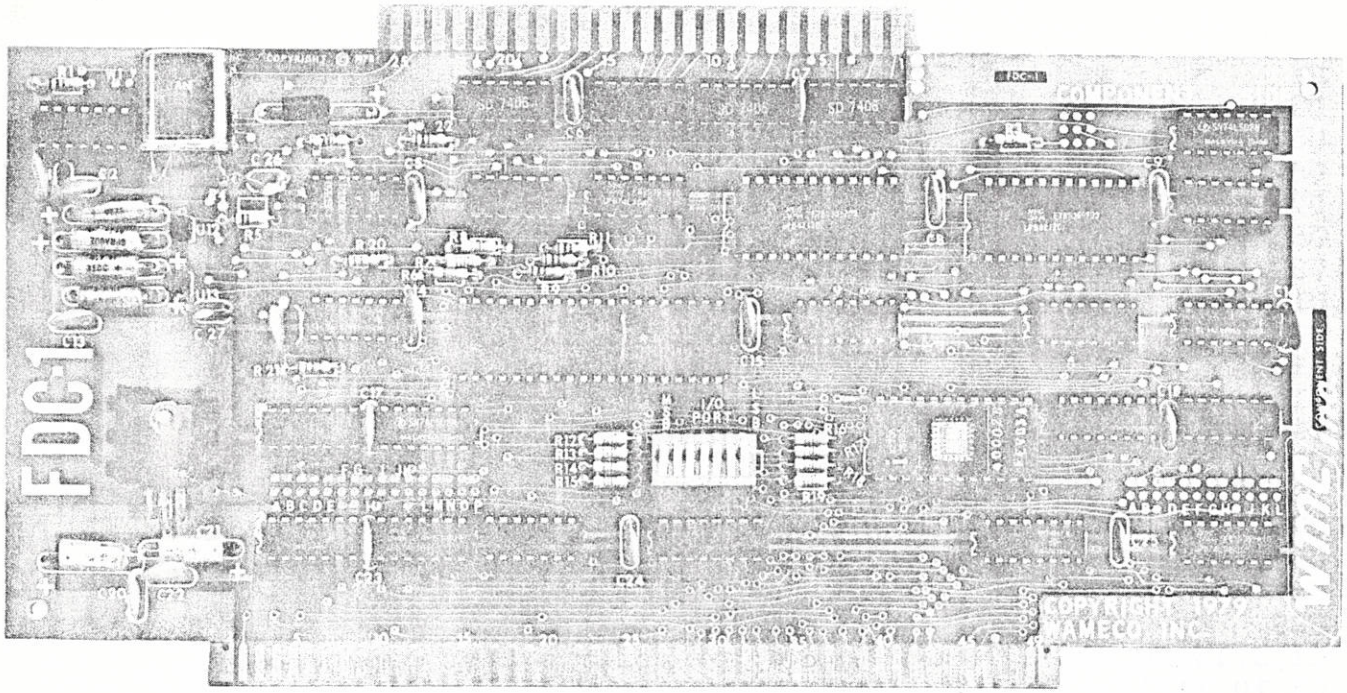
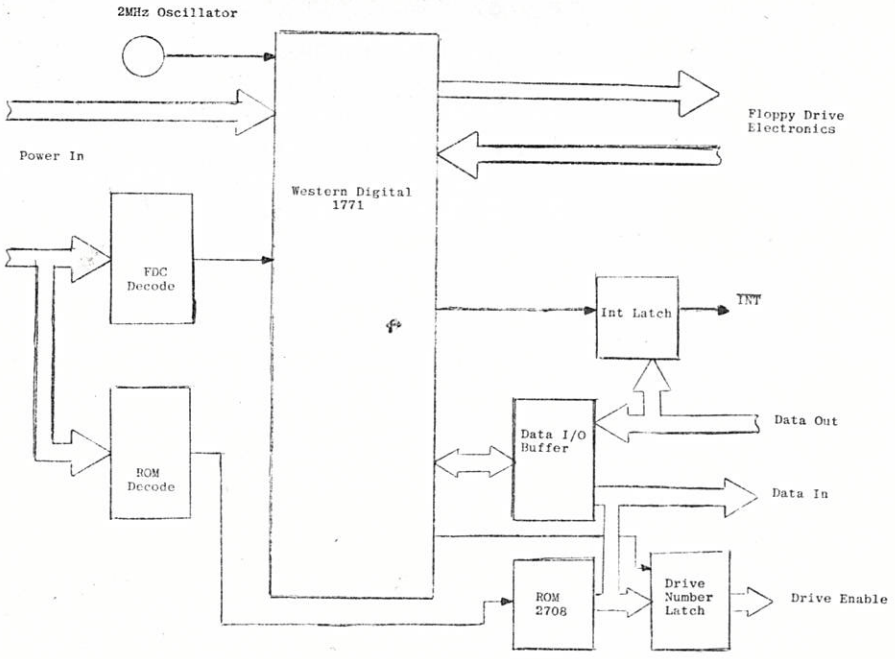
FDC-1

FLOPPY DISC CONTROLLER BOARD

wmc inc.



The WAMECO Floppy Disc Board is designed for the serious micro-computer user. The FDC-1 may be operated with either SHUGART, PERTEC or REMEX standard or mini drives. The board is also designed to operate with CPM (tm) software thus giving the computer user access to a powerful operating system. Some main features include control of up to eight drives, on board PROM for cold boot-up, vector selection of interrupt and compatibility with either 8080A or Z-80 (2MHz or 4MHz) systems. USES WESTERN DIGITAL 1771.



FDC-1 BLOCK DIAGRAM

PARTS LIST

U1	1-	74LS38
U2,3,4,	3-	7406
U5,18,21	3-	74LS02
U6	1-	74121
U7	1-	7407
U8,25,26,30	4-	74LS04
U9,10	2-	8212, 74S412
U11,U14	2-	74LS00
U12	1-	79L05
U13	1-	78L12
U15	1-	FD1771
U16,17	2-	8T26
U19	1-	7805
U20,24	2-	74LS30
U22	1-	2708
U23,29	2-	8097
U27	1-	74LS74
U28	1-	8131
C1,2,13,20,22	5-	0.1uf ceramic disc capacitors
C3,10,11,12	4-	4.7uf 25v Tantalum capacitors
C4	1-	10uf 25v Tantalum capacitor
C5,6,7,8,9,14,15,16,17, 18,23,24,25,28	14-	.01uf 25v ceramic disc capacitors
C19,21	2-	22uf 25v Tantalum capacitors
C26,27	2-	27pf 25v ceramic disc capacitors
R1,5,9,10,11,22	6-	1K 5% ¼ watt resistors
R2	1-	5.1K "
R3,6,7,8,12-19	12-	2.7K "
R4	1-	1.8K "
R20,21	2-	560 Ohm "
RN1	1-	AB-314E221331 220/330 Resistor Terminator Package
	1-	40 pin socket
	3-	24 pin sockets
	18-	14 pin sockets
	5-	16 pin sockets
	1-	7 position DIP switch
	1-	TO 222 heatsink
	1-	6-32 x 3/8" nut, screw and lockwasher
	1-	2 MHz crystal

The FDC-1 floppy disk controller board is designed to allow the user to attach virtually any brand of flexible disk drive to a S-100 based 8080, Z80 system. The on-board 2708 EPROM contains all the commonly used disk subroutines which have been specifically tailored to meet the requirements of CP/M (Copyright, Digital Research). These routines include the routines "GETSYS" and "PUTSYS" as described in the "CP/M Alteration Guide" (Copyright, Digital Research).

The software is to be placed in a 2708 EPROM. Commonly used routines for console interaction, etc., can be located in the nearly 500 empty bytes.

This manual deals with the assembly, options and basic operation of the FDC-1. Other sections deal with additional features on this board. One section is the listing of the copyrighted software to be put in the 2708.

SECTION 1

Programming Model:

The FDC-1 floppy disk controller uses five output and four input ports. Four I/O ports are addressed in a block, and the fifth port (drive selection) is individually locatable. The function of the ports is outlined in the following table:

Address	Input	Output
0	Status	Command
1	Track #	Track #
2	Sector#	Sector#
3	Data	Data
XX	N/A	Drive#

Hardware Description:

The floppy disk controller utilizes the Western Digital FD1771 LSI controller. Complete disk handling for interface to CP/M (Copyright 1976, 1977, Digital Research, Inc.) can easily be accomplished in less than 256 bytes of ROM.

The hardware is fully buffered from the S-100 interface and the connections to the disk drives. Externally adjustable delays are incorporated to accommodate head load delays in the various brands of drives. Access rate (track to track) is controlled by the software imbedded in the access commands.

Interrupt vectoring for end of operation interrupt from the FD1771 chip is jumper selectable.

Drive selection is accomplished by using an 8-bit output port, with one line being assigned to each drive.

Assembly of FDC-1

Before placing any parts on the board, check the boards for any hairline shorts (slivers). All boards have been inspected at least three times before shipping. If any slivers are found, carefully cut and scrape them with an Exacto knife.

Place all the sockets in their positions on the top side of the board.

After positioning all sockets in place, put a book or other flat stiff object on top of the sockets. Hold the book tight against the board and turn them over so that the underside of the board is up. Press down on the board and solder one pin on each end of each socket. This will ensure the sockets are flat against the board. When tacking all sockets is completed, finish soldering all the other pins of the sockets.

NOTE

DO NOT PUT IC'S IN SOCKETS AT THIS TIME.
THEY WILL BE INSTALLED LATER.

Bend the leads on R3, R6-8, R12-19 (2.7K ohm RED, VIOLET, RED) and place in board. Check parts placement drawing (figure 1) for correct locations. Turn the board over and solder all the resistors. Clip the leads of the resistors flush with the underside of the board with the diagonal pliers.

Bend the leads on R1, R5, R9-11, R22 (1K ohm BROWN, BLACK, RED) and place in board. Check parts placement drawing (figure 1) for correct location. Turn the board and solder the resistors. Clip the leads of the resistors, flush with the underside of the board with the diagonal pliers.

Bend the leads on R4 (1.8K ohm BROWN, GRAY, RED) and R20, R21 (560 ohm GREEN, BLUE, BROWN) and place in board. Check parts placement drawing (figure 1) for correct location. Turn the board over and solder the resistors. Clip the leads of the resistors flush with the underside of the board with the diagonal pliers.

NOTE

DO NOT INSTALL R2, THE 5.1K RESISTOR AT THIS TIME.

CAUTION

CHECK DISC CAPACITORS FOR PROPER VALUE BEFORE INSERTING IN BOARD

Insert C1, C2, C13, C20, C22 (0.1uF) disc capacitors in the board. Check parts location drawing (fugure 1) for proper locations. Solder all the capacitors. Clip the leads flush with the underside of the board with the diagonal pliers.

Insert C5-C9, C14-C18, C23-C25, C28 (.01uF) and C26, C27 (27pF) in the board. Check parts location drawing (figure 1) for proper locations. Solder all the capacitors. Clip the leads flush with the underside of the board with the diagonal pliers.

Insert C19, C22 (22uF tantalum), C4 (10uF tantalum) and C3, C10-C12 (4.7uF tantalum) in place. Ensure that the polarities are correct. Check parts placement drawing (figure 1) for correct placement. Solder the capacitors in place. Clip the leads flush with the underside of the board with the diagonal pliers.

Put the seven position dip switch in place. Ensure that the switch is installed so that the off position is toward the gold fingers of the board. Solder the seven position dip switch in place.

Place X1 (2MHz crystal) in place. It is recommended that the leads be bent so that the crystal will be close to the board. If the crystal is to be placed so it will be close to the board, wrap the crystal with a layer of plastic insulation tape so that no shorting of traces will occur. Check parts placement drawing (figure 1) for correct placement. Bend the leads of the crystal to retain it in place until it is soldered. Turn the board over and rest it on books as before. Solder the crystal in place. Clip the leads flush with the underside of the board with the diagonal pliers.

Before installing the 7805 five volt regulator, it is recommended that it be tested for proper voltage regulation.

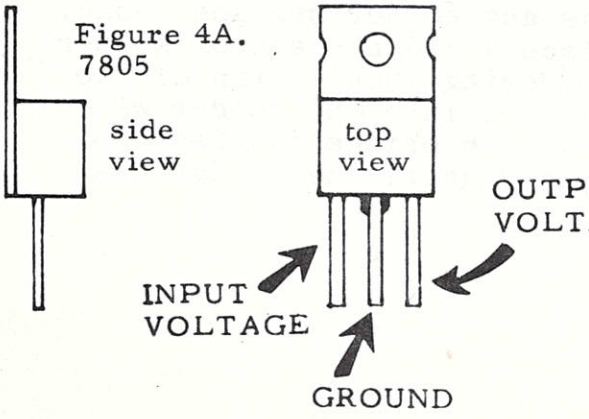


Figure 4A.
7805

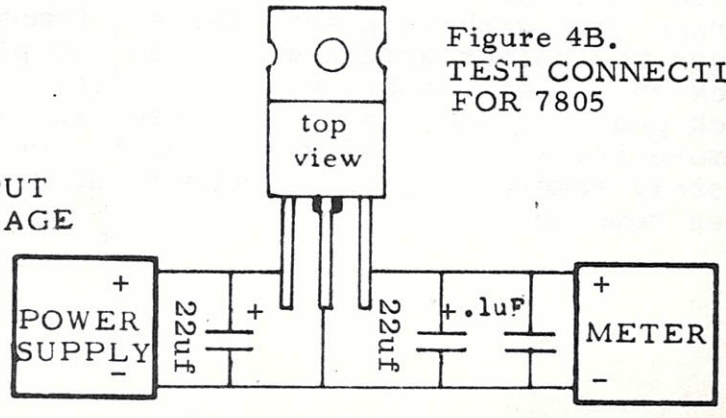


Figure 4B.
TEST CONNECTIONS
FOR 7805

To prevent oscillation of the regulators, assemble a test rig as shown. The capacitors must be installed observing correct polarity. This test rig is for pre-installation testing only. The filter capacitors installed on the board serve the same purpose in the final assembly.

Attach the power supply and multimeter to the 7805 as shown in figure 5. Place the multimeter in a DC voltage range that will allow 10 volts to be displayed. The regulator needs a 2.0 volt minimum difference between the input voltage and the regulated output voltage. If the power supply has a voltmeter, observe the input voltage during the test. If the power supply does not have a voltmeter, switch the + meter lead between the output lead and the input lead of the regulator. The input and regulated voltage can then be observed.

Slowly increase the input voltage and observe the output voltage. When the input voltage is between 7.0 and 7.5 volts, the regulated output of a properly operating 7805 should be between 4.8 and 5.2 volts. Replace the regulator if it does not meet these limits.

When the regulator has been tested, place the regulator on the board so that the mounting hole on the regulator lines up with the corresponding hole on the FDC-1. Check parts placement drawing (figure 1) for the correct placement of this regulator. Note where the leads of the regulator pass over the connection holes on the FDC-1. Bend the leads on the regulator so that the leads can be inserted into the proper holes. Mount the regulator on the board using a #6 nut and a 5/8" 6-30 screw. Insert a heatsink between the board and the 7805. Place a moderate amount of silicone thermal heat grease on the underside of the regulator and the underside of the heatsink.

Perform similar tests for the 79L05 and 78L12 regulators and install them in the correct locations. NOTE: The flat part of the regulators will face to the right side of the board. Solder these in place.

II. INSPECTION AND TESTING

Use a bright light and magnifying glass to inspect all the traces on the underside of the board. If any slivers are found, cut and scrape them with an Exacto knife. Remove any solder bridges found. Cover the solder bridge with flux and place a clean piece of solder wick on top of the bridge. Place the soldering iron on top of the wick and hold until solder is seen flowing up into the solder wick. Remove the iron and wick. Check to see if the bridge has been completely removed. If not, repeat the process until the bridge has been removed.

NOTE

AT THIS TIME NO IC'S HAVE BEEN INSTALLED ON THE BOARD. DO NOT INSTALL IC'S ON THE BOARD UNTIL CALLED FOR IN THE CHECK OUT PROCEDURE.

Place the multimeter in the R x 1 scale. Place one probe on the gold finger for pin 1. Place the other probe on all the other fingers sequentially to check for shorts. Repeat this procedure for each pin. There should be only two sets of pins that are shorted; 1 to 51 and 50 to 100. If any other pair of pins are shorted, use a strong light and magnifying glass to locate the solder bridge or sliver causing the short and correct it. If there is no solder bridge or sliver, a component is shorted. Check the FDC-1 schematic (figure 2) to locate the probable component. Lift one lead of the suspected component and recheck between the two fingers that had a bad reading. If the bad reading is now correct, replace the component. If the reading is still bad, continue trouble-shooting until the faulty component is located and replaced. Ensure all components that had a lead lifted have had the lead reconnected.

WARNING

DO NOT INSTALL OR REMOVE ANY BOARD IN COMPUTER WITH POWER ON. DAMAGE TO BOARDS AND COMPUTER MAY RESULT.

Ensure computer is OFF. Plug the FDC-1 into the motherboard. Turn the computer power ON and check the outputs of the regulators on the FDC-1. If the regulators do not have the proper output voltages, turn the computer power OFF and replace the deffective regulators. Repeat until the regulator voltages are correct. If the unregulated +8 volt line power supply is outputting more than 11 volts this can cause the regulator voltage to fluxuate. If this happens, the power supply voltage will have to be decreased.

Now measure the voltages on the 40 pin socket: pin 1 should have -5.0V; if not, check the 79L05. Pin 40 should have +12.0V; if not, check the 78L12. Pin 21 should have +5.0V; if not, check the 7805. Pin 20 should be at ground.

If any of these measurements are not within 5% of what is listed here, find out why before proceding.

Install all IC's on the FDC-1 except the FD1771. Check parts placement drawing (see figure 1) for proper location and correct polarity of IC's.

When you've verified that the correct IC is in the correct socket, repeat the voltage checks. If the voltages check out o.k., proceed to the options selection.

OPTIONS SELECTION

CHECK OFF ✓

Addressing

I/O port number assignments for the floppy disk controller board are made in two places on the board. First is the four part block occupied by the FD1771. Its address is determined by the 6 dip switch positions numbered 2-7. (The 6 positions furthest from the voltage regulators). The most significant switch is the one numbered "2". It is the second switch from the left. For a "1" to be programmed into a position, set the switch such that it is "open". For a zero, position it closed. NOTE: When the switch is closed, the port address line is grounded, causing a "Ø". The base port number used in the sample software is Hex 80. The I/O ports are therefore numbered 80-83. The switch settings for this case are:

Switch	State
2	Open
All Others	Closed

For other port locations, are as follows:

Switch	2	3	4	5	6	7	Port
State	0	0	0	0	0	0	0
	0	0	0	0	0	1	4
✓ 10_H thru 13_H	0	0	0	0	1	0	8
	0	0	0	0	1	1	C
	0	0	0	1	0	0	10
	0	0	0	1	0	1	14
	⋮	⋮					⋮
	1	1	1	1	1	1	FC

Switch one is used as a write protect switch, so the most significant bit of the address selection is switch two.

NOTE: On some switches the designation will be "on" or "off". In these cases "on" means closed and "off" means open.

The second addressing task is that of the drive selection port. If only one drive is to be used, you may remove U9 and skip the rest of this step. The addressing for this port is done by means of jumpers which will be soldered into the jumper areas to the left of the dip switch marked A-P. Each letter represents a potential jumper. The address of the drive selection port will be uniquely defined in the following manner: Consider the group of 16 letters to be organized as 8 pairs of jumpers. The left side of the pair is a "1" and the right side is an "0". In each pair we will place a jumper over only one of the two letters. The most significant pair, or bit position in the address is on the right, or away from the regulators. If, for example, the port address for drive selection were to be 84H and as given in the example software, the jumpers, from right to left, would be covered in the following manner:

P - Blank
O - Jumper (1)
N - Jumper (0)
M - Blank
L - Jumper (0)
K - Blank
J - Jumper (0)
I - Blank
H - Jumper (0)
G - Blank
F - Blank
E - Jumper (1)
D - Jumper (0)
C - Blank
B - Jumper (0)
A - Blank

"1"	"Ø"	Address Bit
A	B	AØ
C	D	A1
E	F	A2
G	H	A3
I	J	A4
K	L	A5
M	N	A6
O	P	A7

2708 Address

The jumpers for the 2708 address work exactly the same way as the selection port address. These are the letters to the left of the dip switch, and zeros are to the right in each pair. The most significant bit of the address block is on the right end. For a base address of C000 as in the listing, the jumpers are placed from right to left as follows:

L - Blank
K - Jumper (1)
J - Blank
I - Jumper (1)
H - Jumper (0)
G - Blank
F - Jumper (0)
E - Blank
D - Jumper (0)
C - Blank
B - Jumper (0)
A - Blank

"1"	"Ø"	Address Bit
A	B	A1Ø
C	D	A11
E	F	A12
G	H	A13
I	J	A14
K	L	A15

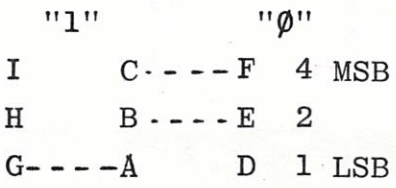
Wait State

The controller board can insert one wait state whenever addressed if so jumpered. This wait state affects both I/O port selection and the 2708. To insert a wait state, insert a jumper between W1 and W2 in the upper left hand corner of the board next to the crystal. This should only be necessary if your CPU runs faster than 2 MHz.

Restart Vector

The final jumpering task is to establish the interrupt vector to be used for "end of operation" by the controller. The board contains all necessary vectoring circuitry, you need decide only which vector the board is to occupy. The software is written to expect Level 1, and the jumpering for that vector is described in the following example.

The jumper area is located in the top right hand corner of the board, just above UIØ. The center column of pads is the operating point of each jumper, and needs to be jumpered either to the right or the left pad in each case. The most significant bit of the vector number (0-7) is the top pad. For a restart one, the jumpers will need to be arranged as follows:



NOTE: If using Wameco CPU-1, jumper W1-W2 on CPU-1 and remove U2 and U4.

The controller board assumes that all TTL interface signals between it and the drive are active low.

Data Separation

The controller board is configured for external data separation. If your drive does not have a data separator, the LSI controller has an internal separator, which may be implemented in the following manner:

1. Cut the trace on the back side of the board which connects to pin 25 of the FD1771.
2. Connect pin 18 of the drive connector to ground.
3. Bring composite read data to pin 19 of the drive connector.
4. Connect pin 25 to pin 21 : This ties $\overline{XTDS}$ high

Note on Data Separation:

The type of data separator used by most drive manufacturers on their more expensive drives will not work with the Western Digital 1771. For this reason we suggest you try to bring up your floppy disk system using the data separator internal to the 1771. Once this is operational, you can try the external separator.

The FDC-1 controller is configured for either step/direction type or step in/step out type of head positioning control. If your drive requires three phase signals, implement the following procedure:

Not Needed

1. Connect pin 18 of the FD1771 to ground (pin 20)
2. Install an additional 7406
3. Connect a wire from pin 17 of the FD1771 to pin 1 of the 7406 you just installed, and a wire from pin 2 of the 7406 to pin 22 of the drive connector. The connections to the input of your three phase driver are then as follows:

PH.1	Pin 29
PH.2	Pin 31
PH.3	Pin 22

Head Load Timing

The 5.1K resistor will compensate for head load delays of 50 milliseconds or shorter. If the delay in your drive is significantly longer or shorter, substitute a different resistor for the 5.1K ohm which has not yet been installed. Look up the right value in the following table:

10KΩ 80 ms

30 MS.	2.7 K ohm
40 MS.	4.3 K ohm
SA400 → 75 MS.	9.1 K ohm
100 MS.	13 K ohm

Now is the right time to find the place on the board marked R2 and solder in the proper resistor.

Multiple Drive Systems

Multiple drive selection is accomplished by means of the drive select output port. One line is assigned to each drive, with an active low select. On each drive in your system, jumper or position switches on the drive such that it will be selected by pulling a single input line low. Then make a connection between the input on the drive and the selected bit on the controller board. Note that these lines are not multiplexed, but that each drive will have a unique select line. All other signals to and from the drives are daisy chained, or wired in parallel. Be sure to remove the termination resistors from the input lines on all drives but the last one in your chain.

Suggestion for users of fewer than eight drives: Most drives, i.e., Remex, Shugart, etc., incorporate multiple line addressing. For example, the Remex drive has two drive address lines. This allows for up to four drives with the manipulation of only two lines.

Programming Your Controller

For a seek track command, the target track is loaded into the data register, and a seek track is written into the command word and the controller attempts to read a sector preamble. If a preamble is successfully located, the track number is read from the data contained there and verification takes place. If verification is not successful,

the controller continues to read sector preamble until it is successful or 3 index holes are detected. At that time an interrupt is generated and the seek error status bit in the status register is set. If the head load bit is set in the command byte, the head is loaded at the beginning of the seek rather than after seek is complete. The access rate is determined by the R1 and RØ bits in the command byte. See the table which follows for command byte format.

The track register is a read/write register which is updated with every seek track command. To save time in multiple drive systems, you may want to read the track register immediately before deselecting any drive and store that information away in system memory. Then when you reselect that drive you can reload the track register and the controller will be set to that drive without necessitating a "home" command. You can then imbed a home command in your error recovery routines for seek errors.

Read/Write Sector Commands

Before issuing a read or write sector command, you must have issued appropriate commands to seek the desired track and loaded the sector register with the desired sector number at which to begin reading or writing. After receipt of the command, the controller will read sector ID fields until a comparison with the track and sector registers is successful, or three index holes have been sensed, whichever comes first. Once ID verification is complete, the controller automatically wades through all the gap, address and data marks, and begins data transfer only when at the beginning of the actual data field. After transfer of data, in a read operation the CRC is verified, and the CRC status bit set accordingly. In a write operation, the CRC is computed and written in the appropriate place immediately following the data. All CRC information is computed and verified by the controller automatically, and the information contained therein transferred to the program via the status register.

Imbedded in the read and write sector commands are three flag bits. The enable flag when set to a "1" assumes the head not to be loaded and goes through the normal head load procedure with accompanying delay at the beginning of the operation. If this flag is set to a "0" the head is assumed loaded and no delay is entailed. The B flag allows for variable length sectors with IBM/Non-IBM format. This is explained in a table which follows. The M flag allows for multiple consecutive sector operations. If the M flag is set to a "1", the sector specified in the sector register is transferred, the sector register is incremented and the following sector is transferred. This loop continues until a force interrupt command terminates the operation or index is encountered, at which time the operation will terminate.

Two additional flag bits in the write sector command determine what the type of data address mark to write, which is useful for some data management techniques. This information is recoverable in the read sector operation in the status register.

There are two additional types of commands available to you. The read address command loads the head and reads the next encountered ID field from the disk. This information is transferred to the system as if

it were a six byte sector. The six bytes, in order of transfer are Track# , 00, Sector# , Sector length, CRC1, CRC2. CRC is verified, and the results set in the status register. In addition the sector is stored into the sector register.

The read and write track commands observe no format, do little automatically for you, and are really useful for non-IBM formats and formatting a clean disk. Details appear later.

The last type of command is the force interrupt command. This command can be loaded at any time. The result of such action is that an interrupt occurs, any current command being executed is terminated.

Command Summary

Command	Bits								
	7	6	5	4	3	2	1	0	
Restore	0	0	0	0	H	V	R1	R0	
Seek	0	0	0	1	H	V	R1	R0	
Step	0	0	1	U	H	V	R1	R0	
Step Out	0	1	1	U	H	V	R1	R0	
Read Sector	1	0	0	M	B	E	0	0	
Write Sector	1	0	1	M	B	E	A1	A0	
Read Address	1	1	0	0	0	1	0	0	
Read Track	1	1	1	0	0	1	0	$\overline{S}$	(not S)
Write Track	1	1	1	1	0	1	0	0	
Force Interrupt	1	1	0	1	I3	I2	I1	I0	

Flag Summary

H - Head Load Flag
- H=1 Load head at beginning
- H=0 Do not load head at beginning

V - Verify Flag
- V=1 Verify on last track
- V=0 Do not verify

R1 R0 - Track Access Rate

R1	R0	Rate (Full Size)	Rate (Mini)
0	0	6 MS	12 MS
0	1	6 MS	12 MS
1	0	10 MS	20 MS
1	1	20 MX	40 MS

U - Update Flag for Step Commands
- U=1 Update track register with step
- U=2 No update

M - Multiple Record Flag
- M=1 Transfer Multiple Records
- M=0 Single Record Transfer

B - Block Length

B=1 IBM format sectors, 128-1024 bytes/sector

B=0 Non-IBM format 16,4096 bytes/sector

The actual length of the sector is determined by the length field in the address field of the sector preamble determined during the format operation

A1 A0 - Data Address Mark

A1	A0	Mark	Meaning
0	0	FB	Data
0	1	FA	User Defined
1	0	F9	User Defined
1	1	F8	Deleted Data

S - Synchronize Flag

$\overline{S}$=0 Synchronize to address marks

$\overline{S}$=1 Ignore address marks

E - Enable Flag

E=1 Enable head load and wait for head load time out

E=0 Head assumed engaged and no delay

I0-I3 - Interrupt Conditions Flags

I0=1 Interrupt on not ready to ready transition

I1=1 Interrupt on ready to not ready transition

I2=1 Interrupt on each index pulse

I3=1 Immediate interrupt

If all bits are "0" no interrupt occurs, but the current command is terminated.

Status Register

Bit	Command Type: Seeks	Read Addr	Read Sctr	Read Track	Write Sctr	Write Track
S7	----------Not Ready----------					
S6	Write Protect	0	Record Type	0	Write Protect	Write Protect
S5	Head Engaged	0	Record Type	0	Write Fault	Write Fault
S4	Seek Error	No ID	No Record	0	No Record	0
S3	------CRC Error------			0	CRC Error	0
S2	Trk 0	---------- Lost Data ----------				
S1	Index	----------Data Request----------				
S0	----------Busy----------					

Read/Write Track Commands

Upon receipt of the read track command, the head is loaded and reading starts with the leading edge of the first encountered index pulse. A data request interrupt is generated for each byte on the track. No CRC checking is performed. Gaps are included in the input data stream. If bit 0 ($\overline{S}$) is a 0, the accumulation of bytes is synchronized to each address mark encountered. Upon completion of the command, a command, termination interrupt is generated.

As in the read track, a write track loads the head, and writing starts with the leading edge of the first encountered index pulse and continues until the next pulse. Address marks and CRC characters are written on the disk by specifying certain data bytes to be written, as shown in the table below.

Data (Hex)	Interpretation	Clock
F7	Write CRC Char	FF
F8	Data addr mark	C7
F9	Data addr mark	C7
FA	Data addr mark	C7
FB	Data addr mark	C7
FC	Index addr mark	D7
FD	Spare	
FE	ID addr mark	C7

The write track command will not execute if the WP switch (Dip Switch position 1) is closed. Instead the write protect status bit is set and a command termination interrupt is generated. In this way the switch can be used as an initialization write protect to prevent inadvertant initialization of diskettes.

Drive Connector

Pin	Line Name	Active Polarity	
Component Side:			
1	Drive 8 Select	Low	Output
2	Drive 7 Select	Low	Output
3	Drive 6 Select	Low	Output
4	Drive 5 Select	Low	Output
5 (16)	Drive 4 Select MOTOR ON	Low	Output
6 (14)	Drive 3 Select	Low	Output
7 (12)	Drive 2 Select	Low	Output
8 (10)	Drive 1 Select	Low	Output
9 (8)	Index	Low	Input
10	File Unsafe	Low	Input
11 (28)	Write Protect	Low	Input
12 (26)	Track 0	Low	Input
13	File Ready	Low	Input
14	Track > 43	Low	Output
15 (20)	Step Track	Low	Output
16 (18)	Direction	Low=In	Output
17	Load Head	Low	Output

18	Separated Clock	Low	Input
19 30	Data/Separated Data	Low	Input
20 32	Write Data	Low	Output
21 34	Write Gate	Low	Output
22	Spare		
23	Spare		
24	Step in	Low	Output
25	Step Out	Low	Output

Solder Side:

A-CC	Ground	Output/Input

Patching CP/M For The FDC-1 Controller

Reference:

It is assumed that you have read and understand to some degree the Manual "CP/M Alteration Guide", (C) Digital Research, which accompanies the CP/M diskette. This portion of the manual is to be considered an addendum to that manual to give you specific alteration instructions as they apply to the FDC-1.

Please note that regardless of memory size (it must be at least 16K to operate CP/M), you must bring up a 16K operating system before expanding it to fit memory capacity. Configuring the system to your memory will not be done until after a successful second level system generation.

First Level System Generation

The software in the on board 2708 provides you with the following routines as described in the alteration guide:

GETSYS
PUTSYS
HOME
SELDSK
SETTRK
SETSEC
SETDMA
READ
WRITE
The first half of the cold start loader

To do the system generation you will need the following:

CP/M Diskette
All CP/M Manuals
This Manual
2 Blank formatted Diskettes

Step 1:

Using the "GETSYS" program which starts at C01B Hex, read in the CP/M Diskette into memory. Remove the CP/M Diskette when this operation has stopped (approximately 8 seconds) and put it aside. We will not use it again for some time. Immediately insert one on the blank diskettes and write the CP/M system onto it using the "PUTSYS" program at C01E Hex. Do the same to the other blank diskette. Number these diskettes copies #1 and #2 for future reference.

The reason for using two diskettes is each step will be placed on alternating diskettes. If a problem is encountered during one of the steps and a diskette is "destroyed", it will only be necessary to back up one step rather than starting from scratch. NOTE: The following steps can be performed on the same diskette if only one diskette is available.

Step 2:

With the CP/M system still in memory, make the following patches. These will set all the disk routines to reflect your FDC-1 controller board.

Memory Location	Change To
3E19	03
3E1C	06 CO
3E1F	09 CO
3E22	OC CO
3E25	OF CO
3E28	12 CO
3E2B	15 CO

When all these patches have been made and verified, use "PUTSYS" to record this version on Diskette #1.

Step 3:

Three short routines must be written and entered into memory. Write a console input routine which conforms to the specifications of the alteration guide. Also a console output and a console status routine must be written. (See sample CONIN, CONOUT, CONST routines found in the appendix) These should be placed into memory starting at HEX address 3E2D. When these routines are in place, patch addresses 3EØ6 through 3EØE to reflect the locations of those routines. Address 3EØ6 is to contain a jump to the console status routine, 3EØ9 a jump to the console input routine and 3E0C a jump to the console output routine. Then use "PUTSYS" to record this version on diskette #2.

Step 4:

Write a routine to do some pre-initializations. This routine should initialize everything in your system that requires initialization, i.e., serial I/O port for your console, etc. CP/M depends on the following to be initialized at this time:

Memory Location	Contents	Function
0005	JMP 3106	BDOS Entry
0000	JMP WBOOTE	Warm Start Entry

We are not going to worry about the second of these for first level generation.

This initialization routine should finish with a JUMP to HEX address 2900. Place this routine in memory following your console routines saved in step 3 and place the address of this routine in Address 3EØ1. Use "PUTSYS" to record this level onto diskette #1.

At this point we will begin the testing of your CP/M operating system. Begin program execution at address 3E00 (the cold start entry point).

The disk drive should become active doing several sector reads. This should conclude with the printing of an "A>" prompt on the console. If this sequence did not occur, go back and check your patches. The most likely problem area is Step 3. If problems still persist, write a small program to perform the console I/O routines and determine if these are executing properly. If this works, insure that everything is being properly initialized.

Do not go on to Step 5 until this step is working.

Step 5:

This step will add a cold start boot loader to the diskette. The first half of this program is in the PROM on the controller board. This software reads the first sector of the first track into RAM starting at address zero. When this has been successfully completed (or at least partially successful), the program branches to address 0008 HEX to continue operation. The software beginning at address 0008 is contained on track 0 sector 1, and this routine reads in the rest of the system. The boot is done in two stages so that a single ROM resident program will work regardless of the memory size for which a particular diskette is configured. The boot program on the diskette itself contains all the size dependent software, on which we will now focus our attention.

Set the following program into memory starting at address 2880 HEX. Do not readdress anything.

```
                ;This program reads in the CP/M operating system into
                ;appropriate place in memory based on the memory size as
                ;defined in the first equate statement

0010            MEMSIZE   EQU     16              ;MEMORY CAPACITY IN K-BYTE
0008                      ORG     8
0008 DB80                 IN      COMM
000A E69D                 ANI     ERRBITS
000C C200C0               JNZ     ROMADDR         ;ERROR IN READING THIS SEL
000F 3E01       AGN       MUI     A,1
0011 D382                 OUT     SECTREG
0013 3EC3                 MUI     A,00C3H
0015 320800               STA     0008H
0018 215700               LXI     H,CHECK
001B 220900               SHLD    0009H
001E 310004               LXI     SP,0400H
0021 218028               LXI     H,CBASE
0024 CD 4700              CALL    READ
                READ1:
0027 3E01                 MVI     A,1
0029 D383                 OUT     DATA
002B 3E1C                 MVI     A,SKCOMM
002D D380                 OUT     COMM
002F CD3F00               CALL    BUSY
0032 3E01                 MVI     A,1
0034 D382                 OUT     SECTREG
```

```
0036 318035             LXI     H,CBASE+26*128
0039 CD4700             CALL    READ
003C C3003E             JMP     (MEMSIZE*1024)512   ;CBIOS ENTRY POINT
003F D880       BUSY    IN      COMM
0041 E681               ANI     0081H
0043 C23F00             JNZ     BUSY
0046 C9                 RET
0047 CD3F00     READ    CALL    BUSY
004A 3E9C               MVI     A,RDTRKCM
004C FB                 EI
004D D380               OUT     COMM
004F 76         LUPI    HLT
0050 DB83               IN      DATA
0052 77                 MOV     M,A
0053 23                 INX     H
0054 C34F00             JMP     LUP1
0057 33         CHECK   INX     SP
0058 33                 INX     SP
0059 DB82               IN      SECTREG
005B 2F                 CMA
005C D3FF               OUT     OFFH
005E 2F                 CMA
005F FE1B               CPI     27
0061 CB                 RZ
0062 DB81               IN      TRACK
0064 B7                 ORA     A
0065 C22700             JNZ     READ1
0068 C30F00             JMP     AGN
006B C9                 RET
0080 =          COMM    EQU     80H
0083 =          DATA    EQU     83H
009C =          RDTRKCM EQU     9CH
009D =          ERRBITS EQU     9DH
0082 =          SECTREG EQU     82H
001C =          SKCOMM  EQU     1CH
0081 =          TRACK   EQU     81H
2880 =          CBASE   EQU     (MEMSIZE*1024)-1780H
C000 =          ROMADDR EQU     0C000H
```

Now make these patches:

Location	Data
3E04	00 C0

It is also now important that the initialization routine written in Step 4 initialize the following:

Location	Data
0000	JMP C000

Now record this version using "PUTSYS" on diskette #2. Once done, you should be able to insert this diskette, execute the program beginning at C000 HEX (the cold start loader) and get the "A>" prompt.

If not successful, check your work. Chances are the problem is in the initialization routine or in getting the bootstrap onto Track 0 Sector 1.

Now is the time to run all the tests as outlined in the Alteration Guide. Do not write on the distribution diskette. Use the GETSYS/ PUTSYS combination to copy diskette #2 onto #1. If all of Digital Research's tests were successful, move on to second level generation.

Second Level System Generation

Step 1:

Bring up your first level system using the boot loader in your PROM. When CP/M prompts your console, remove your customized diskette and insert the distribution diskette you purchased from Digital Research. Type the following command line:

```
A>DDT DDT.COM
```

DDT will type the following:

```
DDT VERS 1.4
NEXT PC
1400 0100
-
```

Subtract the smaller HEX number (01001) from the larger HEX number (1400H), disregarding the two least significant digits. In this case, the result would be:

```
14-01=13H=19(DECIMAL)
```

It is the decimal answer that we are looking for. If during this and the following procedures you get a number with entries in the least significant digits other than zero, round the most significant digit up before doing your subtraction.

Now remove the distribution diskette and reinsert your custom version. Hit the reset button on your CPU. The system should do a cold start. Now type in the following:

```
A SAVE 19 DDT.COM
```

The 19 is the decimal result of the above subtraction. Examine the Directory. You should see that DDT.COM is now present on your diskette. Now proceed with three more files, as follows:

```
A>DDT ED.COM
A>DDT ASM.COM
A>DDT SYSGEN.COM
```

For each of them you will have to go through all the steps listed above to determine their size and then record them on your custom diskette.

Step 2:

You now have a working diskette with DDT, ASM, ED AND SYSGEN recorded from the distribution diskette. Now become familiar with the editor and assembler through reading the manuals and writing and assembling test programs. When you are comfortable with them, write a full blown