3.3.2 Standard Floppy Disk Drive Interface Cable Assembly

The mating ribbon cable connector to the Disk Drive connector (J1) must be a 5Ø-pin mass terminating receptacle (3M No. 3425-6ØØØ or equivalent). For interfacing with industry standard 8" floppy disk drives, this connector must be attached to a 5Ø wire ribbon cable with wire 1 mating to pin 1.

The standard floppy disk drives (Shugart SA-85Ø or equivalent) have a 5Ø-pin card edge connector for the signal interface, with a pin assignment shown in Table 3-6. One 5Ø-pin mass terminating card edge receptacle (3M No. 3415-ØØØ1 or equivalent) must be attached to the ribbon cable for each drive used (maximum of four). The cable length should be less than four feet, however, spacing between connectors is not critical. Figure 3-3 shows the typical connector placement.

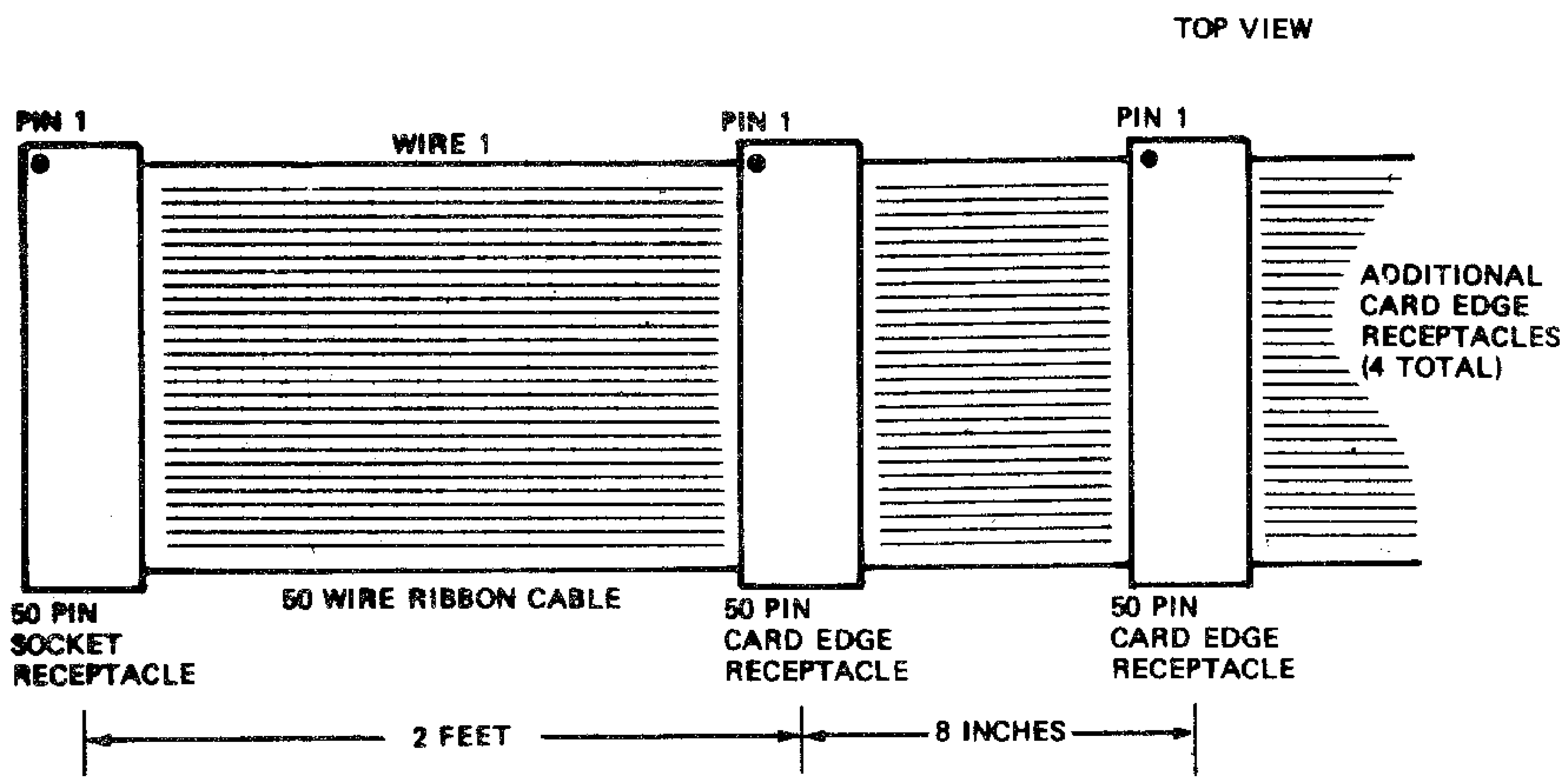

Figure 3-3. Standard Floppy Disk Drive Interface Cable

SECTION 4

FDC MODULE PRIMITIVE ROUTINES

4.1 PRIMITIVE ROUTINE DESCRIPTIONS

The FDC primitive routines simplify the operation of the FDC module by:

- Handling the detail protocol to interface with the FD 1793 Floppy Disk Controller device.
- Providing a set of closed subroutines that initialize and format a disk, position drive heads, write data on a disk, and read data from a disk.

These primitives allow additional higher level functions, such as a disk operating system or specialized data recording, to be easily implemented without being concerned with the detailed operation of the controller device. The optional AIM 65 DOS 1.Ø and AIM 65/4Ø DOS 1.Ø functions described in Section 5 and 6, respectively, use these primitive routines. If you are using one of these DOS versions, skip to the appropriate section. If you are designing a DOS or other specialized disk interface/file handling function using the primitive routines, the detailed description of these routines included in this section will be helpful.

The primitive routines perform the following major functions to support operation of up to four disk drives (single- or double-sided; single- or double-density:

Format a Disk
Read or Write a Sector
Seek or Verify Seek of a Track
Reset or Re-zero the Head
Read or Write Multiple Sectors
Read or Write a Track
Turn Motors On or Off
Select or De-select any Drive

These functions are implemented in 15 major primitive routines. Each of these routines can be classified into one of four general categories:

INITIALIZE - These initialize the FDC module or prepare a disk for use.

DRIVE - These control the selection and read/write head movement of the floppy disk drives.

READ - These support reading data from the floppy disk medium.

WRITE - These support writing data onto the floppy disk medium.

The primitive routines are identified along with their entry points in Table 4-1 (in alphabetical order of their labels). The labels and addresses of the I/O registers are listed in Table 4-2.

The primitive routines are described in Table 4-3. The entry and exit conditions are specified along with the processing performed.

For the primitive routines, the following conditions apply:

Drive number:	Ø to 3.
Side number:	Ø for side one, 1 for side two.
Density:	1 for single, Ø for double.
Track number:	Ø to 34 (5") or Ø to 76 (8").
Sector number:	1 to 16 (5") or 1 to 26 (8").

> NOTE
>
> There is no error checking done on the entry parameters of the primitive routines. This is the responsibility of the user in the disk operating software.

Table 4-1. Primitive Routine Addresses

Label	Address	Function
DESEL	$8CE2	Deselects drive
FORCOF	$8DB9	Cancel any command in progress
FORMAT	$89ØD	Formats a side (with interleaving)
INIT	$886C	Initializes variables and buffers
IRQHAN	$8BED	FDC Interrupt Handler
MOTOFF	$8CF7	Turns motor off
MOTON	$8C53	Sets motor on, selects drive, side and density
RDMSC	$8D8D	Reads multiple sectors within a track
RDSEC	$8D29	Reads a sector of data
RDTRK	$8CFB	Reads entire track (sector headers also)
SEEK	$8938	Seek with verify to a track
SELECT	$8CBC	Selects drive, side and density (only one at a time)
WRTMSC	$8DEA	Writes multiple sectors within a track
WRTRK	$8994	Writes an IBM formatted track
WRTSEC	$8D61	Writes a sector of data

Table 4-2. FDC Module I/O Register Addresses

Label	Address	Description
FCOMR	$8FØØ	Command register (Write only)
FSTAR	$8FØØ	Status register (Read only)
FCYLR	$8FØ1	Track register (Read or Write)
FSECR	$8FØ2	Sector register (Read or Write)
FDAR	$8FØ3	Data register (Read or Write)
DSTAR	$8FØ4	Drive Status register (Read only)
DCONR	$8FØ5	Drive Control register (Write only)
FSTOP	$8F15	Stop CPU (Write only)

Table 4-3. Primitive Routine Descriptions

Routine	Description	Category
DESEL	Deselects all disk drives. Any selected drive is deselected, with Drive Select No. 1 to No. 4 made inactive. The Motor On/Head Load line is left active. On entry, nothing is required. On exit, nothing is saved.	DRIVE
FORCOF	Stops execution of any FDC command in progress by forcing an interrupt command to the FDC device. Also clears the DMA command register (if DMAFLG is enabled) and disables the IRQ latch. This routine should always be called after a hardware reset. Typically, this will be part of the warm RESET processing. On entry, nothing is required. No drive needs to be active. On exit, any errors (type I, if no command was in progress) are indicated in the A register or STFLG.	INITIALIZE
FORMAT	Formats the disk in the selected drive. In single-density mode, FM encoding (IBM 3740) is used with 128 data bytes per sector. For double-density, MFM encoding (System 34) is used with 256 data bytes per sector. In both modes, every sector data field is filled with the character in PADB (default is $E5). For 5" mode, the disk is formatted with CYL5 tracks, SECT5 sectors per track, and sector interleaving as specified in TABLE5 area. For 8" mode, the disk is formatted with CYL8	INITIALIZE

Table 4-3. Primitive Routine Descriptions (Cont'd)

Routine	Description	Category
FORMAT (Cont'd)	tracks, SECTK8 sectors per track, and sector interleaving as specified in TABLE8 area. The read/write head is left on the inner-most track. Refer to Section 4.4.1 for default disk formats. On entry, the drive must be active (MOTON/SELECT). On exit, any errors (the error type depends on the FDC command that was executing when the error occurred) are indicated in the A register or STFLG.	INITIALIZE
INIT	Initializes the RM 65 system for the FDC module. Sets up all default values for the FDC variables (see Table 4-7). This must always precede any other FDC firmware routines. Typically, this will be part of the cold RESET processing. On entry, nothing is required. On exit, nothing is saved.	INITIALIZE
IRQHAN	Interrupt handling routine that services all RM 65 FDC interrupts. The system interrupt vector must be set up with IRQHAN before the FDC firmware routines are used. Typically, this would be set-up in the cold RESET processing. If an additional interrupt service routine is to follow, IRQOUT must point to it. Since this is an interrupt handler, it is <u>never called directly</u> by a user program, but is used by most of the firmware routines.	INITIALIZE

Table 4-3. Primitive Routine Descriptions (Cont'd)

Routine	Description	Category
MOTOFF	Turns off all drive motors (5")/unloads the read/write heads (8") and deselects all disk drives. Any selected drive is deselected, with Drive Select No. 1 to No. 4 made inactive. The drive motor flag MOFLG is reset and the Motor On/Head Load line is made inactive. On entry, nothing is required. On exit, nothing is saved.	DRIVE
MOTON	Turns on all drive motors(5")/loads the read/write heads (8") and selects the desired disk drive. Any selected drive is deselected then the desired Drive No. 1 to No. 4 is made active (uses SELECT), the drive motor flag MOFLG is set, and the Motor On/Head Load line is made active. The number of heads selected by the Single/Double Sided Drive jumper is set in NHEAD. On Entry, the drive (Ø-3) is in A, the side (Ø,1) is in Y, and the density (1 for single, Ø for double) is in X. On exit, nothing is saved.	DRIVE

Table 4-3. Primitive Routine Descriptions (Cont'd)

Routine	Description	Category
RDMSC	Reads multiple sectors of data from the present track of selected disk drive. Any number of sequential sectors on a given cylinder may be read. The read buffer (RDBUF) must be as large as the number of sectors being read. RDMSC will use the DMA module if DMAFLG is enabled; the Source Bank must correspond to the FDC module, while the Destination Bank is the RDBUF RAM buffer. On entry, the drive must be active (MOTON/SELECT) with the read/write head positioned on the proper cylinder (SEEK). The starting sector is in A and the last sector is in X. On exit, the data read is pointed to by RDBUF. Any errors (type II) will be indicated in the A register or STFLG.	READ
RDSEC	Reads the desired sector of data from the present track of the selected disk drive. The read buffer (RDBUF) must be as large as the sector size. RDSEC will use the DMA module if DMAFLG is enabled; the Source Bank must correspond to the FDC module, while the Destination Bank is the RDBUF RAM buffer. On entry, the drive must be active (MOTON/SELECT) with the read/write head positioned on the proper track (SEEK). The sector number is in A. On exit, the data read is pointed to by RDBUF. Any errors (type II) will be indicated in A register or STFLG.	READ

Table 4-3. Primitive Routines Description (Cont'd)

Routine	Description	Category
RDTRK	Reads the present track from the selected disk drive. This reads the entire track -- including data, clock, address marks and gaps--starting and ending at the index hole. The read buffer (RDBUF) must be large enough for the entire track. RDTRK will use the DMA module if DMAFLG is enabled; the Source Bank must correspond to the FDC module, while the Destination Bank is the RDBUF RAM buffer. On entry, the drive must be active (MOTON/SELECT) with the read/write head positioned on the proper cylinder (SEEK). On exit, everything read from the track is left in the buffer pointed to by RDBUF. Any errors (type III) will be indicated in the A register or STFLG.	READ
SEEK	Seeks to the specified track of the selected disk drive and verifies that the track was reached. The selected disk drive is restored (or re-zeroed) by a SEEK to track Ø. This recalibrates the disk drive read/write head back out to track ØØ. On entry, the drive must be active (MOTON/SELECT). The track number to seek is in the A register. On exit, any errors (type I) are indicated in the A register or STFLG.	DRIVE

Table 4-3. Primitive Routines Description (Cont'd)

Routine	Description	Category
SELECT	Selects the desired disk drive. The drive is made active and the read/write head is loaded (8" drives only). Only one drive may be active (by MOTON or SELECT) at a time, so other drives will first be deselected (DESEL) automatically. On entry, the drive (Ø-3) is in A, the side (Ø = one, 1 = two) is in Y, and the density (Ø = double, 1 = single) is in X. The drive motors must be on (MOTON). On exit, if the motors are not on, an error is indicated by an $8Ø in the A register or STFLG.	DRIVE
WRTMSC	Writes multiple sectors of data to the present track of the selected disk drive. Any number of sequential sectors on a given track may be written. The write buffer (WRBUF) must be filled with the data for all the sectors. WRTMSC will use the DMA module if DMAFLG is enabled. In this case, Source Bank must correspond to the WRBUF RAM buffer, while the Destination Bank is the FDC module. On entry, the drive must be active (MOTON/SELECT) with the read/write head positioned on the proper track (SEEK). The start sector number is in A and the last sector number is in X. The data written is pointed to by WRBUF. On exit, any errors (type II) are indicated in the A register or STFLG.	WRITE

Table 4-3. Primitive Routine Descriptions (Cont'd)

Routine	Description	Category
WRTRK	Writes an IBM compatible track onto the selected disk drive. This writes the entire track--including data, clock, address marks and gaps--starting and ending at the index hole. The track format used depends on the density and the disk size. In single-density mode, FM encoding (IBM 374Ø) is used with 128 data bytes per sector. For double-density, MFM encoding (System 34) is used with 256 data bytes per sector. The data fields are filled with PADB characters. For 5" disks, the number of sectors written is from SECT5 with interleaving from TABLE5 area. For 8" disks, the number of sectors written is from SECTK8 with interleaving from TABLE8. A disk can be formatted by performing WRTRK for every track. Refer to Section 4.4.1 for the default formats. WRTRK does not use the DMA module. On entry, the drive must be active (MOTON/SELECT) with the read/write head positioned on the proper track (SEEK). On exit, any errors (type III) are indicated in the A register or STFLG.	WRITE

Table 4-3. Primitive Routine Descriptions (Cont'd)

Routine	Description	Category
WRTSEC	Writes the desired sector of the data to the present track of the selected disk drive. The write buffer (WRBUF) must be filled with the data. WRTSEC will use the DMA module if DMAFLG is enabled. In this case, the Source Bank must correspond to the WRBUF RAM buffer, while the Destination Bank is the FDC module. On entry, the drive must be active (MOTON/SELECT) with the read/write head positioned on the proper cylinder (SEEK). The sector number is in the A register. The data written is pointed to by WRTBUF. On exit, any errors (type II) are indicated in the A register or STFLG.	WRITE

4.2 CALLING CONSIDERATIONS

The primitives are designed to work with any RM 65 system, with the memory constraints described in Section 3.1. This software can be interfaced to work with any system since it is I/O independent. A necessary vector to be set is the IRQ interrupt. The address of the routine IRQHAN must be put into the IRQ vector.

Due to the physical timing constraints of the floppy disk medium (i.e., the disk rotating at a fixed speed), the FDC routines that read or write data from the disk are time-critical and must not be interrupted (by either IRQ or NMI interrupts) except by the FDC device. If an interruption does occur, the FDC operation will be invalid and the error will be flagged in the status register.

To avoid such errors, calls to a primitive read or write routine should always be preceeded by a disabling of all interrupt sources except the FDC module (this cannot be an SEI instruction). Care must be taken with the non-maskable interrupt to ensure proper error handling. After the returning from the primitive routine, the interrupts can then be re-enabled.

There are two data buffers needed for input and output of data to/from the disk. The addresses of these buffers should be put into RDBUF and WRBUF. These buffers should be 128 bytes for single-density and 256 bytes for double-density.

When calling a primitive routine, the action of the CPU depends on the type of the FDC command being performed (every primitive routine typically corresponds to one FDC command). After the FDC device (and DMA controller if applicable) is set up with the appropriate control data, the FDC command is then issued to the FDC device. If a type I command is issued, the CPU will wait until an interrupt is received, indicating command completion. For the type II and III commands - in which data is moved between the disk and memory--the CPU controls the transfer of the data a byte at a time (the FDC device stops the

CPU with the Ready line), with an interrupt indicating completion of the command. When the DMA module is used with type II and III commands, the DMA controls the transfer of data while the CPU will wait for the command completion interrupt.

After receiving an interrupt indicating command completion, the status register is checked to report if the primitive routine was successful. A return is now done to the calling program, which should always examine the status in A or STFLG to ensure that a disk error did not occur before continuing (see Section 4.5).

4.3 DISK INITIALIZATION

4.3.1 Procedure

The standard initialization procedure for the FDC Module consists of calling INIT. This routine sets up the default variables by initializing the RAM locations used by the operating software and clears the FDC I/O logic. Before initialization, insure that all switches, jumpers, and headers are properly set. If any of the default values must be changed, it must be done after the initialization. Refer to Table 4-1 for the primitive routine addresses.

The typical sequence for initialization (which would be part of the cold RESET procedure) is as follows:

a) Load the IRQ interrupt request vector for the system interrupt handler with the address of IRQHAN (beginning of the FDC interrupt service routine).

b) Load the address of the IRQ service routine to follow the FDC interrupt handler into IRQOUT. If there are no additional routines, this IRQOUT can point to an RTI instruction.

c) Call the INIT subroutine.

d) Change any default variables to their new value (refer to Sections 4.3.2 and 4.6).

e) If the DMA module is being used, load the page number (high order byte of the assigned base address) into DMAADR and the source bank, destination bank, and enable information into DMAFLG.

The system is now prepared to operate with any of the primitive routines (refer to Section 4.4).

Whenever a reset occurs while an FDC operation is in progress, it is necessary to force a software interrupt to the FDC device. This is done by the FORCOF routine. Thus, the FORCOF routine will typically be included as part of the warm RESET processing.

4.3.2 Default Conditions

When the FDC module is initialized, the default parameters that are set up will be used for a majority of applications of the module. Any of these default values can be changed to meet the user's requirements. A list of default conditions that might be changed is given in Table 4-4. Refer to Table 4-7 for the variable addresses.

Table 4-4. Primary User-Alterable Variable Default Values

Variable	Default	Description
CYL5	35	Number of tracks on 5" disk
CYL8	77	Number of tracks on 8" disk
DMAADR	Ø	Pointer to page in which the DMA module resides
DMAFLG	Ø	DMA module enable and bank selection flags
IRQOUT	None	Pointer to additional interrupt service routines
PADB	$E5	Pad character used to fill data field
RDBUF	$1ØØØ	Read buffer pointer

Table 4-4. Primary User-Alterable Variable Default Values (Cont'd)

Variable	Default	Description
SECTK5	16	Number of sectors/track on 5" disk
SECTK8	26	Number of sectors/track on 8" disk
TABLE5	Section 4.4.1	Sector interleaving table for 5" disk
TABLE8	Section 4.4.1	Sector interleaving table for 8" disk
TUNDEL	1ØØ	Write tunnel delay of 1mS (needed by some drives)
WRTBUF	$2ØØØ	Write buffer pointer

4.4 DISK PRIMITIVE OPERATION

After the RM 65 FDC Module has been initialized (refer to Section 4.3) the module is prepared for all disk operations. The procedures for some of the more common primitive operations are described in this section.

The primitive routines all fall into four major categories-- Initialize, Drive, Read, and Write. These categories provide a functional grouping of commands. In building application programs with the primitives, these categories should be kept in mind. For example, in a typical disk read access:

Set up or recover system	-	INIT	(Initialize)
Select drive and track	-	MOTON and SEEK	(Drive)
Read the data	-	RDSEC	(Read)
Deselect the drive	-	MOTOFF	(Drive)

The system or disk must always be set up first (Initialize). After this, the drive selection and head positioning must take place (Drive). The disk access operation can now be performed (Read or Write), followed by drive deselection (Drive). The routines are not fully independent, requiring proper sequencing for meaningful results.

4.4.1 Formatting a Disk

Before any disk can be used for read/write data, the disk must be initialized or formatted. This consists of writing on all tracks of the disk: clock information; track, sector, and side identification (ID Field); data marker and data (Data Field). This disk initialization is compatable with the IBM 3740 format in single-density mode and IBM System 34 format in double-density mode.

In typical IBM compatible disks, sectors will be written sequentially on each track--1, 2, 3, ..., 15, 16 for 5" disks and 1, 2, 3, ..., 25, 26 for 8" disks. When doing sequential disk accesses (which is most of the time), all data movement and calculations preparing for the next sector must be done within the short intersector gap time, otherwise the sector is missed and an entire disk revolution must pass before the sector is available again. This makes the typical access time between sectors very long.

To shorten the typical access time between adjacent sectors, the disks are formatted with sequential sectors not adjacent, i.e., they are interleaved. Sequential sectors are separated by twelve sectors for 5" disks and 20 sectors for 8" disks, which gives at least 12 full sector time periods for data movement and calculations between sectors. This interleaving actually shortens the typical access time between sectors. For 5" disks, the physical sector format recorded on the disk is 1, 6, 11, 16, 5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12. For 8" disks, the sector order is 1, 6, 11, 16, 21, 26, 5, 10, 15, 20, 25, 4, 9, 14, 19, 24, 3, 8, 13, 18, 23, 2, 7, 12, 17, 22.

The sector interleaving information is kept as a table of data starting at TABLE5 for 5" disks and TABLE8 for 8" disks. Because this is RAM based, the interleaving order can be modified--either to support additional sectors per track or to optimize the inter-sector time period for a specific application.

NOTES

1. Disks formatted by the FDC module using the FORMAT routine are IBM compatible in recording technique (FM for single density, MFM for double density) and field structure, therefore they can be read/written on other IBM compatible equipment. This does not imply compatibility of the file structure used or the meaning of the data stored on the disk with other IBM compatible systems.
2. Diskettes created on the Rockwell System 65 cannot be used with the firmware routines because they are not IBM compatible.

There are four disk formats as summarized in Table 4-5.

Table 4-5. Disk Format Parameters

Disk Format	Bytes per Sector	Sectors per Track	Total No. Tracks
Single-density, 5" (FM)	128	16	35
Single-density, 8" (FM)	128	26	77
Double-density, 5" (MFM)	256	16	35
Double-density, 8" (MFM)	256	26	77

To format a disk, the following steps are taken:

a. Initialize the FDC module (refer to Section 4.3).

b. Call MOTON (for initial selection), or SELECT (on subsequent selections) to select the drive, side, and density of the disk to be formatted. Check the A register or STFLG and process any errors.

c. Call FORMAT. The disk in the selected drive will now be formatted for the selected density (refer to Table 4-5). This will take a few seconds. Upon return from FORMAT, check the A register for error conditions (such as disk write protected).

d. The read/write head is left at the inner-most track. Call SEEK to bring the head back to the desired track (typically, track Ø).

e. Call DESEL or MOTOFF to deselect the drive.

The disk is now prepared for read/write operations. All data fields on the disk are filled with the character in PADB (default is $E5).

4.4.2 Reading a Sector

To read a sector of data from a formatted disk, the following steps may be followed:

a. Initialize the FDC module (refer to Section 4.3).

b. Call MOTON (for initial selection) or SELECT (on subsequent selections) to select the drive, side, and density of the disk being read. Check the A register or STFLG and process any errors.

c. Call SEEK to move the read/write head to the desired track to be read. The head will remain at this track until another head movement command is performed. Check the A register for errors.

d. Call RDSEC to read the desired sector of data. The data left is pointed to by RDBUF. Check the A register for errors.

e. Call DESEL or MOTOFF to deselect the drive.

The read buffer now contains the sector of data read from the disk.

4.4.3 Writing a Sector

To write a sector of data to a formatted disk, the following steps may be followed:

a. Initialize the FDC module (refer to Section 4.3).

b. Call MOTON or SELECT to select the drive, side and density of the disk to be written. Check the A register or STFLG and process any errors.

c. Call SEEK to move the read/write head to the desired track. The head will remain at this track until another head movement command is performed. Check the A register for errors.

d. Call WRTSEC to write data to the desired sector. The data must be in the buffer pointed to by WRBUF. Check the A register for errors.

e. Call DESEL or MOTOFF to deselect the drive.

The write buffer data is now written on the disk.

4.5 PRIMITIVE ROUTINES

Whenever a disk operation is performed, error conditions can keep the operation from being completed. Most of the FDC primitive routines return with a status of the operation-- this status reports on incorrect or incomplete operation. After primitive disk functions, the status is left in the A register and STFLG. This status is typically the FDC device status register contents. The meaning of the status bits depends on the type of the routine. Table 4-6 lists the error definitions.

The Type I routines involve the selection of drives or movement of the read/write head. The Type I routines are: FORCOF and SEEK.

The Type II routines control reading and writing sectors of data to and from the disk. The Type II routines are: RDMSC, RDSEC, WRTMSC, and WRTSEC.

The Type III routines are for reading and writing of data and clock information, a track at a time, as well as reading the ID Field. The Type III routines are: FORMAT, RDTRK, and WRTRK.

All the above routines (types I, II, and III) can also return a motors not on (the SELECT routine does also) or no drive selected error ($80).

The remaining primitive routines do not return any error information: DESEL, INIT, MOTOFF, MOTON.

Dependent on the routine called, certain error checks should always be performed on return. Some of the error conditions (such as write protection) are checked within the routines, while others rely entirely on the application software for correction.

4.6 PRIMITIVE ROUTINES VARIABLES

The variables used by the primitive routines are located and identified in Table 4-7.

Table 4-6. Primitive Routine Error Definitions

Type	STFLG(A)	Description
I	Bit 1	Drive not ready (8" Drive only).
	Bit 6	Disk is write protected
	Bit 5	- - -
	Bit 4	Seek error
	Bit 3	CRC error in ID Field
	Bit 2	Track 00
	Bit 1	- - -
	Bit 0	FDC device is busy
II and III	Bit 7	Drive not ready (8" Drive only).
	Bit 6	Disk is write protected
	Bit 5	Read Record Type error (1 = Deleted Data Mark) Write error
	Bit 4	Record not found
	Bit 3	CRC error in ID Field
	Bit 2	Lost Data error
	Bit 1	- - -
	Bit 0	FDC device busy.

NOTE

An $80 indicates a Motors Not On or No Drives Selected error for Type I, II, and III routines.

Table 4-7. Primitive Routine Variables

Addr (Hex)	Label	No. Bytes	Init Value (Hex)	Definition
ØDB	PTR	2	-	Source/Dest. Buffer Pointer
ØDD	DMAPTR	2	-	Pointer to DMA Module Addr.
4AØ	IRQOUT	2	-	IRQ Exit Address*
4A2	FLAG	1	ØØ	Last Command Performed
4A3	CNTR	1	ØØ	Counter
4A4	TEMP	2	ØØ ØØ	Temporary Storage
4A6	TEMP2	1	ØØ	Temporary Storage
4A7	MSC	1	ØØ	DMA Most Significant Count
4A8	LSC	1	ØØ	DMA Least Significant Count
4A9	NCYL	1	ØØ	No. of Tracks
4AA	NSEC	1	ØØ	No. of Sectors
4AB	CURSEC	1	ØØ	Current Sector No.
4AC	LSECR	1	ØØ	Current Sector To Read Or Write
4AD	STFLG	1	ØØ	Status Flag
4AE	CURCMD	1	ØØ	Current Command Being Executed
4AF	WRTFLG	1	ØØ	Write Command Flag
4BØ	SELFLG	1	ØØ	Select Flag
4B1	MOFLG	1	ØØ	Motor Flag
4B2	USIDE	1	ØØ	Side No.
4B3	UCYL	1	ØØ	Track No.
4B4	UDRV	1	ØØ	Drive No.
4B5	CURCYL	4	ØØ -> ØØ	Current Track For Each Drive
4B9	DMAADR	1	ØØ	DMA MSB I/O Address*
4BA	DMAFLG	1	ØØ	DMA Flag*
4BB	FORFLG	1	ØØ	Write Track Flag
4BE	DBLCNT	1	FF	Double-Density Counter
4BF	SNGLCT	1	7F	Single-Density Counter

NOTES

*User-alterable.

Table 4-7. Primitive Routine Variables (Cont'd)

Addr (Hex)	Label	No. Bytes	Init Value (Hex)	Definition
4CØ	NBYTE	1	8Ø	Bytes Per Sector
4C1	NHEAD	1	Ø1	No. of Heads*
4C2	CYL5	1	23	Tracks Per 5" Disk*
4C3	CYL8	1	4D	Tracks Per 8" Disk*
4C4	SECTK5	1	1Ø	Sectors Per 5" Disk*
4C5	SECTK8	1	1A	Sectors Per 8" Disk*
4C6	HDEL	1	1E	Delay for Head Load*
4C7	SFLAG	1	Ø6	Side CMP & Data Mark (C&E)
4C8	TUNDEL	1	64	Delay for Non-Shugart Tunnel
4C9	RDBUF	2	ØØ 3E	FDC Source Buffer Vector*
4CB	WRBUF	2	ØØ 3F	FDC Destination Buffer Vector*
4CD	RATE	1	Ø7	Stepping Rate
4CF	RETRY	1	Ø5	Number Of Retries
4CF	PADB	1	E5	Format Byte Pattern*
4DØ	TABLE5	16	(Note 1)	5" Sector Interleaving Table*
4EØ	TABLE8	26	(Note 2)	8" Sector Interleaving Table*

NOTES

*User-alterable.

1. 6,7,8,9,1Ø,11,12,13,14,15,16,Ø,2,3,4,5 (decimal).
2. 6,7,8,9,1Ø,11,12,13,14,15,16,17,18,19,2Ø,21,22,23,24,25, 26,Ø,2,3,4,5 (decimal).