive (A, B, C or D)? $'
      72207468
      65206472
      69766520
      28412C20
      422C2043
      206F7220
      44293F20
      24
03EE  456E7465  0294 ESD:     DB      'Enter the side (0 or 1)? $'
      72207468
      65207369
      64652028
      30206F72
      2031293F
      2024
0408  0D0A4F70  0295 OPT:     DB      13,10,'Options for FORMAT:',13,10
      74696F6E
      7320666F
      7220464F
      524D4154
      3A0D0A
041F  312E2045  0296          DB      '1. Exit',13,10,'2. Change drive and side'
      7869740D
      0A322E20
      4368616E
      67652064
      72697665
      20616E64
      20736964
      65
0440  0D0A332E  0297          DB      13,10,'3. Change side only'
      20436861
      6E676520
      73696465
      206F6E6C
      79
0455  0D0A342E  0298          DB      13,10,'4. Single density',13,10
      2053696E
      676C6520
      64656E73
      6974790D
      0A
046A  352E2044  0299          DB      '5. Double density',13,10
      6F75626C
```

```
                65206465
                6E736974
                790D0A
047D  362E2044  0300          DB      '6. Double density (track 0 single'
      6F75626C
      65206465
      6E736974
      79202874
      7261636B
      20302073
      696E676C
      65
049E  2064656E  0301          DB      ' density)',13,10,'Enter option #: $'
      73697479
      290D0A45
      6E746572
      206F7074
      696F6E20
      233A2024
04BA  0D0A5265  0302 RDY:     DB      13,10,'Ready to format with option '
      61647920
      746F2066
      6F726D61
      74207769
      7468206F
      7074696F
      6E20
04D8  (0001)    0303 OPTION:  DS      1               ;Option
04D9  206F6E20  0304          DB      ' on drive '
      64726976
      6520
04E3  (0001)    0305 DRIVE:   DS      1               ;Drive
04E4  2C207369  0306          DB      ', side '
      646520
04EB  (0001)    0307 SIDE:    DS      1               ;Side (head)
04EC  2028592F  0308          DB      ' (Y/N)? $'
      4E293F20
      24
04F5  0D0A4472  0309 DNR:     DB      13,10,'Drive is not ready (make sure door is '
      69766520
      6973206E
      6F742072
      65616479
      20286D61
      6B652073
      75726520
      646F6F72
      20697320
051D  636C6F73  0310          DB      'closed).$'
      6564292E
      24
```

CROMEMCO CDOS Z80 ASSEMBLER version 02.15 PAGE 0009
Format Program for FDC-I

```
0526 0D0A2A2A      0311 PROT:   DB      13,10,'*** Fatal Error:   diskette is '
     2A204661
     74616C20
     4572726F
     723A2020
     20646973
     6B657474
     65206973
     20
0547 77726974      0312         DB      'write protected.',13,10,'Make sure "write '
     65207072
     6F746563
     7465642E
     0D0A4D61
     6B652073
     75726520
     22777269
     746520
056A 70726F74      0313         DB      'protect" notch on diskette is covered.$'
     65637422
     206E6F74
     6368206F
     6E206469
     736B6574
     74652069
     7320636F
     76657265
     642E24
0591 0D0A5265      0314 RCER:   DB      13,10,'Recalibrate error!  Trying again...$'
     63616C69
     62726174
     65206572
     726F7221
     20205472
     79696E67
     20616761
     696E2E2E
     2E24
05B7 0D0A2A2A      0315 UTR:    DB      13,10,'*** Fatal Error:  unable to '
     2A204661
     74616C20
     4572726F
     723A2020
     756E6162
     6C652074
     6F20
05D5 72656361      0316         DB      'recalibrate.$'
     6C696272
     6174652E
     24
05E2 0D0A5365      0317 SKER:   DB      13,10,'Seek error!  Trying again...$'
```

```
                656B2065
                72726F72
                21202054
                7279696E
                67206167
                61696E2E
                2E2E24
0601  0D0A2A2A  0318 UTS:   DB      13,10,'*** Fatal Error:  unable to '
      2A204661
      74616C20
      4572726F
      723A2020
      756E6162
      6C652074
      6F20
061F  7365656B  0319        DB      'seek track.$'
      20747261
      636B2E24
062B  0D0A2A2A  0320 FMER0: DB      13,10,'*** Fatal Error:  can not format '
      2A204661
      74616C20
      4572726F
      723A2020
      63616E20
      6E6F7420
      666F726D
      617420
064E  74726163  0321        DB      'track number: $'
      6B206E75
      6D626572
      3A2024
065D  0D0A5472  0322 FMER1: DB      13,10,'Try another diskette.'
      7920616E
      6F746865
      72206469
      736B6574
      74652E
0674  0D0A466F  0323 FMER2: DB      13,10,'Formating error, trying once again...$'
      726D6174
      696E6720
      6572726F
      722C2074
      7279696E
      67206F6E
      63652061
      6761696E
      2E2E2E24
069C  0D0A5265  0324 RDER:  DB      13,10,'Read error!  Trying again...$'
      61642065
      72726F72
      21202054
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0011
Format Program for FDC-I

           7279696E
           67206167
           61696E2E
           2E2E24
06BB       0D0A2A2A      0325 BAD:    DB      13,10,'*** Fatal Error:  sector $'
           2A204661
           74616C20
           4572726F
           723A2020
           73656374
           6F722024
06D7       206F6E20      0326 BAD1:   DB      ' on track $'
           74726163
           6B2024
06E2       20697320      0327 BAD2:   DB      ' is bad.',13,10,'Try again or '
           6261642E
           0D0A5472
           79206167
           61696E20
           6F7220
06F9       616E6F74      0328         DB      'another diskette.',13,10,'$'
           68657220
           6469736B
           65747465
           2E0D0A24
070D       0D0A4675      0329 FCOM:   DB      13,10,'Function complete:  diskette is '
           6E637469
           6F6E2063
           6F6D706C
           6574653A
           20206469
           736B6574
           74652069
           7320
072F       666F726D      0330         DB      'formatted.$'
           61747465
           642E24
                         0331 ;
                         0332 ;
073A       (0001)        0333 CMD:    DS      1
073B       (0001)        0334 DRV:    DS      1
073C       (0001)        0335 N:      DS      1
073D       (0001)        0336 SC:     DS      1
073E       (0001)        0337 GPL:    DS      1
073F       (0001)        0338 FB:     DS      1
0740       (0001)        0339 LTRK:   DS      1
0741       (0001)        0340 TRK:    DS      1
0742       (0001)        0341 HEAD:   DS      1
0743       (0004)        0342 BUFF:   DS      4
0747       (0001)        0343 FTRY:   DS      1               ;Format retrys
0748       (0001)        0344 RTRY:   DS      1               ;Read retrys
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0012
Format Program for FDC-I

0749  (1A00)        0345 VBUFF:  DS      26*256          ;Verify buffer
2149  (0000)        0346         END

Errors              0
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001
CP/M 1.4 Cold Start Loader for FDC-I

                        0002 ;Version 9-11-79
                        0003 ;
                        0004 ;FDC-I monitor interfacing vectors:
      (E000)            0005 FDC     EQU     0E000H
      (E00F)            0006 OUTCHR  EQU     FDC+15          ;Output char
      (E027)            0007 STSECT  EQU     FDC+27H         ;Set the starting sector
      (E02D)            0008 SEEK    EQU     FDC+2DH         ;Seek a track
      (E030)            0009 READ    EQU     FDC+30H         ;Read floppy disk
                        0010 ;
                        0011 ;CP/M parameters:
      (0030)            0012 MEM     EQU     48              ;Memory size in K
      (8000)            0013 BASE    EQU     (MEM-16)*1024
      (A900)            0014 CCP     EQU     2900H+BASE
      (BE00)            0015 BIOS    EQU     3E00H+BASE
                        0016 ;
0000                    0017         ORG     0
0000  310001            0018         LD      SP,100H         ;Set up the stack
0003  3E02              0019         LD      A,2
0005  CD27E0            0020         CALL    STSECT
0008  2100A9            0021         LD      HL,CCP
000B  1E19              0022         LD      E,25
000D  CD30E0            0023         CALL    READ            ;Read trk 0, sctr's 2-26
0010  200F              0024         JR      NZ,CSERR
0012  3C                0025         INC     A
0013  CD27E0            0026         CALL    STSECT
0016  CD2DE0            0027         CALL    SEEK            ;Seek track 1
0019  1E15              0028         LD      E,21
001B  CD30E0            0029         CALL    READ            ;Read trk 1, sctr's 1-21
001E  CA00BE            0030         JP      Z,BIOS
0021  212D00            0031 CSERR:  LD      HL,ERR
0024  7E                0032 PMSG:   LD      A,(HL)
0025  CD0FE0            0033         CALL    OUTCHR
0028  23                0034         INC     HL
0029  B7                0035         OR      A
002A  F8                0036         RET     M
002B  18F7              0037         JR      PMSG
                        0038 ;
002D  0D0A4361          0039 ERR:    DB      13,10,'Can not load system','!'+80H
      6E206E6F
      74206C6F
      61642073
      79737465
      6DA1
                        0040 ;
0043  (0000)            0041         END

Errors                  0
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                PAGE 0001
CP/M 1.4 BIOS for FDC-I

                        0002 ;
                        0003 ;Copyright (C) 1979, Teletek Enterprises, Inc.
                        0004 ;
                        0005 ;Revision date:  8-3-80  Release 1.3
                        0006 ;
                        0007 ;FDC-I monitor interfacing vectors:
     (E000)             0008 FDC:    EQU     0E000H          ;Monitor location
     (E018)             0009 CONST:  EQU     FDC+18H         ;Input device status
     (E009)             0010 ICH:    EQU     FDC+9           ;Input routine
     (E00F)             0011 OCH:    EQU     FDC+15          ;Output routine
     (E012)             0012 LOUT:   EQU     FDC+12H         ;Listing routine
     (E024)             0013 STDRV:  EQU     FDC+24H         ;Set drive
     (E027)             0014 STSECT: EQU     FDC+27H         ;Set sector
     (E02A)             0015 RECAL:  EQU     FDC+2AH         ;Recalibrate
     (E02D)             0016 SEEK:   EQU     FDC+2DH         ;Seek a track
     (E030)             0017 READF:  EQU     FDC+30H         ;Read a sector
     (E033)             0018 WRITF:  EQU     FDC+33H         ;Write a sector
                        0019 ;
                        0020 ;CP/M parameters:
     (0030)             0021 MEM:    EQU     48              ;Memory size in K
     (8000)             0022 BASE:   EQU     (MEM-16)*1024
     (A900)             0023 CCP:    EQU     2900H+BASE
     (B100)             0024 BDOS:   EQU     3100H+BASE
     (0004)             0025 CDRV:   EQU     4               ;Current drive
                        0026 ;
0000                    0027         ORG     40H
0040 (0001)             0028 NDRV:   DS      1               ;No. of drives
0041 (0001)             0029 RWOP:   DS      1               ;Read/Write flag
0042 (0001)             0030 CHC:    DS      1               ;Character count
0043 (0001)             0031 DEL:    DS      1               ;DELete flag
0044 (0001)             0032 RETRY:  DS      1               ;Error retry count
0045 (0002)             0033 DMA:    DS      2               ;DMA addr for R/W operation
                        0034 ;
0047                    0035         ORG     3E00H+BASE
                        0036 ;BIOS jump vectors:
BE00 C32DBE             0037         JP      BOOT
BE03 C383BE             0038 JWBT:   JP      WBOOT
BE06 C318E0             0039         JP      CONST
BE09 C3C9BE             0040         JP      CONIN
BE0C C3E1BE             0041         JP      CONOUT
BE0F C300BF             0042         JP      LIST
BE12 C300BF             0043         JP      PUNCH
BE15 C304BF             0044         JP      READER
BE18 C307BF             0045         JP      HOME
BE1B C321BF             0046         JP      SELDSK
BE1E C340BF             0047         JP      SETTRK
BE21 C34ABF             0048         JP      SETSEC
BE24 C34EBF             0049         JP      SETDMA
BE27 C353BF             0050         JP      READ
BE2A C357BF             0051         JP      WRITE
                        0052 ;
BE2D 217CBF             0053 BOOT:   LD      HL,SOMSG        ;Print sign on message
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0002
CP/M 1.4 BIOS for FDC-I

BE30  CD7ABE      0054          CALL    PMSG
BE33  21A7BF      0055 BT:      LD      HL,GNDR         ;Get # of drives
BE36  CD7ABE      0056          CALL    PMSG
BE39  CDC9BE      0057          CALL    CONIN
BE3C  CD0FE0      0058          CALL    OCH
BE3F  D631        0059          SUB     '1'
BE41  38F0        0060          JR      C,BT
BE43  FE04        0061          CP      4
BE45  30EC        0062          JR      NC,BT
BE47  324000      0063          LD      (NDRV),A        ;Save no. of drives
BE4A  97          0064          SUB     A
BE4B  320400      0065          LD      (CDRV),A        ;Set current drive to A:
BE4E  324200      0066          LD      (CHC),A         ;Reset char count
BE51  324300      0067          LD      (DEL),A         ;& DELete flag
BE54  3EC3        0068 SETUP:   LD      A,0C3H          ;Set up jump vectors
BE56  320000      0069          LD      (0),A
BE59  320500      0070          LD      (5),A
BE5C  2103BE      0071          LD      HL,JWBT
BE5F  220100      0072          LD      (1),HL
BE62  2106B1      0073          LD      HL,BDOS+6
BE65  220600      0074          LD      (6),HL
BE68  218000      0075          LD      HL,80H          ;Set init DMA addr
BE6B  224500      0076          LD      (DMA),HL
BE6E  3EE0        0077          LD      A,FDC/256       ;Save location of monitor
BE70  324000      0078          LD      (40H),A         ;for utility programs
BE73  3A0400      0079          LD      A,(CDRV)
BE76  4F          0080          LD      C,A
BE77  C300A9      0081          JP      CCP
                  0082 ;
BE7A  7E          0083 PMSG:    LD      A,(HL)          ;Print ASCII message
BE7B  23          0084          INC     HL
BE7C  CD0FE0      0085          CALL    OCH
BE7F  B7          0086          OR      A
BE80  F8          0087          RET     M               ;Exit if bit 7 set
BE81  18F7        0088          JR      PMSG
                  0089 ;
BE83  97          0090 WBOOT:   SUB     A               ;Boot from drive 0
BE84  CD24E0      0091          CALL    STDRV
BE87  318000      0092          LD      SP,80H          ;Set up stack
BE8A  CD2AE0      0093 WBT:     CALL    RECAL           ;Recalibrate drive
BE8D  2807        0094          JR      Z,WBT0
BE8F  CD18E0      0095          CALL    CONST
BE92  28F6        0096          JR      Z,WBT           ;Loop until no errors
BE94  1828        0097          JR      WBERR
BE96  CD2DE0      0098 WBT0:    CALL    SEEK            ;Set track 0
BE99  3E02        0099          LD      A,2
BE9B  CD27E0      0100          CALL    STSECT
BE9E  2100A9      0101          LD      HL,CCP
BEA1  1E19        0102          LD      E,25
BEA3  CD30E0      0103          CALL    READF           ;Read trk 0, sectors 2-24
BEA6  2016        0104          JR      NZ,WBERR
BEA8  3C          0105          INC     A
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0001
CP/M 1.4 BIOS for FDC-I

                        0002 ;
                        0003 ;Copyright (C) 1979, Teletek Enterprises, Inc.
                        0004 ;
                        0005 ;Revision date:  8-3-80  Release 1.3
                        0006 ;
                        0007 ;FDC-I monitor interfacing vectors:
     (E000)             0008 FDC:    EQU     0E000H          ;Monitor location
     (E018)             0009 CONST:  EQU     FDC+18H         ;Input device status
     (E009)             0010 ICH:    EQU     FDC+9           ;Input routine
     (E00F)             0011 OCH:    EQU     FDC+15          ;Output routine
     (E012)             0012 LOUT:   EQU     FDC+12H         ;Listing routine
     (E024)             0013 STDRV:  EQU     FDC+24H         ;Set drive
     (E027)             0014 STSECT: EQU     FDC+27H         ;Set sector
     (E02A)             0015 RECAL:  EQU     FDC+2AH         ;Recalibrate
     (E02D)             0016 SEEK:   EQU     FDC+2DH         ;Seek a track
     (E030)             0017 READF:  EQU     FDC+30H         ;Read a sector
     (E033)             0018 WRITF:  EQU     FDC+33H         ;Write a sector
                        0019 ;
                        0020 ;CP/M parameters:
     (0030)             0021 MEM:    EQU     48              ;Memory size in K
     (8000)             0022 BASE:   EQU     (MEM-16)*1024
     (A900)             0023 CCP:    EQU     2900H+BASE
     (B100)             0024 BDOS:   EQU     3100H+BASE
     (0004)             0025 CDRV:   EQU     4               ;Current drive
                        0026 ;
0000                    0027         ORG     40H
0040 (0001)             0028 NDRV:   DS      1               ;No. of drives
0041 (0001)             0029 RWOP:   DS      1               ;Read/Write flag
0042 (0001)             0030 CHC:    DS      1               ;Character count
0043 (0001)             0031 DEL:    DS      1               ;DELete flag
0044 (0001)             0032 RETRY:  DS      1               ;Error retry count
0045 (0002)             0033 DMA:    DS      2               ;DMA addr for R/W operation
                        0034 ;
0047                    0035         ORG     3E00H+BASE
                        0036 ;BIOS jump vectors:
BE00 C32DBE             0037         JP      BOOT
BE03 C383BE             0038 JWBT:   JP      WBOOT
BE06 C318E0             0039         JP      CONST
BE09 C3C9BE             0040         JP      CONIN
BE0C C3E1BE             0041         JP      CONOUT
BE0F C300BF             0042         JP      LIST
BE12 C300BF             0043         JP      PUNCH
BE15 C304BF             0044         JP      READER
BE18 C307BF             0045         JP      HOME
BE1B C321BF             0046         JP      SELDSK
BE1E C340BF             0047         JP      SETTRK
BE21 C34ABF             0048         JP      SETSEC
BE24 C34EBF             0049         JP      SETDMA
BE27 C353BF             0050         JP      READ
BE2A C357BF             0051         JP      WRITE
                        0052 ;
BE2D 217CBF             0053 BOOT:   LD      HL,SOMSG        ;Print sign on message
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0002
CP/M 1.4 BIOS for FDC-I

BE30  CD7ABE      0054          CALL    PMSG
BE33  21A7BF      0055 BT:      LD      HL,GNDR       ;Get # of drives
BE36  CD7ABE      0056          CALL    PMSG
BE39  CDC9BE      0057          CALL    CONIN
BE3C  CD0FE0      0058          CALL    OCH
BE3F  D631        0059          SUB     '1'
BE41  38F0        0060          JR      C,BT
BE43  FE04        0061          CP      4
BE45  30EC        0062          JR      NC,BT
BE47  324000      0063          LD      (NDRV),A      ;Save no. of drives
BE4A  97          0064          SUB     A
BE4B  320400      0065          LD      (CDRV),A      ;Set current drive to A:
BE4E  324200      0066          LD      (CHC),A       ;Reset char count
BE51  324300      0067          LD      (DEL),A       ;& DELete flag
BE54  3EC3        0068 SETUP:   LD      A,0C3H        ;Set up jump vectors
BE56  320000      0069          LD      (0),A
BE59  320500      0070          LD      (5),A
BE5C  2103BE      0071          LD      HL,JWBT
BE5F  220100      0072          LD      (1),HL
BE62  2106B1      0073          LD      HL,BDOS+6
BE65  220600      0074          LD      (6),HL
BE68  218000      0075          LD      HL,80H        ;Set init DMA addr
BE6B  224500      0076          LD      (DMA),HL
BE6E  3EE0        0077          LD      A,FDC/256     ;Save location of monitor
BE70  324000      0078          LD      (40H),A       ;for utility programs
BE73  3A0400      0079          LD      A,(CDRV)
BE76  4F          0080          LD      C,A
BE77  C300A9      0081          JP      CCP
                  0082 ;
BE7A  7E          0083 PMSG:    LD      A,(HL)        ;Print ASCII message
BE7B  23          0084          INC     HL
BE7C  CD0FE0      0085          CALL    OCH
BE7F  B7          0086          OR      A
BE80  F8          0087          RET     M             ;Exit if bit 7 set
BE81  18F7        0088          JR      PMSG
                  0089 ;
BE83  97          0090 WBOOT:   SUB     A             ;Boot from drive 0
BE84  CD24E0      0091          CALL    STDRV
BE87  318000      0092          LD      SP,80H        ;Set up stack
BE8A  CD2AE0      0093 WBT:     CALL    RECAL         ;Recalibrate drive
BE8D  2807        0094          JR      Z,WBT0
BE8F  CD18E0      0095          CALL    CONST
BE92  28F6        0096          JR      Z,WBT         ;Loop until no errors
BE94  1828        0097          JR      WBERR
BE96  CD2DE0      0098 WBT0:    CALL    SEEK          ;Set track 0
BE99  3E02        0099          LD      A,2
BE9B  CD27E0      0100          CALL    STSECT
BE9E  2100A9      0101          LD      HL,CCP
BEA1  1E19        0102          LD      E,25
BEA3  CD30E0      0103          CALL    READF         ;Read trk 0, sectors 2-24
BEA6  2016        0104          JR      NZ,WBERR
BEA8  3C          0105          INC     A
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0003
CP/M 1.4 BIOS for FDC-I

BEA9  CD27E0          0106          CALL    STSECT
BEAC  CD2DE0          0107          CALL    SEEK          ;Seek track 1
BEAF  1E11            0108          LD      E,17
BEB1  CD30E0          0109          CALL    READF         ;Read trk 1, sectors 1-17
BEB4  2008            0110          JR      NZ,WBERR
BEB6  3A0400          0111          LD      A,(CDRV)      ;Restore current drive
BEB9  CD24E0          0112          CALL    STDRV
BEBC  1896            0113          JR      SETUP
BEBE  21D5BF          0114 WBERR:   LD      HL,WBER       ;Warm boot error
BEC1  CD7ABE          0115          CALL    PMSG
BEC4  CDC9BE          0116          CALL    CONIN
BEC7  18C1            0117          JR      WBT
                      0118 ;
BEC9  CD09E0          0119 CONIN:   CALL    ICH           ;Console input
BECC  E67F            0120          AND     7FH           ;Reset bit 7
BECE  214200          0121          LD      HL,CHC        ;Point to char count
BED1  34              0122          INC     (HL)          ;Inc char count
BED2  FE08            0123          CP      8             ;Back space?
BED4  2002            0124          JR      NZ,CIN1       ;No
BED6  3E7F            0125          LD      A,7FH         ;Yes, convert it to DEL
BED8  FE7F            0126 CIN1:    CP      7FH           ;DELete?
BEDA  C0              0127          RET     NZ            ;No
BEDB  35              0128          DEC     (HL)          ;Yes, dec char count
BEDC  C8              0129          RET     Z             ;If zero, return
BEDD  23              0130          INC     HL            ;Point to DELete flag
BEDE  3608            0131          LD      (HL),8        ;& set it
BEE0  C9              0132          RET
                      0133 ;
BEE1  214300          0134 CONOUT:  LD      HL,DEL        ;Console output
BEE4  7E              0135          LD      A,(HL)
BEE5  B7              0136          OR      A             ;DEL flag set?
BEE6  280D            0137          JR      Z,COUT        ;No
BEE8  3600            0138          LD      (HL),0        ;Yes, reset it
BEEA  2B              0139          DEC     HL            ;Point to char count
BEEB  35              0140          DEC     (HL)          ;& dec char count
BEEC  4F              0141          LD      C,A
BEED  CD0FE0          0142          CALL    OCH           ;Send a BS, space & BS sequence
BEF0  3E20            0143          LD      A,' '
BEF2  CD0FE0          0144          CALL    OCH
BEF5  79              0145 COUT:    LD      A,C
BEF6  CD0FE0          0146          CALL    OCH           ;Output char
BEF9  FE0D            0147          CP      13            ;CR?
BEFB  C0              0148          RET     NZ            ;No
BEFC  2B              0149          DEC     HL
BEFD  3600            0150          LD      (HL),0        ;Yes, reset char count
BEFF  C9              0151          RET
                      0152 ;
BF00  79              0153 LIST:    LD      A,C           ;Listing device
BF01  C312E0          0154          JP      LOUT
                      0155 ;
      (BF00)          0156 PUNCH:   EQU     LIST          ;Punching device
                      0157 ;
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0004
CP/M 1.4 BIOS for FDC-I

BF04  3E1A        0158 READER: LD    A,1AH        ;Reading device
BF06  C9          0159         RET               ;Not implemented
                  0160 ;
BF07  CD2AE0      0161 HOME:   CALL  RECAL        ;Recalibrate
BF0A  C8          0162         RET   Z
BF0B  CD10BF      0163         CALL  SKHMER
BF0E  18F7        0164         JR    HOME
BF10  21BABF      0165 SKHMER: LD    HL,SHER      ;Seek or recal error
BF13  CD7ABE      0166         CALL  PMSG
BF16  3A0400      0167         LD    A,(CDRV)
BF19  C641        0168         ADD   'A'
BF1B  CD0FE0      0169         CALL  OCH          ;Print drive
BF1E  C3C9BE      0170         JP    CONIN
                  0171 ;
BF21  3A4000      0172 SELDSK: LD    A,(NDRV)     ;Select drive
BF24  B9          0173         CP    C
BF25  3807        0174         JR    C,SDERR      ;Jp if error
BF27  79          0175         LD    A,C
BF28  320400      0176         LD    (CDRV),A
BF2B  C324E0      0177         JP    STDRV
BF2E  21E7BF      0178 SDERR:  LD    HL,DERR      ;Drive selected is
BF31  CD7ABE      0179         CALL  PMSG         ;not on line
BF34  3E41        0180         LD    A,'A'
BF36  81          0181         ADD   C
BF37  CD0FE0      0182         CALL  OCH
BF3A  CD7ABE      0183         CALL  PMSG
BF3D  C383BE      0184         JP    WBOOT        ;Reboot CP/M
                  0185 ;
BF40  79          0186 SETTRK: LD    A,C          ;Set track
BF41  CD2DE0      0187         CALL  SEEK         ;Seek the track
BF44  C8          0188         RET   Z
BF45  CD10BF      0189         CALL  SKHMER
BF48  18F6        0190         JR    SETTRK
                  0191 ;
BF4A  79          0192 SETSEC: LD    A,C          ;Set sector
BF4B  C327E0      0193         JP    STSECT
                  0194 ;
BF4E  ED434500    0195 SETDMA: LD    (DMA),BC     ;Set DMA addr
BF52  C9          0196         RET
                  0197 ;
BF53  3EFF        0198 READ:   LD    A,-1         ;Read 1 sector
BF55  1801        0199         JR    RORW         ;Set read operation
BF57  97          0200 WRITE:  SUB   A            ;Write 1 sector
BF58  324100      0201 RORW:   LD    (RWOP),A     ;Set read/write flag
BF5B  3E0A        0202         LD    A,10
BF5D  324400      0203 RW:     LD    (RETRY),A
BF60  2A4500      0204         LD    HL,(DMA)
BF63  1E01        0205         LD    E,1
BF65  CD72BF      0206         CALL  RDORWT
BF68  C8          0207         RET   Z
BF69  3A4400      0208         LD    A,(RETRY)
BF6C  3D          0209         DEC   A
```

```
CROMEMCO CDOS Z80 ASSEMBLER version 02.15                    PAGE 0005
CP/M 1.4 BIOS for FDC-I

BF6D  20EE          0210          JR      NZ,RW
BF6F  3E01          0211          LD      A,1             ;Couldn't read/write sector
BF71  C9            0212          RET
BF72  3A4100        0213 RDORWT:  LD      A,(RWOP)        ;Check for read or write
BF75  B7            0214          OR      A
BF76  C230E0        0215          JP      NZ,READF        ;Read operation
BF79  C333E0        0216          JP      WRITF           ;Write operation
                    0217 ;
                    0218 ;BIOS messages to user:
BF7C  0C54656C      0219 SOMSG:   DB      12,'Teletek''','s FDC-I ',MEM/10+'0'
      6574656B
      27732046
      44432D49
      2034
BF8E  384B2043      0220          DB      MEM MOD 10 +'0','K CP/M 1.4 '
      502F4D20
      312E3420
BF9A  5B52656C      0221          DM      '[Release 1.3]'
      65617365
      20312E33
      DD
BFA7  0D0A486F      0222 GNDR:    DM      13,10,'How many drives? '
      77206D61
      6E792064
      72697665
      733FA0
BFBA  0D0A5365      0223 SHER:    DM      13,10,'Seek/Home error on drive '
      656B2F48
      6F6D6520
      6572726F
      72206F6E
      20647269
      7665A0
BFD5  0D0A5761      0224 WBER:    DM      13,10,'Warm boot error!'
      726D2062
      6F6F7420
      6572726F
      72A1
BFE7  0D0A4472      0225 DERR:    DM      13,10,'Drive '
      697665A0
BFEF  20697320      0226          DM      ' is not on line!'
      6E6F7420
      6F6E206C
      696E65A1
                    0227 ;
BFFF  (0000)        0228          END

Errors              0
```

```
                        ;*****************************************************
                        ;*       Copyright (C) Teletek Enterprises, Inc.     *
                        ;*          2K Monitor for the Teletek's FDC-I       *
                        ;*                Date 9-25-81                       *
                        ;*                                                   *
                        ;*                Written by Aram Perez              *
                        ;*****************************************************
                        ;
                        ;option definitions:
FFFF                    true    equ     -1
0000                    false   equ     not true
FFFF                    eight   equ     true            ;init for 8" drives,
                                                        ;make false for 5.25"
0000                    auto    equ     false           ;auto boot the dos,
                                                        ;make false for monitor
0000                    persci  equ     false           ;for persci drives
0000                    tandon  equ     false           ;for tandon 8" drives
                        ;
                        ;symbol definitions:
0028                    rls     equ     40              ;release number x10
0061                    vrsn    equ     'a'             ;version
0008                    bs      equ     8               ;back space
000A                    lf      equ     10              ;line feed
000D                    cr      equ     13              ;carriage return
0018                    can     equ     18h             ;^x, cancel
007F                    del     equ     7fh             ;delete (or rubout)
FFFF                    n$i     equ     0ffffh
FFFF                    usr     equ     n$i             ;user routine (n$i)*
                        ;*routines that are not implemented are marked "(n$i)".
                        ;
                                if      vrsn eq 'a'
E000                    mon     equ     0e000h          ;location of monitor
FE00                    sysram  equ     0fe00h          ;    "     " system ram
                                endif
                        ;
                                if      vrsn eq 'b'
                        mon     equ     0f000h          ;location of monitor
                        sysram  equ     0ee00h          ;    "     " system ram
                                endif
                        ;
                                if      vrsn eq 'c'
                        mon     equ     0f800h          ;location of monitor
                        sysram  equ     0f600h          ;    "     " system ram
                                endif
                        ;
FFFE                    iv      equ     sysram/256      ;interrupt vector ("i" reg)
FEE0                    fivt    equ     sysram+0e0h     ;start of fdc-i int table
FF40                    stack   equ     sysram+140h     ;location of system stack
                        ;
```

```
                        ;fdc-i i/o ports:
0000                    sad     equ     0               ;sio a data
0001                    sac     equ     1               ;sio a control/status
0002                    sbd     equ     2               ;sio b data
0003                    sbc     equ     3               ;sio b control/status
0004                    pad     equ     4               ;pio a data
0005                    pac     equ     5               ;pio a control/status
0006                    pbd     equ     6               ;pio b data
0008                    ctc0    equ     8               ;ctc, channel 0
000C                    fdcst   equ     12              ;fdc status
000D                    fdcdt   equ     13              ;fdc data
0010                    dma     equ     16              ;fdc dma acknowledge
0014                    tc      equ     20              ;fdc terminal count
0018                    wait    equ     24              ;fdc wait for data
                        ;
                                subttl Interfacing vectors and initialization
                                page
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80      PAGE    1-3
Interfacing vectors and initialization


                                        .phase  mon
                                ;
                                ;interfacing vectors:
E000    C3 E05D                         jp      init            ;initialize
E003    C3 E0DA                         jp      warm            ;monitor warm entry
E006    C3 E3BC                         jp      inpm            ;input routine, bit 7 reset
E009    C3 E3CA                         jp      input           ;input routine
E00C    C3 E3E5                         jp      sbin            ;input from sio b
E00F    C3 E413                         jp      outch           ;output routine
E012    C3 E42D                         jp      sbout           ;output to sio b
E015    C3 E443                         jp      pout            ;  "     " pio a
E018    C3 E3AB                         jp      gstat           ;get input device status
E01B    C3 E3B8                         jp      statin          ;get inp stat & char
E01E    C3 E3C2                         jp      statb           ;get input status sio b
E021    C3 E48E                         jp      stmode          ;set fdc mode
E024    C3 E4A4                         jp      sthdr           ; "   "  head/drive
E027    C3 E4B1                         jp      stsect          ; "   "  sector
E02A    C3 E4B5                         jp      recal           ;fdc, recalibrate drive
E02D    C3 E4CB                         jp      seek            ; " , seek a track
E030    C3 E4E6                         jp      read            ; " , read
E033    C3 E4F7                         jp      write           ; " , write
E036    C3 E39B                         jp      rtst0           ; " , return st0 address
E039    C3 E39F                         jp      rtsec           ;return sec address
E03C    C3 E37C                         jp      asio            ;assign i/o (iobyte)
E03F    C3 E380                         jp      aspa            ;assign pio a (in/out)
E042    C3 E397                         jp      srtdf           ;set/reset time display flag
E045    C3 E34B                         jp      stssp           ;set serial speed
E048    C3 FFFF                         jp      usr             ;optional user routine (n$i)
E04B    C3 E52E                         jp      readid          ;fdc, read id
E04E    C3 E543                         jp      drvst           ; " , return drive status
E051    C3 E3A3                         jp      rtdrdy          ; " , return drdy0 address
E054    C3 E478                         jp      spec            ; " , do specify command
E057    C3 E45A                         jp      stsize          ; " , set drive size
E05A    C3 E3A7                         jp      rtivt           ;return addr of int table
                                ;
                                ;initialization routine:
E05D    21 E6F4                 init:   ld      hl,piotb        ;point to init data
E060    16 07                           ld      d,7
E062    0E 05                           ld      c,pac           ;start with pio a
E064    46                      init1:  ld      b,(hl)          ;init pio and ctc
E065    23                              inc     hl
E066    ED B3                           otir
E068    0C                              inc     c
E069    15                              dec     d
E06A    20 F8                           jr      nz,init1
E06C    16 02                           ld      d,2
E06E    0E 01                           ld      c,1
E070    46                      init2:  ld      b,(hl)          ;init sio
E071    23                              inc     hl
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-4
Interfacing vectors and initialization

  E072    ED B3                         otir
  E074    0C                            inc     c
  E075    0C                            inc     c
  E076    15                            dec     d
  E077    20 F7                         jr      nz,init2
  E079    11 FEE0                       ld      de,fivt
  E07C    01 0021                       ld      bc,itl
  E07F    ED B0                         ldir                    ;set up interrupt table
  E081    31 FF40                       ld      sp,stack        ;set up the stack
  E084    3E FE                         ld      a,iv            ;set up interrupt mode 2
  E086    ED 47                         ld      i,a
  E088    ED 5E                         im      2
  E08A    11 FF9C                       ld      de,iobyte       ;set up the system ram
  E08D    01 000D                       ld      bc,srl
  E090    ED B0                         ldir
  E092    FB                            ei
  E093    DB 04                         in      a,(pad)         ;enable handshake lines
                                ;
                                        if      eight ;inch drives
  E095    3E FF                         ld      a,-1
                                        else ;five and a quarter inch drives
                                        ld      a,0
                                        endif
                                ;
  E097    CD E337                       call    sdrv            ;set drive size
  E09A    21 FFB0                       ld      hl,drdy0        ;check rdy on all 4 drives
  E09D    01 0400                       ld      bc,400h
  E0A0    79                    init3:  ld      a,c
  E0A1    CD E4A4                       call    sthdr           ;set drive
  E0A4    CD E543                       call    drvst           ;and get status
  E0A7    E6 20                         and     20h             ;ready?
  E0A9    77                            ld      (hl),a          ;save rdy status
  E0AA    23                            inc     hl
  E0AB    0C                            inc     c
  E0AC    10 F2                         djnz    init3
                                ;
                                        if      auto
                                        nop                     ;fall thru to boot
                                        nop
                                        else
  E0AE    18 24                         jr      wtio            ;skip boot
                                        endif
                                ;
                                ;boot is the floppy boot routine.  it reads in the 1st
                                ;sector on track 0 on drive 0 at location 0000, and
                                ;tranfers to 0000.  it will keep trying till succesful.
  E0B0    97                    boot:   sub     a
  E0B1    CD E48E                       call    stmode          ;set single density mode
  E0B4    CD E4A4                       call    sthdr           ;set drive
  E0B7    3C                            inc     a
  E0B8    32 FFBC                       ld      (sect),a        ;sector 1
```

Interfacing vectors and initialization

```
E0BB    CD E167         bt0:    call    cancel?
E0BE    3A FFB0                 ld      a,(drdy0)
E0C1    B7                      or      a
E0C2    28 F7                   jr      z,bt0
E0C4    CD E4B5                 call    recal           ;recalibrate drive
E0C7    21 0000                 ld      hl,0
E0CA    1E 01                   ld      e,1
E0CC    CD E4E6                 call    read            ;read 1st sector
E0CF    20 EA                   jr      nz,bt0          ;loop till no errors
E0D1    CD 0000                 call    0
                        ;
                        ;wtio waits until i/o is assigned.
E0D4    3A FF9C         wtio:   ld      a,(iobyte)
E0D7    B7                      or      a
E0D8    28 FA                   jr      z,wtio
                                subttl Monitor routines
                                page
```

```
                        ;
                        ;warm is the warm entry point to the monitor.
E0DA    21 E783 warm:   ld      hl,som          ;print sign-on message
E0DD    CD E1B7         call    pstr
E0E0    31 FF40 ldsp:   ld      sp,stack        ;load stack pointer
E0E3    CD E3B8         call    statin          ;get rid of any char in buf
                        ;
                        ;incmd inputs the desired command.
E0E6    21 E797 incmd:  ld      hl,prmt
E0E9    CD E121         call    instr           ;prompt & get command
E0EC    21 FFC9         ld      hl,cbuff
E0EF    7E              ld      a,(hl)          ;get command
E0F0    E6 5F           and     5fh             ;convert to u.c.
E0F2    47              ld      b,a
E0F3    23              inc     hl
E0F4    7E              ld      a,(hl)          ;get 2nd char
E0F5    FE 20           cp      ' '
E0F7    28 03           jr      z,cmd1
E0F9    E6 5F           and     5fh
E0FB    23              inc     hl
E0FC    36 BF   cmd1:   ld      (hl),'?'+80h    ;in case of error
E0FE    23              inc     hl
E0FF    22 FFA3         ld      (cptr),hl       ;save for conv
E102    80              add     a,b
E103    80              add     a,b
E104    06 12           ld      b,nocmd         ;no. of cmd's
E106    21 E7B1         ld      hl,cmdtb        ;get command table
E109    BE      cmd2:   cp      (hl)            ;find command?
E10A    23              inc     hl
E10B    28 0C           jr      z,cmd3          ;yes
E10D    23              inc     hl
E10E    23              inc     hl
E10F    10 F8           djnz    cmd2
E111    21 FFC9         ld      hl,cbuff
E114    CD E1B7         call    pstr
E117    18 CD           jr      incmd
E119    5E      cmd3:   ld      e,(hl)          ;get routine addr
E11A    23              inc     hl
E11B    56              ld      d,(hl)
E11C    CD E261         call    xcmd            ;execute routine
E11F    18 C5           jr      incmd
                        ;
                        ;instr prints a message and then inputs a data string
                        ;and stores it at cbuff.  it exits on cr or when cbl
                        ;characters have been input.
E121    CD E1B7 instr:  call pstr
E124    06 37           ld      b,cbl
E126    21 FFC9         ld      hl,cbuff
E129    22 FFA3         ld      (cptr),hl
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80      PAGE   1-7
Monitor routines

 E12C    C5                     push    bc
 E12D    E5                     push    hl
 E12E    36 20          inst1:  ld      (hl),' '        ;clear buffer
 E130    23                     inc     hl
 E131    10 FB                  djnz    inst1
 E133    E1                     pop     hl
 E134    C1                     pop     bc
 E135    CD E3BC        inst2:  call    inpm            ;get a char
 E138    FE 18                  cp      can             ;cancel?
 E13A    28 36                  jr      z,cncl
 E13C    FE 0D                  cp      cr
 E13E    28 24                  jr      z,inst6         ;exit on cr
 E140    FE 08                  cp      bs              ;back space?
 E142    20 0E                  jr      nz,inst3
 E144    78                     ld      a,b
 E145    FE 37                  cp      cbl
 E147    28 EC                  jr      z,inst2
 E149    04                     inc     b
 E14A    2B                     dec     hl
 E14B    3E 08                  ld      a,bs
 E14D    CD E413                call    outch
 E150    18 E3                  jr      inst2
 E152    FE 20          inst3:  cp      ' '             ;space?
 E154    20 01                  jr      nz,inst4
 E156    7E                     ld      a,(hl)
 E157    FE 7F          inst4:  cp      del             ;delete?
 E159    20 02                  jr      nz,inst5
 E15B    3E 20                  ld      a,' '
 E15D    77             inst5:  ld      (hl),a
 E15E    23                     inc     hl
 E15F    CD E413                call    outch
 E162    10 D1                  djnz    inst2
 E164    C3 E3ED        inst6:  jp      crlf
                        ;
                        ;cancel? checks if a ^x was the last char entered.  if it
                        ;is it prints "^X" and jumps to the command input routine.
 E167    E5             cancel?: push   hl
 E168    2A FF9F                 ld      hl,(queout)
 E16B    7E                      ld      a,(hl)
 E16C    E1                      pop     hl
 E16D    E6 7F                   and     7fh
 E16F    FE 18                   cp      can             ;cancel?
 E171    C0                      ret     nz              ;no
 E172    21 E79B        cncl:    ld      hl,cmsg
 E175    CD E1B7                 call    pstr
 E178    C3 E0E0                 jp      ldsp
                        ;
                        ;ashex converts a.reg from ascii to hex.  on exit, if
                        ;cy= 0 then a.reg was an invalid hex digit.
 E17B    FE 61          ashex:   cp      'a'             ;lower case?
 E17D    38 02                   jr      c,ashx0         ;no
```

```
FDC-I Monitor, Release 4.0   MACRO-80 3.4   01-Dec-80   PAGE   1-8
Monitor routines

E17F   E6 5F                        and     5fh             ;yes, convert to u.c.
E181   D6 30                ashx0:  sub     '0'             ;get rid of ascii bias
E183   38 0E                        jr      c,ashx1         ;check for valid hex digit
E185   FE 17                        cp      17h
E187   D0                           ret     nc
E188   FE 0A                        cp      10
E18A   D8                           ret     c
E18B   FE 11                        cp      11h
E18D   38 04                        jr      c,ashx1
E18F   D6 07                        sub     7
E191   37                           scf
E192   C9                           ret
E193   B7                  ashx1:  or      a               ;clear cy
E194   C9                           ret
                            ;
                            ;conv will load de with the ascii hex number pointed to
                            ;by (cptr).
E195   E5                  conv:   push    hl
E196   21 0000                     ld      hl,0
E199   ED 5B FFA3                  ld      de,(cptr)       ;get pointer
E19D   1A                  conv1:  ld      a,(de)
E19E   13                           inc     de
E19F   CD E17B                      call    ashex
E1A2   30 08                        jr      nc,conv2        ;exit on invalid hex #
E1A4   29                           add     hl,hl
E1A5   29                           add     hl,hl
E1A6   29                           add     hl,hl
E1A7   29                           add     hl,hl
E1A8   85                           add     a,l
E1A9   6F                           ld      l,a
E1AA   18 F1                        jr      conv1
E1AC   EB                  conv2:  ex      de,hl           ;put # in de
E1AD   22 FFA3                      ld      (cptr),hl       ;save pointer
E1B0   E1                           pop     hl
E1B1   C9                           ret
                            ;
                            ;conve calls conv and then loads a with e.
E1B2   CD E195              conve:  call    conv
E1B5   7B                           ld      a,e
E1B6   C9                           ret
                            ;
                            ;pstr prints an ascii string.
E1B7   7E                   pstr:   ld      a,(hl)
E1B8   23                           inc     hl
E1B9   CD E413                      call    outch
E1BC   B7                           or      a
E1BD   F8                           ret     m               ;exit if bit 7 set
E1BE   18 F7                        jr      pstr
                            ;
                            ;da will dump memory in ascii.
E1C0   CD E195              da:     call    conv            ;get address
```

```
E1C3  EB
E1C4  CD E409
E1C7  E5
E1C8  06 08
E1CA  7E
E1CB  CD E40E
E1CE  23
E1CF  10 F9
E1D1  E1
E1D2  06 08
E1D4  7E
E1D5  E6 7F
E1D7  FE 20
E1D9  38 04
E1DB  FE 7F
E1DD  38 02
E1DF  3E 2E
E1E1  CD E413
E1E4  23
E1E5  10 ED
E1E7  CD E3BC
E1EA  FE 20
E1EC  C0
E1ED  CD E3ED
E1F0  18 D2

E1F2  CD E1B2
E1F5  32 FFB9
E1F8  CD E543
E1FB  C3 E3F6

E1FE  CD E195
E201  CD E23B
E204  1A
E205  CD E40E
E208  CD E3BC
E20B  FE 20
E20D  28 1F
E20F  FE 08
E211  28 25
E213  CD E413
E216  CD E17B
E219  D0
E21A  07
E21B  07
E21C  07
E21D  07
```

```
        ex    de,hl
da0:    call  rthl
        push  hl
        ld    b,8
da1:    ld    a,(hl)
        call  rtas
        inc   hl
        djnz  da1
        pop   hl
        ld    b,8
da2:    ld    a,(hl)          ;get contents
        and   7fh             ;reset bit 7
        cp    ' '             ;is it printable?
        jr    c,skip          ;no
        cp    del
        jr    c,print         ;yes
skip:   ld    a,'.'           ;print a period instead
print:  call  outch
        inc   hl
        djnz  da2
        call  inpm
        cp    ' '
        ret   nz
        call  crlf
        jr    da0
;
;drst will print the status of a given drive.
drst:   call  conve           ;get drive
        ld    (hdr),a
        call  drvst
        jp    rthex           ;print status
;
;em will enter new contents into memory.  it will
;display the contents of the memory location
;before and after the new contents are entered.
em:     call  conv            ;get addr
em1:    call  rtde
        ld    a,(de)          ;get contents &
        call  rtas            ;display them
        call  inpm            ;get new contents
        cp    ' '
        jr    z,em4           ;skip on space
        cp    bs
        jr    z,em6           ;go back on bs
        call  outch
        call  ashex
        ret   nc
        rlca
        rlca
        rlca
        rlca
```

```
E21E    47
E21F    CD E3BC
E222    CD E413
E225    CD E17B
E228    D0
E229    80
E22A    12
E22B    CD E411
E22E    1A
E22F    CD E3F6
E232    13
E233    CD E3ED
E236    18 C9
E238    1B
E239    18 F8

E23B    EB
E23C    CD E409
E23F    EB
E240    C9


E241    CD E24F
E244    CD E195
E247    73
E248    23
E249    0B
E24A    78
E24B    B1
E24C    20 F9
E24E    C9


E24F    CD E195
E252    EB
E253    CD E195
E256    13
E257    7B
E258    95
E259    4F
E25A    7A
E25B    9C
E25C    47
E25D    C9


E25E    CD E195
E261    EB
E262    E9
```

```
        ld      b,a
        call    inpm
        call    outch
        call    ashex
        ret     nc
        add     a,b
        ld      (de),a          ;store new value
        call    space
em4:    ld      a,(de)          ;get contents
        call    rthex
        inc     de
em5:    call    crlf
        jr      em1
em6:    dec     de
        jr      em5
;
;rtde prints de in hex followed by a space.
rtde:   ex      de,hl
        call    rthl
        ex      de,hl
        ret
;
;fill will fill a memory block with a given byte.
fill:   call    bklen
        call    conv            ;get fill byte
fil:    ld      (hl),e
        inc     hl
        dec     bc
        ld      a,b
        or      c
        jr      nz,fil
        ret
;
;bklen calculates the length of a memory block.
bklen:  call    conv
        ex      de,hl           ;hl= beg of block
        call    conv
        inc     de              ;de= end of block+1
        ld      a,e             ;calculate length
        sub     l
        ld      c,a
        ld      a,d
        sbc     a,h
        ld      b,a             ;bc= length
        ret
;
;go will transfer to a given address.
go:     call    conv            ;get addr
xcmd:   ex      de,hl
        jp      (hl)
;
```

```
                        ;inp will input and print a byte from a given port.
E263  CD E195           inp:    call    conv
E266  4B                        ld      c,e
E267  ED 78             in:     in      a,(c)
E269  CD E40E                   call    rtas
E26C  CD E3BC                   call    inpm
E26F  FE 20                     cp      ' '
E271  28 F4                     jr      z,in
E273  C9                        ret
                        ;
                        ;move will move a block of memory to another location.
E274  CD E24F           move:   call    bklen           ;get memory block
E277  CD E195                   call    conv            ;get beg addr
E27A  ED B0                     ldir
E27C  21 FFCB                   ld      hl,cbuff+2
E27F  22 FFA3                   ld      (cptr),hl
                        ;
                        ;verfy will verify a block of memory with another block.
                        ;if any error occur they will be printed.  after 16 errors
                        ;are printed verfy will wait for any key to be type before
                        ;printing the next 16 errors.
E282  CD E24F           verfy:  call    bklen           ;verify memory
E285  CD E195                   call    conv
E288  1B                verf:   dec     de
E289  3E 10             ver0:   ld      a,16
E28B  32 FFA5           ver1:   ld      (vcnt),a        ;set verify count
E28E  CD E167           ver2:   call    cancel?         ;quit?
E291  13                        inc     de
E292  1A                        ld      a,(de)
E293  ED A1                     cpi
E295  28 15                     jr      z,ver3          ;jp if no error
E297  F5                        push    af
E298  2B                        dec     hl
E299  CD E409                   call    rthl
E29C  7E                        ld      a,(hl)
E29D  23                        inc     hl
E29E  CD E40E                   call    rtas
E2A1  CD E23B                   call    rtde
E2A4  1A                        ld      a,(de)
E2A5  CD E3F6                   call    rthex
E2A8  CD E3ED                   call    crlf
E2AB  F1                        pop     af
E2AC  E0                ver3:   ret     po
E2AD  28 DF                     jr      z,ver2
E2AF  3A FFA5                   ld      a,(vcnt)
E2B2  3D                        dec     a               ;only print 16
E2B3  20 D6                     jr      nz,ver1         ;errors at a time,
E2B5  CD E3CA                   call    input           ;then wait for a key
E2B8  CD E3ED                   call    crlf
E2BB  18 CC                     jr      ver0
                        ;
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-12
Monitor routines

                                ;out will output a byte to a given port.
E2BD    CD E24F         out:    call    bklen           ;l= port, e= byte+1
E2C0    4D                      ld      c,l
E2C1    1D                      dec     e
E2C2    ED 59           ot:     out     (c),e
E2C4    CD E3BC                 call    inpm
E2C7    FE 20                   cp      ' '
E2C9    28 F7                   jr      z,ot
E2CB    C9                      ret
                                ;
                                ;prog will program a 2716.
E2CC    CD E24F         prog:   call    bklen           ;get memory block
E2CF    CD E195                 call    conv            ;get addr of 2716
E2D2    3E C8                   ld      a,200
E2D4    F5              prog1:  push    af
E2D5    C5                      push    bc
E2D6    D5                      push    de
E2D7    E5                      push    hl
E2D8    ED B0                   ldir
E2DA    CD E167                 call    cancel?
E2DD    E1                      pop     hl
E2DE    D1                      pop     de
E2DF    C1                      pop     bc
E2E0    F1                      pop     af
E2E1    3D                      dec     a
E2E2    20 F0                   jr      nz,prog1
E2E4    18 A2                   jr      verf            ;verify data
                                ;
                                ;rec is the monitor command routine to recalibrate a drive.
E2E6    CD E1B2         rec:    call    conve           ;get drive
E2E9    32 FFB9                 ld      (hdr),a
E2EC    CD E4B5                 call    recal
E2EF    C8                      ret     z               ;return if no error
E2F0    3A FFC2                 ld      a,(st0)
E2F3    C3 E3F6                 jp      rthex           ;print error
                                ;
                                ;rdatd will read 1-26 sectors in mfm format
                                ;from a given track and drive.
E2F6    3E 40           rdatd:  ld      a,40h
E2F8    18 01                   jr      rdat
                                ;
                                ;rdats will read 1-26 sectors in fm format
                                ;from a given track and drive.
E2FA    97              rdats:  sub     a
E2FB    CD E31B         rdat:   call    gtdat
E2FE    CD E1B7                 call    pstr
E301    CD E121                 call    instr           ;get address and
E304    CD E24F                 call    bklen           ;number of sectors
E307    1D                      dec     e               ;e= # of sectors
E308    CD E4E6                 call    read
E30B    CD E409         rtres:  call    rthl            ;print results
```

```
E30E  06 07              ld    b,7
E310  21 FFC2            ld    hl,st0
E313  7E           rtr:  ld    a,(hl)
E314  CD E40E            call  rtas
E317  23                 inc   hl
E318  10 F9              djnz  rtr
E31A  C9                 ret
                   ;
E31B  CD E48E      gtdat: call  stmode          ;set fdc mode
E31E  CD E1B2            call  conve           ;get head/drive
E321  CD E4A4            call  sthdr
E324  CD E1B2            call  conve           ;get track
E327  CD E4CB            call  seek
E32A  CD E1B2            call  conve           ;get sector
E32D  32 FFBC            ld    (sect),a
E330  21 E79E            ld    hl,addr
E333  C9                 ret
                   ;
                   ;sd is the command to set the drive size.
E334  CD E1B2      sd:   call  conve
                   ;sdrv sets the drive size, either 5.25" or 8", along with t
                   ;765 timing constants.
E337  CD E45A      sdrv: call  stsize
                         ;if size is 8", then srt=10 ms, hut=240 ms, hlt=36
E33A  11 6F22            ld    de,6f22h
E33D  C2 E478            jp    nz,spec         ;do specify
                         ;size is 5.25", so srt=32 ms, hut=480 ms, hlt=42 ms
E340  11 0F28            ld    de,0f28h
E343  C3 E478            jp    spec            ;do specify
                   ;
                   ;stsp sets the baud rate on the serial ports.
E346  CD E24F      stsp: call  bklen           ;get port & speed
E349  1B                 dec   de
E34A  4D                 ld    c,l
E34B  0D           stssp: dec    c             ;enter with c= serial port
E34C  0D                 dec   c               ;and de the # of the speed
E34D  06 02              ld    b,2
E34F  21 E750            ld    hl,baudtb
E352  19                 add   hl,de
E353  19                 add   hl,de           ;point to baud rate
E354  ED B3              otir
E356  C9                 ret
                   ;
                   ;wdatd will write 1-26 sectors in mfm format.
E357  3E 40        wdatd: ld    a,40h
E359  18 01              jr    wdat
                   ;wdats will write 1-26 sectors in fm format.
E35B  97           wdats: sub   a
E35C  CD E31B      wdat:  call  gtdat
E35F  CD E121            call  instr           ;get addresses
E362  CD E24F            call  bklen
```

```
E365    E5                      push    hl
E366    60                      ld      h,b
E367    69                      ld      l,c
E368    3A FFBD                 ld      a,(n)           ;calculate # of sectors
E36B    B7                      or      a
E36C    20 01                   jr      nz,wdt1
E36E    29                      add     hl,hl
E36F    5C              wdt1:   ld      e,h
E370    7D                      ld      a,l
E371    B7                      or      a
E372    28 01                   jr      z,wdt2
E374    1C                      inc     e               ;e= # of sectors
E375    E1              wdt2:   pop     hl
E376    CD E4F7                 call    write
E379    C3 E30B                 jp      rtres
                        ;
                        ;
                        ;asio assigns the input and output device(s).
                        ;the iobyte is defined as follows:
                        ;bit    device          function
                        ; 7     reserved        input
                        ; 6     reserved          "   (changed from rel 3.2)
                        ; 5     sio a             "
                        ; 4     pio a             "
                        ; 3     user routine    output
                        ; 2     sio b             "
                        ; 1     sio a             "
                        ; 0     pio a             "
                        ;whenever a bit is set that device function is enabled.
E37C    32 FF9C         asio:   ld      (iobyte),a
E37F    C9                      ret
                        ;
                        ;aspa assigns the function of pio a.  changes register a.
E380    01 0305         aspa:   ld      bc,305h
E383    B7                      or      a               ;pio a as output?
E384    20 09                   jr      nz,asp0         ;no
E386    21 E760                 ld      hl,podt         ;yes, set up pio a as output
E389    ED B3                   otir
E38B    3D                      dec     a
E38C    D3 04                   out     (pad),a         ;set bit 7 (strobe bit)
E38E    C9                      ret
E38F    21 E6F5         asp0:   ld      hl,pidt         ;set up pio a as input
E392    ED B3                   otir
E394    DB 04                   in      a,(pad)         ;enable handshake lines
E396    C9                      ret
                        ;
                        ;srtdf sets/resets the time display flag.
E397    32 FFA6         srtdf:  ld      (tdf),a
E39A    C9                      ret
                        ;
                        ;rtst0 returns in hl the address of st0.
```

```
E39B    21 FFC2             rtst0:  ld      hl,st0
E39E    C9                          ret
                            ;
                            ;rtsec returns in hl the address of sec.
E39F    21 FFA9             rtsec:  ld      hl,sec
E3A2    C9                          ret
                            ;
                            ;rtdrdy returns in hl the address of drdy0.
E3A3    21 FFB0             rtdrdy: ld      hl,drdy0
E3A6    C9                          ret
                            ;
                            ;rtivt returns in hl the address of the interrupt
                            ;vector table.
E3A7    21 FE00             rtivt:  ld      hl,sysram
E3AA    C9                          ret
                                    subttl I/O & floppy disk routines
                                    page
```

```
                                ;
                                ;gstat gets the input device status.  it returns with z set
                                ;and a=0 if no data, otherwise z is reset and a=0ffh
                                ;if data is available.
E3AB    E5                      gstat:  push    hl
E3AC    2A FF9F                         ld      hl,(queout)
E3AF    3A FF9D                         ld      a,(quein)
E3B2    95                              sub     l
E3B3    E1                              pop     hl
E3B4    C8                              ret     z
E3B5    3E FF                           ld      a,-1
E3B7    C9                              ret
                                ;
                                ;statin calls gstat and returns if no data available,
                                ;otherwise it falls thru to inpm.
E3B8    CD E3AB                 statin: call    gstat
E3BB    C8                              ret     z
                                ;
                                ;inpm will input data with bit 7 reset.
E3BC    CD E3CA                 inpm:   call    input
E3BF    E6 7F                           and     7fh             ;reset bit 7
E3C1    C9                              ret
                                ;
                                ;statb gets the input status of sio b.  it returns with z
                                ;set and a=0 if no data otherwise z is reset and a=0ffh
                                ;if data is available.
E3C2    DB 03                   statb:  in      a,(sbc)
E3C4    E6 01                           and     1               ;data available?
E3C6    C8                              ret     z               ;no
E3C7    3E FF                           ld      a,-1            ;yes
E3C9    C9                              ret
                                ;
                                ;input will input data from the assigned input device.
E3CA    CD E3AB                 input:  call    gstat
E3CD    28 FB                           jr      z,input         ;loop till data available
E3CF    E5                              push    hl
E3D0    F3                              di
E3D1    2A FF9F                         ld      hl,(queout)
E3D4    7E                              ld      a,(hl)
E3D5    F5                              push    af
E3D6    2C                              inc     l
E3D7    3E 80                           ld      a,low quend
E3D9    BD                              cp      l
E3DA    20 02                           jr      nz,inpt
E3DC    2E 40                           ld      l,low queue
E3DE    22 FF9F                 inpt:   ld      (queout),hl
E3E1    FB                              ei
E3E2    F1                              pop     af
E3E3    E1                              pop     hl
```

```
E3E4  C9                  ret
                          ;
                          ;sbin inputs data from the second serial port (sio b).
E3E5  DB 03               sbin:   in      a,(sbc)
E3E7  0F                          rrca
E3E8  30 FB                       jr      nc,sbin         ;loop till data available
E3EA  DB 02                       in      a,(sbd)
E3EC  C9                          ret
                          ;
                          ;crlf prints a cr, lf sequence.
E3ED  3E 0D               crlf:   ld      a,cr
E3EF  CD E413                     call    outch
E3F2  3E 0A                       ld      a,lf
E3F4  18 1D                       jr      outch
                          ;
                          ;rthex prints a in hex.
E3F6  F5                  rthex:  push    af
E3F7  1F                          rra
E3F8  1F                          rra
E3F9  1F                          rra
E3FA  1F                          rra
E3FB  CD E3FF                     call    rthx
E3FE  F1                          pop     af
E3FF  E6 0F               rthx:   and     0fh
E401  C6 90                       add     a,90h
E403  27                          daa
E404  CE 40                       adc     a,40h
E406  27                          daa
E407  18 0A                       jr      outch
                          ;
                          ;rthl prints hl in hex followed by a space.
E409  7C                  rthl:   ld      a,h
E40A  CD E3F6                     call    rthex
E40D  7D                          ld      a,l
                          ;
                          ;rtas prints a in hex followed by a space.
E40E  CD E3F6             rtas:   call    rthex
                          ;
                          ;space prints a space.
E411  3E 20               space:  ld      a,' '
                          ;
                          ;outch will output data to the current output device.
E413  E5                  outch:  push    hl
E414  21 FF9C                     ld      hl,iobyte
E417  CB 5E                       bit     3,(hl)
E419  C4 E6EC                     call    nz,usout
E41C  CB 56                       bit     2,(hl)
E41E  C4 E42D                     call    nz,sbout
E421  CB 4E                       bit     1,(hl)
E423  C4 E438                     call    nz,saout
E426  CB 46                       bit     0,(hl)
```

```
E428  C4 E443                     call    nz,pout
E42B  E1                          pop     hl
E42C  C9                          ret
                                  ;
                                  ;sbout is the output routine for sio b.
E42D  F5                  sbout:  push    af
E42E  DB 03               sb:     in      a,(sbc)
E430  E6 04                       and     4
E432  28 FA                       jr      z,sb            ;loop till ready
E434  F1                          pop     af
E435  D3 02                       out     (sbd),a
E437  C9                          ret
                                  ;
                                  ;saout is the output routine for sio a.
E438  F5                  saout:  push    af
E439  DB 01               sa:     in      a,(sac)
E43B  E6 04                       and     4
E43D  28 FA                       jr      z,sa            ;loop till ready
E43F  F1                          pop     af
E440  D3 00                       out     (sad),a
E442  C9                          ret
                                  ;
                                  ;pout is the output routine for pio a.
                                  ;it is set up to use bit 7 as a negative strobe to
                                  ;connect to a "Centronics" printer.
E443  F5                  pout:   push    af
E444  3A FFA2             pout0:  ld      a,(outst)
E447  B7                          or      a               ;wait till port ready
E448  28 FA                       jr      z,pout0
E44A  3C                          inc     a
E44B  32 FFA2                     ld      (outst),a       ;set status
E44E  F1                          pop     af
E44F  F5                          push    af
E450  E6 7F                       and     7fh             ;strobe bit 7
E452  D3 04                       out     (pad),a
E454  F6 80                       or      80h
E456  D3 04                       out     (pad),a
E458  F1                          pop     af
E459  C9                          ret
                                  ;
                                  ;stsize sets the size for the 765.  it sets up the
                                  ;data pointer and changes the clock speed.  it uses
                                  ;the register pair hl.  it assumes that bit 0 of
                                  ;pio b is jumpered to change the clock speed to the 765.
E45A  21 E76D             stsize: ld      hl,fm8
E45D  B7                          or      a               ;check size
E45E  20 03                       jr      nz,sts0         ;jp if 8"
E460  21 E763                     ld      hl,fm5          ;point to 5.25" data
E463  22 FFB4             sts0:   ld      (rwdt),hl       ;set r/w data pointer
E466  32 FFB6                     ld      (drvsz),a       ;and size
E469  F5                          push    af
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80      PAGE    1-19
I/O & floppy disk routines

E46A    DB 06                           in      a,(pbd)
E46C    28 04                           jr      z,sts1
E46E    CB 87                           res     0,a             ;reset bit 0 for 8"
E470    18 02                           jr      sts2
E472    CB C7                   sts1:   set     0,a             ;set bit 0 for 5.25"
E474    D3 06                   sts2:   out     (pbd),a
E476    F1                              pop     af
E477    C9                              ret
                                ;
                                ;spec does a specify command.  it expects srt/hut in d and
                                ;hut in e(bits 7-1).  dma mode is always chosen.
E478    3E 03                   spec:   ld      a,3             ;do a fdc specify
E47A    CD E484                         call    fddrw
E47D    7A                              ld      a,d
E47E    CD E484                         call    fddrw
E481    7B                              ld      a,e
E482    E6 FE                           and     0feh            ;set dma mode
                                ;
                                ;fddrw outputs a to the fdc data reg.
E484    F5                      fddrw:  push    af
E485    DB 0C                   fddr:   in      a,(fdcst)
E487    07                              rlca                    ;wait for rqm
E488    30 FB                           jr      nc,fddr
E48A    F1                              pop     af
E48B    D3 0D                           out     (fdcdt),a
E48D    C9                              ret
                                ;
                                ;stmode sets the fdc mode for the read/write operations.
                                ;a should have the mode as follows:
                                ;bit 7 set for multi-track
                                ; "  6  "   "  mfm operation (double density)
                                ; "  5  "   "  skip deleted data address mark
                                ;bits 4-0 don't care.
                                ;it uses register pairs bc, de, and hl.
E48E    E6 E0                   stmode: and     0e0h            ;clear bits 4-0
E490    32 FFB8                         ld      (mode),a
E493    2A FFB4                         ld      hl,(rwdt)
E496    11 FFBD                         ld      de,n
E499    01 0005                         ld      bc,5
E49C    CB 77                           bit     6,a             ;mfm set?
E49E    28 01                           jr      z,stmu          ;no
E4A0    09                              add     hl,bc           ;yes
E4A1    ED B0                   stmu:   ldir                    ;set up n,eot,gpl & dtl
E4A3    C9                              ret
                                ;
                                ;sthdr sets the head and drive for the disk operations.
E4A4    32 FFB9                 sthdr:  ld      (hdr),a
E4A7    E6 04                           and     4               ;head 1?
E4A9    28 02                           jr      z,shd           ;no
E4AB    3E 01                           ld      a,1
E4AD    32 FFBB                 shd:    ld      (hd),a
```

```
E4B0    C9                      ret
                        ;
                        ;stsect sets the sector for the read/write operations.
E4B1    32 FFBC         stsect: ld      (sect),a
E4B4    C9                      ret
                        ;
                        ;recal is the routine that recalibrates a drive (moves the
                        ;head to track 0).  the zero flag is set and a=0 if no
                        ;errors occurred, otherwise the zero flag is reset and
                        ;a is non-zero.
E4B5    CD E559         recal:  call    motor
E4B8    97                      sub     a
E4B9    32 FFBA                 ld      (trk),a         ;set track to zero
E4BC    3E 07                   ld      a,7
E4BE    CD E484                 call    fddrw
E4C1    3A FFB9                 ld      a,(hdr)         ;get drive
                        ;
                                if      persci
                                and     0feh            ;reset bit 0
                                else
E4C4    E6 FF                   and     0ffh
                                endif
                        ;
E4C6    CD E484                 call    fddrw
E4C9    18 3D                   jr      fdcdn
                        ;
                        ;seek sets the track for disk operations and also seeks the
                        ;track.  the zero flag is set and a=0 if no errors occurred,
                        ;otherwise the zero flag is reset and a is non-zero.
E4CB    32 FFBA         seek:   ld      (trk),a
E4CE    CD E559                 call    motor
E4D1    3E 0F                   ld      a,15
E4D3    CD E484                 call    fddrw
E4D6    3A FFB9                 ld      a,(hdr)         ;get drive
                        ;
                                if      persci
                                and     0feh            ;reset bit 0
                                else
E4D9    E6 FF                   and     0ffh
                                endif
                        ;
E4DB    CD E484                 call    fddrw
E4DE    3A FFBA                 ld      a,(trk)
E4E1    CD E484                 call    fddrw
E4E4    18 22                   jr      fdcdn
                        ;
                        ;read will read from the floppy disk.  it assumes that the
                        ;mode, head/drive, track, and sector have already been set.
                        ;enter with hl= address to read data into and e= the numb
                        ;of sectors to read.  it ses all the registers.
                        ;the zero flag is set and a=0 if no errors occurred,
```

```
                        ;otherwise the zero flag is reset and a is non-zero.
E4E6    3E 06   read:   ld      a,6
E4E8    CD E514         call    stfdc           ;set up fdc chip
E4EB    42      rdfd:   ld      b,d
E4EC    DB 18   rdf:    in      a,(wait)
E4EE    ED A2           ini                     ;read data from fdc
E4F0    20 FA           jr      nz,rdf
E4F2    1D              dec     e
E4F3    20 F6           jr      nz,rdfd
E4F5    18 0F           jr      dotc
                        ;
                        ;write will write to the floppy disk. it assumes that the
                        ;mode, head/drive, track, and sector have already been set.
                        ;enter with hl= address to write data from and e= the number
                        ;of sectors to write.  it uses all the registers.
                        ;the zero flag is set and a=0 if no errors occurred,
                        ;otherwise the zero flag is reset and a is non-zero.
E4F7    3E 05   write:  ld      a,5
E4F9    CD E514         call    stfdc           ;set up fdc chip
E4FC    42      wtfd:   ld      b,d
E4FD    DB 18   wtf:    in      a,(wait)
E4FF    ED A3           outi                    ;write data to fdc
E501    20 FA           jr      nz,wtf
E503    1D              dec     e
E504    20 F6           jr      nz,wtfd
E506    DB 14   dotc:   in      a,(tc)          ;do a terminal count
                        ;
                        ;fdcdn checks the error code after a fdc operation.  if a=0
                        ;no errors occurred.  this routine is executed after every
                        ;fdc interrupt unless the interrupt cause was a change of
                        ;state in the ready line.  see the fdcint routine for more
                        ;details.
E508    3A FFB7 fdcdn:  ld      a,(fdone)
E50B    B7              or      a
E50C    28 FA           jr      z,fdcdn         ;jp if not done
E50E    3A FFC2         ld      a,(st0)
E511    E6 C0           and     0c0h            ;normal termination?
E513    C9              ret
                        ;
                        ;stfdc sets up the upd765 chip for a read or write
                        ;operation.  enter with a equal 5 for write or equal 6 for
                        ;read.  on exit d= sector size, b= 0 and c= fdc dma port.
E514    E5      stfdc:  push    hl
E515    21 FFB8         ld      hl,mode
E518    B6              or      (hl)
E519    CD E484         call    fddrw
E51C    CD E559         call    motor
E51F    06 08           ld      b,8
E521    23      stf1:   inc     hl              ;output rest of data
E522    7E              ld      a,(hl)
E523    CD E484         call    fddrw
```

```
E526  10 F9                djnz    stf1
E528  0E 10                ld      c,dma           ;dma port
E52A  23                   inc     hl
E52B  56                   ld      d,(hl)          ;d= sector size
E52C  E1                   pop     hl
E52D  C9                   ret
                      ;
                      ;readid reads the first correct id information on a track.
                      ;the zero flag is set and a=0 if no errors occurred,
                      ;otherwise the zero flag is reset and a is non-zero.
E52E  CD E559         readid: call    motor
E531  3A FFB8              ld      a,(mode)
E534  E6 40                and     40h             ;get mf bit from mode
E536  F6 0A                or      0ah             ;and or in command
E538  CD E484              call    fddrw
E53B  3A FFB9              ld      a,(hdr)
E53E  CD E484              call    fddrw
E541  18 C5                jr      fdcdn
                      ;
                      ;drvst returns the status a drive.
E543  CD E559         drvst:  call    motor
E546  3E 04                ld      a,4             ;sense drive status cmd
E548  CD E484              call    fddrw
E54B  3A FFB9              ld      a,(hdr)         ;get head/drive
E54E  CD E484              call    fddrw
E551  DB 0C           dst:    in      a,(fdcst)
E553  07                   rlca                    ;wait for rqm
E554  30 FB                jr      nc,dst
E556  DB 0D                in      a,(fdcdt)       ;get st3
E558  C9                   ret
                      ;
                      ;motor is the routine that turns on the dc motor on mini-
                      ;floppy drives.  /mc is the motor control line, located
                      ;on pio b, bit 2.
E559  AF              motor:  xor     a
E55A  32 FFB7              ld      (fdone),a       ;reset fdc done flag
E55D  3A FFB6              ld      a,(drvsz)
E560  B7                   or      a               ;mini drive?
                      ;
                           if      persci or tandon
                           nop                     ;ignore drive size
                           else
E561  C0                   ret     nz              ;return on 8"
                           endif
                      ;
E562  3E 1E                ld      a,30            ;set timer
E564  32 FFA7              ld      (mtm),a         ;for 30 seconds
E567  DB 06                in      a,(pbd)
E569  CB 57                bit     2,a             ;motor already on?
E56B  C8                   ret     z               ;yes
E56C  CB 97                res     2,a             ;no,
```

```
E56E    D3 06                   out     (pbd),a         ;turn motor on
E570    3A FFA7         w2sec:  ld      a,(mtm)         ;and wait 2 seconds
E573    FE 1C                   cp      28
E575    20 F9                   jr      nz,w2sec
E577    C9                      ret
                        subttl On-board interrupt routines
                        page
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-24
On-board interrupt routines

                                ;
                                ;interrupt routine for sio b.
  E578    ED 73 FF98            srcb:   ld      (tssp),sp       ;save sp
  E57C    31 FF98                       ld      sp,intsp
  E57F    F5                            push    af              ;special receive condition
  E580    3E 30                         ld      a,30h           ;reset error
  E582    D3 03                         out     (sbc),a
  E584    C3 E60C                       jp      ioret
                                ;
                                ;interrupt routines for sio a.
  E587    ED 73 FF98            rcaa:   ld      (tssp),sp       ;save sp
  E58B    31 FF98                       ld      sp,intsp
  E58E    F5                            push    af              ;receive char available
  E58F    3A FF9C                       ld      a,(iobyte)
  E592    B7                            or      a               ;i/o assigned yet?
  E593    28 06                         jr      z,ios?          ;no
  E595    CB 6F                         bit     5,a
  E597    DB 00                         in      a,(sad)
  E599    18 58                         jr      filque?
  E59B    DB 00                 ios?:   in      a,(sad)         ;assign i/o to sio a?
  E59D    E6 7F                         and     7fh
  E59F    FE 0D                         cp      cr              ;only if cr
  E5A1    20 0B                         jr      nz,chgbd
  E5A3    3E 10                         ld      a,10h           ;reset ext/status
  E5A5    D3 01                         out     (sac),a
  E5A7    3E 22                         ld      a,22h           ;assign i/o to sio a
  E5A9    32 FF9C               aio:    ld      (iobyte),a
  E5AC    18 5E                         jr      ioret
  E5AE    C5                    chgbd:  push    bc              ;change baud rate
  E5AF    E5                            push    hl
  E5B0    21 FFA5                       ld      hl,bctr
  E5B3    35                            dec     (hl)            ;decrement counter
  E5B4    20 0D                         jr      nz,chgb
  E5B6    36 02                         ld      (hl),2          ;reset counter
  E5B8    01 0208                       ld      bc,2*256+ctc0
  E5BB    2A FFA3                       ld      hl,(bptr)       ;get pointer,
  E5BE    ED B3                         otir                    ;change baud rate,
  E5C0    22 FFA3                       ld      (bptr),hl       ;and save pointer
  E5C3    E1                    chgb:   pop     hl
  E5C4    C1                            pop     bc
  E5C5    18 45                         jr      ioret
                                ;
  E5C7    ED 73 FF98            srca:   ld      (tssp),sp       ;save sp
  E5CB    31 FF98                       ld      sp,intsp
  E5CE    F5                            push    af              ;special receive cond
  E5CF    3E 30                         ld      a,30h           ;reset error
  E5D1    D3 01                         out     (sac),a
  E5D3    18 37                         jr      ioret
                                ;
```

```
                        ;kbdin is the interrupt routine for the keyboard (pio a).
E5D5  ED 73 FF98  kbdin:  ld    (tssp),sp       ;save sp
E5D9  31 FF98             ld    sp,intsp
E5DC  F5                  push  af
E5DD  3A FF9C             ld    a,(iobyte)
E5E0  B7                  or    a               ;i/o assigned yet?
E5E1  20 0C               jr    nz,kbin
E5E3  DB 04               in    a,(pad)         ;assign i/o to keyboard
E5E5  E6 7F               and   7fh
E5E7  FE 0D               cp    cr
E5E9  20 21               jr    nz,ioret
E5EB  3E 18               ld    a,18h           ;assign i/o to
E5ED  18 BA               jr    aio             ;keyboard & video
E5EF  CB 67       kbin:   bit   4,a
E5F1  DB 04               in    a,(pad)
E5F3              filque?: ;fill the input queue?
E5F3  28 17               jr    z,ioret         ;no
E5F5  E5                  push  hl              ;yes
E5F6  2A FF9D             ld    hl,(quein)
E5F9  77                  ld    (hl),a
E5FA  2C                  inc   l
E5FB  3E 80               ld    a,low quend
E5FD  BD                  cp    l
E5FE  20 02               jr    nz,flque
E600  2E 40               ld    l,low queue
E602  3A FF9F     flque:  ld    a,(queout)
E605  BD                  cp    l               ;queue full?
E606  28 03               jr    z,fque          ;yes
E608  22 FF9D             ld    (quein),hl      ;no
E60B  E1          fque:   pop   hl
E60C  F1          ioret:  pop   af
E60D  ED 7B FF98          ld    sp,(tssp)       ;restore sp
E611  FB          jret:   ei
E612  ED 4D               reti
                        ;
                        ;
                        ;pint is the interrupt routine when pio a is an output port.
E614  ED 73 FF98  pint:   ld    (tssp),sp       ;save sp
E618  31 FF98             ld    sp,intsp
E61B  F5                  push  af
E61C  3E FF               ld    a,-1
E61E  32 FFA2             ld    (outst),a       ;reset status
E621  18 E9               jr    ioret
                        ;
                        ;timer is the routine that keeps the real time clock.  the
                        ;time is kept in binary and is a 24 hour clock.  one option
                        ;is provided for the user if he wants to implement it:
                        ;ustime, this routine is for printing the time (or any other
                        ;function that the user desires).  upon entry, hl points to
                        ;sec in the system ram.  this routine is called only if the
                        ;time display flag (tdf) is set in the system ram.  the user
```

```
                        ;must save any registers he uses except for af and hl.
                        ;also, only 16 bytes of stack is available.  timer also
                        ;keeps a date calender.  it will update the day, month and
                        ;year whenever the time rolls over the 24 hours.  it does
                        ;not keep track of leap year nor centuries.
E623  ED 73 FF98        timer:  ld      (tssp),sp       ;save sp
E627  31 FF98                   ld      sp,intsp        ;set up interrupt stack
E62A  F5                        push    af
E62B  E5                        push    hl
E62C  21 FFA7                   ld      hl,mtm
E62F  7E                        ld      a,(hl)          ;get motor time
E630  B7                        or      a               ;already zero?
E631  28 09                     jr      z,tim0          ;yes, don't do anything
E633  35                        dec     (hl)            ;decrement time
E634  20 06                     jr      nz,tim0
E636  DB 06                     in      a,(pbd)         ;if zero, turn off motor
E638  CB D7                     set     2,a             ;reset /mc
E63A  D3 06                     out     (pbd),a
E63C  23                tim0:   inc     hl
E63D  7E                        ld      a,(hl)          ;get user time-out
E63E  B7                        or      a
E63F  28 01                     jr      z,tim1
E641  35                        dec     (hl)
E642  3E 3B             tim1:   ld      a,59
E644  23                        inc     hl
E645  34                        inc     (hl)            ;increment seconds
E646  BE                        cp      (hl)
E647  30 37                     jr      nc,tim4
E649  36 00                     ld      (hl),0
E64B  23                        inc     hl
E64C  34                        inc     (hl)            ;increment minutes
E64D  BE                        cp      (hl)
E64E  30 30                     jr      nc,tim4
E650  36 00                     ld      (hl),0
E652  3E 17                     ld      a,23
E654  23                        inc     hl
E655  34                        inc     (hl)            ;increment hours
E656  BE                        cp      (hl)
E657  30 27                     jr      nc,tim4
E659  36 00                     ld      (hl),0
E65B  C5                        push    bc
E65C  D5                        push    de
E65D  23                        inc     hl
E65E  23                        inc     hl              ;point to day
E65F  3A FFAE                   ld      a,(month)       ;get month
E662  4F                        ld      c,a
E663  06 00                     ld      b,0
E665  11 E776                   ld      de,dpm-1        ;point to day/month table
E668  EB                        ex      de,hl
E669  09                        add     hl,bc
E66A  EB                        ex      de,hl
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-27
On-board interrupt routines

E66B    1A                      ld      a,(de)          ;get number of days
E66C    34                      inc     (hl)            ;increment day
E66D    BE                      cp      (hl)            ;next month?
E66E    30 0E                   jr      nc,tim3         ;no
E670    36 01                   ld      (hl),1          ;yes, day=1
E672    23                      inc     hl
E673    34                      inc     (hl)            ;increment month
E674    3E 0C                   ld      a,12
E676    BE                      cp      (hl)            ;next year?
E677    30 05                   jr      nc,tim3         ;no
E679    36 01                   ld      (hl),1          ;yes, month=1
E67B    2B                      dec     hl
E67C    2B                      dec     hl
E67D    34                      inc     (hl)            ;year= year+1
E67E    D1              tim3:   pop     de
E67F    C1                      pop     bc
E680    21 FFA9         tim4:   ld      hl,sec          ;point to sec
E683    3A FFA6                 ld      a,(tdf)
E686    B7                      or      a
E687    C4 E6F0                 call    nz,ustime
E68A    E1                      pop     hl
E68B    C3 E60C                 jp      ioret
                        ;
                        ;fdcint is the interrupt routine for the upd765 chip.
E68E    ED 73 FF98      fdcint: ld      (tssp),sp       ;save sp
E692    E3                      ex      (sp),hl
E693    22 FF9A                 ld      (tshl),hl
E696    31 FF98                 ld      sp,intsp        ;set up interrupt stack
E699    F5                      push    af
E69A    C5                      push    bc
E69B    21 FFC2                 ld      hl,st0
E69E    01 080D                 ld      bc,8*256+fdcdt  ;b=08, c=0d
E6A1    DB 0C                   in      a,(fdcst)
E6A3    E6 40                   and     40h             ;dio=1?
E6A5    20 0A                   jr      nz,fdci2        ;yes
E6A7    ED 41                   out     (c),b           ;do sense interrupt status
E6A9    DB 0C           fdci1:  in      a,(fdcst)       ;get st0,st1,st2,c,h,r & n
E6AB    07                      rlca
E6AC    30 FB                   jr      nc,fdci1        ;wait for rqm
E6AE    07                      rlca
E6AF    30 04                   jr      nc,fdci3
E6B1    ED A2           fdci2:  ini
E6B3    18 F4                   jr      fdci1
E6B5    10 15           fdci3:  djnz    fdci6           ;jp if sense int was exec
E6B7    CD E543                 call    drvst           ;to set seek/rw on 765
E6BA    3E FF           fdci4:  ld      a,-1
E6BC    32 FFB7                 ld      (fdone),a       ;set fdc done flag
E6BF    21 E508                 ld      hl,fdcdn        ;make fdcdn return addr
E6C2    C1              fdci5:  pop     bc
E6C3    F1                      pop     af
E6C4    ED 7B FF98              ld      sp,(tssp)       ;restore stack
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-28
On-board interrupt routines

E6C8    E3                      ex      (sp),hl         ;& return address
E6C9    FB                      ei
E6CA    ED 4D                   reti
E6CC    3A FFC2         fdci6:  ld      a,(st0)         ;sense int was executed
E6CF    CB 6F                   bit     5,a             ;seek end?
E6D1    20 E7                   jr      nz,fdci4        ;yes
E6D3    E6 03                   and     3               ;no, must be change in the
E6D5    4F                      ld      c,a             ;ready line
E6D6    06 00                   ld      b,0             ;bc= drive
E6D8    3A FFC2                 ld      a,(st0)
E6DB    CB 5F                   bit     3,a             ;drive ready?
E6DD    3E 00                   ld      a,0
E6DF    20 01                   jr      nz,fdci7        ;no
E6E1    2F                      cpl                     ;yes
E6E2    21 FFB0         fdci7:  ld      hl,drdy0        ;point to appropiate
E6E5    09                      add     hl,bc           ;drive ready flag
E6E6    77                      ld      (hl),a          ;and (re)set it
E6E7    2A FF9A                 ld      hl,(tshl)       ;get return addr
E6EA    18 D6                   jr      fdci5
                        subttl Optional user routines
                        page
```

```
                        ;
                        ;the following routine may be implemented by programing a
                        ;nop (00) instead of the ret and the address of the routine
                        ;instead of the n$i (ffff) address.
                        ;
                        ;usout is a user output routine.  it is called whenever the
                        ;output routine is called (outch) and bit 3 of the iobyte is
                        ;set.  this bit is set if upon initialization a cr is
                        ;received from the parallel port (pio a) or by the user
                        ;through a command or the asio routine.
E6EC    C9              usout:  ret                     ;user output routine (n$i)
E6ED    C3 FFFF                 jp      n$i
                        ;
                        ;ustime is a user time routine.  it is called if tdf is set
                        ;from the timer routine.  see the timer routine.
E6F0    C9              ustime: ret                     ;user time routine (n$i)
E6F1    C3 FFFF                 jp      n$i
                                subttl Initialization & I/O data
                                page
```

```
                                ;
                                ;set-up data for pio a.
E6F4  03                        piotb:  db      3
E6F5  F8                        pidt:   db      0f8h            ;interrupt vector
E6F6  4F                                db      4fh             ;input
E6F7  87                                db      87h             ;enable int's
                                ;
                                ;set-up data for pio b.
E6F8  01                                db      1
E6F9  21                                db      21h
E6FA  04                                db      4
E6FB  FA                                db      0fah            ;interrupt vector
E6FC  FF                                db      0ffh            ;mode 3
E6FD  88                                db      88h             ;control word
E6FE  07                                db      7               ;no interrupts
                                ;
                                ;set-up data for ctc 0-3.
0047                            ct      equ     47h             ;counter
0007                            tm      equ     7               ;timer
E6FF  03                                db      3
E700  F0                                db      0f0h            ;interrupt vector
E701  47                                db      ct              ;ch 0: no int's, counter
E702  0D                                db      13              ;ch 0 time const (9600
E703  02                                db      2
E704  07                                db      tm              ;ch 1: no int's ,timer (/16)
E705  34                                db      52              ;ch 0 time const (300 baud)
E706  02                                db      2
E707  27                                db      27h             ;ch 2: no int's, timer(/256)
E708  7D                                db      7dh             ;ch 2 time const
E709  02                                db      2
E70A  C7                                db      0c7h            ;ch 3: int's & counter
E70B  7D                                db      7dh             ;ch 3 time const
                                ;
                                ;set-up data for sio a.
E70C  09                                db      9
E70D  18                                db      18h             ;reset chan
E70E  14                                db      14h             ;  "   ext stat
E70F  4C                                db      4ch             ;x16, 2 stop, no parity
E710  03                                db      3
E711  E1                                db      0e1h            ;8 bits, rx ena, auto ena
E712  05                                db      5
E713  EA                                db      0eah            ;8 bits, dtr, rts, tx ena
E714  11                                db      11h             ;reset ext stat
E715  18                                db      18h             ;int rx, no tx or ext int
                                ;
                                ;set-up data for sio b.
E716  0B                                db      11
E717  18                                db      18h             ;reset chan
E718  02                                db      2
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-31
Initialization & I/O data

E719    E0                      db      0e0h            ;interrupt vector
E71A    14                      db      14h             ;reset ext stat
E71B    4C                      db      4ch             ;x16, 2 stop, no parity
E71C    03                      db      3
E71D    E1                      db      0e1h            ;8 bits, rx enable
E71E    05                      db      5
E71F    EA                      db      0eah            ;8 bits, dtr, rts, tx ena
E720    11                      db      11h             ;reset ext stat
E721    04                      db      04h             ;no rx/tx int, stat affects
                        ;
                        ;interrupt service routine vector table.  the vectors
                        ;pointing to "jret" just return from interrupt, they are
                        ;not normally used in the fdc-i monitor.
E722    E611            ivt:    dw      jret            ;sio b, tx buffer empty
E724    E611                    dw      jret            ; "  b, ext/status change
E726    E611                    dw      jret            ; "  b, receive char avail
E728    E578                    dw      srcb            ; "  b, special receive cond
E72A    E611                    dw      jret            ;sio a, tx buffer empty
E72C    E611                    dw      jret            ;  "  , ext/status change
E72E    E587                    dw      rcaa            ;  "  , receive char avail
E730    E5C7                    dw      srca            ;  "  , special receive cond
E732    E611                    dw      jret            ;ctc0
E734    E611                    dw      jret            ;ctc1
E736    E611                    dw      jret            ;ctc2, real-time clock
E738    E623                    dw      timer           ;ctc3,   "   "    "
E73A    E5D5                    dw      kbdin           ;pio a, input (keyboard)
E73C    E611                    dw      jret            ;pio b
E73E    E614                    dw      pint            ;pio a, output (centronics)
E740    FF                      db      -1              ;adjust for upd765 interrupt
E741    E68E                    dw      fdcint          ;upd765
0021                    itl     equ     $-ivt
                        ;
                        ;set-up data to initialize the system ram.
E743    00              srdat:  db      0               ;no i/o assigned
E744    FF40                    dw      queue           ;queue in =
E746    FF40                    dw      queue           ;queue out
E748    00                      db      0               ;no inp data avail, sio b
E749    FF                      db      -1              ;pio a output ready
E74A    E752                    dw      bdtb            ;baud rate pointer
E74C    02                      db      2               ;  "    "  change counter
E74D    00                      db      0               ;reset time display
E74E    00                      db      0               ;reset mini-disk motor time
E74F    1E                      db      30              ;30 sec for user time-out
000D                    srl     equ     $-srdat
                        ;
                        ;baud rate table
E750    47 0D           baudtb: db      ct,13           ;9600 baud
E752    47 1A           bdtb:   db      ct,26           ;4800  "
E754    47 34                   db      ct,52           ;2400  "
E756    47 68                   db      ct,104          ;1200  "
E758    47 D0                   db      ct,208          ; 600  "
```

```
E75A    07 34                   db      tm,52           ; 300  "
E75C    07 68                   db      tm,104          ; 150  "
E75E    07 8E                   db      tm,142          ; 110  "
                        ;
                        ;set-up data for pio a as an output port.
E760    FC              podt:   db      0fch            ;interrupt vector
E761    0F                      db      0fh             ;output
E762    87                      db      87h             ;enable interrupt
                        ;
                        ;read/write set-up data for 5.25" drives.
E763    00              fm5:    db      0               ;single density n
E764    12                      db      18              ;  "      "     eot
E765    05                      db      5               ;  "      "     gpl
E766    80                      db      128             ;  "      "     dtl
E767    80                      db      128             ;  "      "     sector size
E768    01                      db      1               ;double   "     n
E769    12                      db      18              ;  "      "     eot
E76A    0A                      db      10              ;  "      "     gpl
E76B    FF                      db      -1              ;  "      "     dtl
E76C    00                      db      256 mod 256     ;  "      "     sector size
                        ;
                        ;read/write set-up data for 8" drives.
E76D    00              fm8:    db      0               ;single density n
E76E    1A                      db      26              ;  "      "     eot
E76F    07                      db      7               ;  "      "     gpl
E770    80                      db      128             ;  "      "     dtl
E771    80                      db      128             ;  "      "     sector size
E772    01                      db      1               ;double   "     n
E773    1A                      db      26              ;  "      "     eot
E774    0E                      db      14              ;  "      "     gpl
E775    FF                      db      -1              ;  "      "     dtl
E776    00                      db      256 mod 256     ;  "      "     sector size
                        ;
                        ;dpm contains the number of days per month.
E777    1F              dpm:    db      31              ;january
E778    1C                      db      28              ;febuary
E779    1F                      db      31              ;march
E77A    1E                      db      30              ;april
E77B    1F                      db      31              ;may
E77C    1E                      db      30              ;june
E77D    1F                      db      31              ;july
E77E    1F                      db      31              ;august
E77F    1E                      db      30              ;september
E780    1F                      db      31              ;october
E781    1E                      db      30              ;november
E782    1F                      db      31              ;december
                                subttl Monitor messages and command table
                                page
```

```
                                ;
                                ;monitor messages:
E783    0D 0A 46 44             som:    db      cr,lf,'FDC-I '
E787    43 2D 49 20
E78B    34 2E                           db      rls/10+'0','.'
E78D    30 61                           db      rls mod 10 + '0',vrsn
E78F    20 4D 6F 6E                     dc      ' Monitor'
E793    69 74 6F F2
E797    0D 0A                   prmt:   db      cr,lf
E799    3E A0                           dc      '> '
E79B    20 5E D8                cmsg:   dc      ' ^X'
E79E    41 64 64 72             addr:   dc      'Addr '
E7A2    A0
E7A3    26 20 23 20                     dc      '& # of sctr''s '
E7A7    6F 66 20 73
E7AB    63 74 72 27
E7AF    73 A0
                                ;
                                ;monitor command table:
E7B1    D8                      cmdtb:  db      2*'B'+'T'
E7B2    E0B0                            dw      boot
E7B4    C9                              db      2*'D'+'A'
E7B5    E1C0                            dw      da
E7B7    DB                              db      2*'D'+'S'
E7B8    E1F2                            dw      drst
E7BA    AA                              db      2*'E'+' '
E7BB    E1FE                            dw      em
E7BD    AC                              db      2*'F'+' '
E7BE    E241                            dw      fill
E7C0    AE                              db      2*'G'+' '
E7C1    E25E                            dw      go
E7C3    B2                              db      2*'I'+' '
E7C4    E263                            dw      inp
E7C6    BA                              db      2*'M'+' '
E7C7    E274                            dw      move
E7C9    BE                              db      2*'O'+' '
E7CA    E2BD                            dw      out
E7CC    C0                              db      2*'P'+' '
E7CD    E2CC                            dw      prog
E7CF    E7                              db      2*'R'+'C'
E7D0    E2E6                            dw      rec
E7D2    E8                              db      2*'R'+'D'
E7D3    E2F6                            dw      rdatd
E7D5    F7                              db      2*'R'+'S'
E7D6    E2FA                            dw      rdats
E7D8    EA                              db      2*'S'+'D'
E7D9    E334                            dw      sd
E7DB    F9                              db      2*'S'+'S'
E7DC    E346                            dw      stsp
```

```
E7DE    CC                      db      2*'V'+' '
E7DF    E282                    dw      verfy
E7E1    F2                      db      2*'W'+'D'
E7E2    E357                    dw      wdatd
E7E4    01                      db      (2*'W'+'S') mod 256
E7E5    E35B                    dw      wdats
0012                    nocmd   equ     ($-cmdtb)/3     ;no. of commands
                        ;
07E7                    mlen    equ     $-mon           ;monitor length
                                .list
                                .dephase
                        ;
                                subttl System RAM Definition
                                page
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80      PAGE   1-35
System RAM Definition


                        ;
                                .phase  stack
                        ;
FF40                    queue:  ds      64              ;64 character input queue
FF80                    quend   equ     $               ;queue end
                        ;
FF80                            ds      24              ;interrupt stack space
FF98                    intsp   equ     $               ;    "       "   pointer
FF98                    tssp:   ds      2               ;tempo storage for sp
FF9A                    tshl:   ds      2               ;  "      "     " hl
                        ;
FF9C                    iobyte: ds      1               ;monitor i/o byte
                        ;
FF9D                    quein:  ds      2               ;queue in pointer
FF9F                    queout: ds      2               ;queue out pointer
FFA1                    instb:  ds      1               ;input status sio b
FFA2                    outst:  ds      1               ;output status pio a
                        ;
FFA3                    bptr:   ds      2               ;baud rate pointer
FFA5                    bctr:   ds      1               ;  "    "  change counter
                        ;
FFA3                    cptr    equ     bptr            ;conv routine pointer
FFA5                    vcnt    equ     bctr            ;verify errors count
                        ;
FFA6                    tdf:    ds      1               ;time display flag
FFA7                    mtm:    ds      1               ;mini-floppy motor time
FFA8                    ustm:   ds      1               ;user time-out
FFA9                    sec:    ds      1               ;real-time clock, seconds
FFAA                    min:    ds      1               ;  "    "    "  , minutes
FFAB                    hour:   ds      1               ;  "    "    "  , hours
FFAC                    year:   ds      1               ;  "    "    "  , year
FFAD                    day:    ds      1               ;  "    "    "  , day
FFAE                    month:  ds      1               ;  "    "    "  , month
                        ;
FFAF                    indtb:  ds      1               ;input sio b data
                        ;
FFB0                    drdy0:  ds      1               ;fdc, drive 0 ready flag
FFB1                    drdy1:  ds      1               ;  " ,    "   1   "     "
FFB2                    drdy2:  ds      1               ;  " ,    "   2   "     "
FFB3                    drdy3:  ds      1               ;  " ,    "   3   "     "
FFB4                    rwdt:   ds      2               ;  " , read/write data ptr
FFB6                    drvsz:  ds      1               ;  " , drive size
FFB7                    fdone:  ds      1               ;  " , done flag
FFB8                    mode:   ds      1               ;  " , mode
FFB9                    hdr:    ds      1               ;  " , head/drive
FFBA                    trk:    ds      1               ;  " , track no (c)
FFBB                    hd:     ds      1               ;  " , head (h)
FFBC                    sect:   ds      1               ;  " , sector no (r)
FFBD                    n:      ds      1               ;  " , bytes/sector (n)
```

```
FDC-I Monitor, Release 4.0      MACRO-80 3.4    01-Dec-80       PAGE    1-36
System RAM Definition

  FFBE                  eot:    ds      1               ; " , end of track
  FFBF                  gpl:    ds      1               ; " , gap length
  FFC0                  dtl:    ds      1               ; " , data length
  FFC1                  rsz:    ds      1               ; " , sector size
  FFC2                  st0:    ds      1               ; " , st 0
  FFC3                  st1:    ds      1               ; " , st 1 (can also be pcn)
  FFC4                  st2:    ds      1               ; " , st 2
  FFC5                  rc:     ds      1               ; " , result phase c
  FFC6                  rh:     ds      1               ; " ,   "      "  h
  FFC7                  rr:     ds      1               ; " ,   "      "  r
  FFC8                  rn:     ds      1               ; " ,   "      "  n
                        ;
  0037                  cbl     equ     sysram+512-$    ;command buffer length
  FFC9                  cbuff:  ds      cbl             ;   "      "
                                .dephase
                                subttl Symbol Table
                        ;
                                end
```

FDC-I Monitor, Release 4.0 MACRO-80 3.4 01-Dec-80 PAGE S
Symbol Table

Macros:

Symbols:

ADDR	E79E	AIO	E5A9	ASHEX	E17B	ASHX0	E181
ASHX1	E193	ASIO	E37C	ASP0	E38F	ASPA	E380
AUTO	0000	BAUDTB	E750	BCTR	FFA5	BDTB	E752
BKLEN	E24F	BOOT	E0B0	BPTR	FFA3	BS	0008
BTO	E0BB	CAN	0018	CANCEL	E167	CBL	0037
CBUFF	FFC9	CHGB	E5C3	CHGBD	E5AE	CMD1	E0FC
CMD2	E109	CMD3	E119	CMDTB	E7B1	CMSG	E79B
CNCL	E172	CONV	E195	CONV1	E19D	CONV2	E1AC
CONVE	E1B2	CPTR	FFA3	CR	000D	CRLF	E3ED
CT	0047	CTC0	0008	DA	E1C0	DA0	E1C4
DA1	E1CA	DA2	E1D4	DAY	FFAD	DEL	007F
DMA	0010	DOTC	E506	DPM	E777	DRDY0	FFB0
DRDY1	FFB1	DRDY2	FFB2	DRDY3	FFB3	DRST	E1F2
DRVST	E543	DRVSZ	FFB6	DST	E551	DTL	FFC0
EIGHT	FFFF	EM	E1FE	EM1	E201	EM4	E22E
EM5	E233	EM6	E238	EOT	FFBE	FALSE	0000
FDCDN	E508	FDCDT	000D	FDCI1	E6A9	FDCI2	E6B1
FDCI3	E6B5	FDCI4	E6BA	FDCI5	E6C2	FDCI6	E6CC
FDCI7	E6E2	FDCINT	E68E	FDCST	000C	FDDR	E485
FDDRW	E484	FDONE	FFB7	FIL	E247	FILL	E241
FILQUE	E5F3	FIVT	FEE0	FLQUE	E602	FM5	E763
FM8	E76D	FQUE	E60B	GO	E25E	GPL	FFBF
GSTAT	E3AB	GTDAT	E31B	HD	FFBB	HDR	FFB9
HOUR	FFAB	IN	E267	INCMD	E0E6	INDTB	FFAF
INIT	E05D	INIT1	E064	INIT2	E070	INIT3	E0A0
INP	E263	INPM	E3BC	INPT	E3DE	INPUT	E3CA
INST1	E12E	INST2	E135	INST3	E152	INST4	E157
INST5	E15D	INST6	E164	INSTB	FFA1	INSTR	E121
INTSP	FF98	IOBYTE	FF9C	IORET	E60C	IOS?	E59B
ITL	0021	IV	FFFE	IVT	E722	JRET	E611
KBDIN	E5D5	KBIN	E5EF	LDSP	E0E0	LF	000A
MIN	FFAA	MLEN	07E7	MODE	FFB8	MON	E000
MONTH	FFAE	MOTOR	E559	MOVE	E274	MTM	FFA7
N	FFBD	N$I	FFFF	NOCMD	0012	OT	E2C2
OUT	E2BD	OUTCH	E413	OUTST	FFA2	PAC	0005
PAD	0004	PBD	0006	PERSCI	0000	PIDT	E6F5
PINT	E614	PIOTB	E6F4	PODT	E760	POUT	E443
POUTO	E444	PRINT	E1E1	PRMT	E797	PROG	E2CC
PROG1	E2D4	PSTR	E1B7	QUEIN	FF9D	QUEND	FF80
QUEOUT	FF9F	QUEUE	FF40	RC	FFC5	RCAA	E587
RDAT	E2FB	RDATD	E2F6	RDATS	E2FA	RDF	E4EC
RDFD	E4EB	READ	E4E6	READID	E52E	REC	E2E6
RECAL	E4B5	RH	FFC6	RLS	0028	RN	FFC8
RR	FFC7	RSZ	FFC1	RTAS	E40E	RTDE	E23B
RTDRDY	E3A3	RTHEX	E3F6	RTHL	E409	RTHX	E3FF
RTIVT	E3A7	RTR	E313	RTRES	E30B	RTSEC	E39F
RTSTO	E39B	RWDT	FFB4	SA	E439	SAC	0001
SAD	0000	SAOUT	E438	SB	E42E	SBC	0003
SBD	0002	SBIN	E3E5	SBOUT	E42D	SD	E334

FDC-I Monitor, Release 4.0 MACRO-80 3.4 01-Dec-80 PAGE S-1
Symbol Table

SDRV	E337	SEC	FFA9	SECT	FFBC	SEEK	E4CB
SHD	E4AD	SKIP	E1DF	SOM	E783	SPACE	E411
SPEC	E478	SRCA	E5C7	SRCB	E578	SRDAT	E743
SRL	000D	SRTDF	E397	ST0	FFC2	ST1	FFC3
ST2	FFC4	STACK	FF40	STATB	E3C2	STATIN	E3B8
STF1	E521	STFDC	E514	STHDR	E4A4	STMODE	E48E
STMU	E4A1	STS0	E463	STS1	E472	STS2	E474
STSECT	E4B1	STSIZE	E45A	STSP	E346	STSSP	E34B
SYSRAM	FE00	TANDON	0000	TC	0014	TDF	FFA6
TIM0	E63C	TIM1	E642	TIM3	E67E	TIM4	E680
TIMER	E623	TM	0007	TRK	FFBA	TRUE	FFFF
TSHL	FF9A	TSSP	FF98	USOUT	E6EC	USR	FFFF
USTIME	E6F0	USTM	FFA8	VCNT	FFA5	VER0	E289
VER1	E28B	VER2	E28E	VER3	E2AC	VERF	E288
VERFY	E282	VRSN	0061	W2SEC	E570	WAIT	0018
WARM	E0DA	WDAT	E35C	WDATD	E357	WDATS	E35B
WDT1	E36F	WDT2	E375	WRITE	E4F7	WTF	E4FD
WTFD	E4FC	WTIO	E0D4	XCMD	E261	YEAR	FFAC

No Fatal error(s)