HOW MAGNETIC DISKS WORK

INTRODUCTION

Nearly every microcomputer includes some type of mass information storage system that enables it to store data or programs for an extended period of time. Unlike primary memory devices, which are fast and have a relatively low storage capacity, mass storage systems are usually slower and possess much larger storage potential. These systems represent an acceptable alternative to the IC RAM/ROM devices used on system and video cards. As with the ROM devices, these systems must be capable of holding information even when the computer is turned off. On the other hand, these systems are similar to RAM devices in that their information can be updated and changed often.

The most widely used mass storage systems have typically involved covering some medium with a magnetic coating.

These include flexible mylar disks (referred to as floppy disks), rigid, aluminum hard disks, and various widths of flexible mylar tape. The information to be stored on the medium is converted into electromagnetic pulses, which, in turn are used to create tiny positive and negative magnetized spots on the magnetic surface. To retrieve, or read, the information back from the surface, the storage system must detect the spots and decode them. The stored information can be changed at any time by re-magnetizing the surface with the new information.

MAGNETIC DISKS

Magnetic disks resemble phonograph records without grooves, and they fall into two general categories: high-speed hard disks and slower flexible disks. Data bits are recorded serially in concentric circles, called **tracks**, around the disk.

Because the tracks toward the outer edge of the disk are longer than the inner tracks, all tracks are divided into an equal number of equal-size data blocks, called **sectors**. Therefore, each block of data has an address that is the combination of its track number and its sector number. Each sector can be accessed for a read or write operation as quickly as any other sector on the disk, so disk memories are classified as **direct access memory**.

The tracks of the disk are numbered, beginning with 00, from the outer edge of the disk proceeding inward. Each side of the disk can hold 80 or more tracks, depending on the type of disk and the drive used. When disks are stacked, such as in a hard-disk drive, all of the tracks with the same number are referred to collectively as a **cylinder**. The number of sectors on each track runs between 8 and 50, also depending on the disk and drive type.

In some floppy-disk systems, the sectors are identified by holes along the outer or inner periphery of the disk. This is referred to as **hard sectoring** because hardware is used to identify the sectors by physically counting the holes as the disk rotates. In other flexible-and hard-disk systems, track and sector address information is contained in a track/sector identification code recorded on the disk. This method of address specification is known as **soft sectoring** because the sector information is written in software. Figure 1 depicts a typical soft-sectored track and sector arrangement. PC-compatible systems use soft-sectored disks.

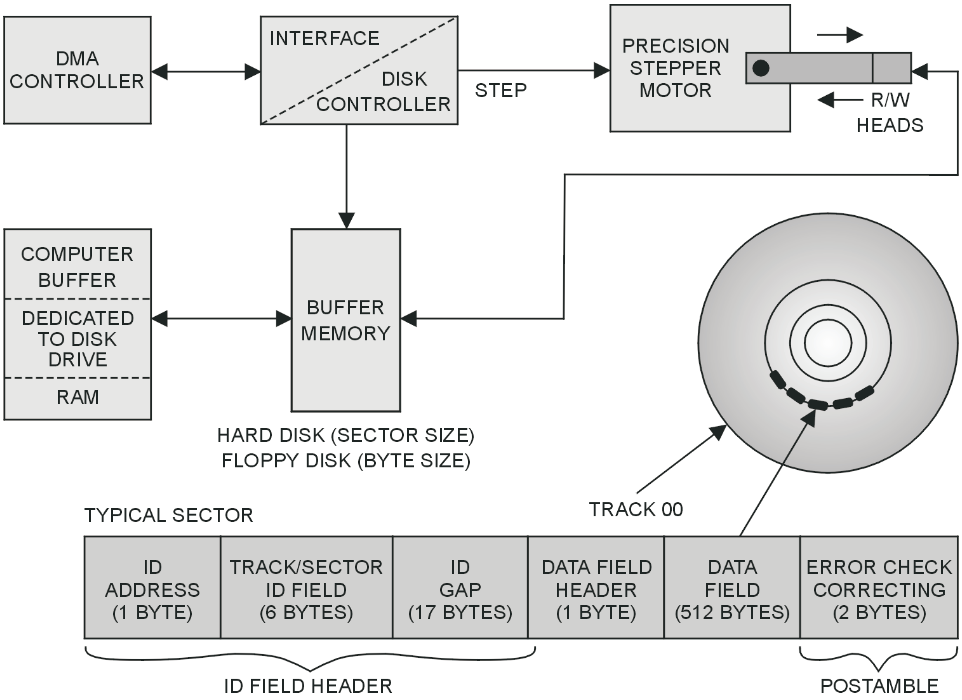


Figure 1: Disk Tracks and Sectors

Each sector is separated from the previous and following sectors by a small **gap** of unrecorded space. A typical sector contains 256 (2^8) or 512 (2^9) bytes, but on some systems this value may range as low as 128 (2^7) bytes or as high as 1024 (2^{10}) bytes per sector. The most common sector size in the IBM-compatible/DOS world is 512 bytes. Within these confines, the sector is segmented, beginning with an **ID field header**, which tells the controlling circuitry that an ID area containing the physical address information is approaching. A small **data field header** precedes the actual **data field**. The data field is followed by a **postamble** containing error-checking and correcting codes for the data recorded in the sector.

In their original conditions, hard disks and soft-sectored disks, as well as magnetic tapes, are blank. The system must prepare them to hold data. The system's disk-drive control circuitry accomplishes this by writing track/sector identification and gap locations on the disk, in a process known as **formatting**. This leads to some confusion when disk storage capacity is specified. Capacity may be stated for either formatted or un-formatted conditions. Obviously, the capacity of an un-formatted disk is greater because no gap or ID information has been added to the disk.

Reading and Writing on Magnetic Surfaces

Data is read from, or written to the disk one sector at a time. To perform a read or write operation, the address of the particular track and sector to be accesses is applied to a stepper motor, which moves a **read/write** (**R/W**) **head** over the desired track. As the desired sector passes beneath the R/W head, the data transfer occurs.

Information read from or to be written to the disk is usually held in a dedicated part of the computer's RAM memory. The system then accesses the data from this memory location at speeds compatible with the microprocessor.

The R/W head consists of a coil of wire wrapped around a soft iron core, as depicted in Figure 2. A small **air gap** in the core rides above the magnetic coating on the disk's surface. Data is written to the disk by pulsing the coil with a surge of current, which produces **magnetic lines of flux** in the soft iron core. At the air gap, the lines of flux dip down into the disk's magnetic coating due to its low reluctance (compared to air). This, in turn, causes the **magnetic domains** in the recording surface to align themselves in a direction dictated by the direction of current flow in the coil. The magnetic domains of the surface can assume one of three possible states, depending on the direction of current flow through the R/W head:

- Un-magnetized (randomly arranged domains)
- Domains magnetized in a positive direction
- Domains magnetized in a negative direction

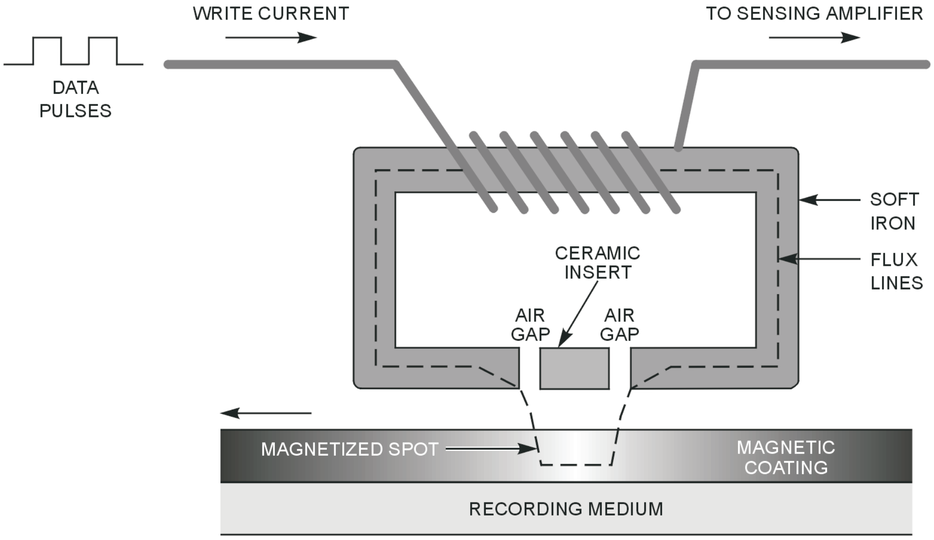


Figure 2: Typical R/W Head

Data is read from the disk in a reversal of this process. As the magnetized spots on the surface pass by the head, changes in magnetic polarity induce lines of flux into the iron core. This, in turn, induces a small voltage in the coil. This voltage is sensed and amplified to the proper digital logic levels by the drive's read circuitry. One significant difference exists between reading and writing on magnetic surfaces, however. **Writing** on the surface records successive zones of positive or negative flux, but **reading** can detect only flux changes (such as the boundaries between zones of differing polarity).

Data Encoding

It may seem logical to simply represent 1s and 0s with positive and negative magnetic spots, but the reality is that this brute force recording method is very ineffective and error prone. To compensate for these drawbacks, several encoding methods have been devised to represent 1s and 0s.

An early example of encoding was the **frequency modulation (FM)** method that used two recording frequencies to modulate the data. An advanced FM version called **modified FM** (or **MFM**) cuts out half the pulses and doubles the density of the data on the disk. This method is used to record double-density floppy disks in PC-compatible systems. Additional advancements in data storage are gained by using **group coded recording (GCR)** methods. This method uses selective bit patterns to encode blocks of data on the disk. The representative bit patterns are established so that they maximize the magnetic relationship between 0s and 1s to gain the best response from the R/W heads.

Head Construction

R/W heads are constructed in many different ways, depending on their intended application. R/W heads intended for use with disks are usually small and electrically simple compared to R/W heads used with tape drives. These types of heads normally have ferrite cores and a common winding for both reading and writing. Conversely, tape heads may have independent read and write cores, as well as multiple record/playback channels.

Quite often, the R/W head for data recording contains two independent cores, one designed for writing and the other designed for reading. The **READ gap** is placed downstream from the **WRITE gap** so that data being written to the surface can be read immediately and checked for errors. Some types of R/W heads contain as many as five air gaps, as shown in Figure 3. A gap may be only 30 to 100 micro-inches, but this is where all the work of recording and playback occurs.

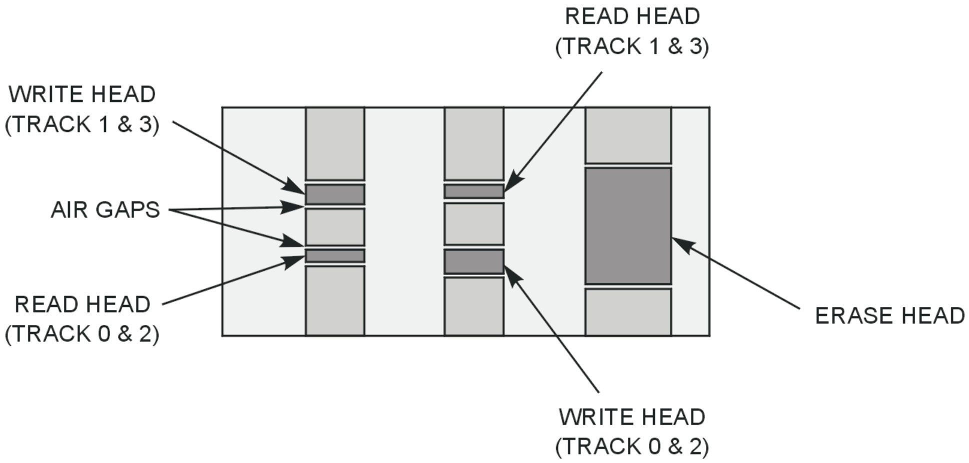


Figure 3: Multiple-Gap R/W Head

Multiple-gap R/W heads are generally used with magnetic tape operations, as opposed to disks, which use single-gap heads. An exception to this trend is the R/W head associated with floppy disks. Floppy-disk heads contain a R/W gap followed by a **trim-erase gap**. The erase gap is used to trim the fringes of the recorded data to improve tracking. This type of head is also referred to as a tunnel, or straddle-erase head.

The minimum distance between two successive flux reversals, called the **flux density**, is measured in **flux-changes-per-inch (fci)** or **flux-reversals-per-inch (frpi)**. These terms are a measurement of the number of bits of data that can be stored in a given area of media. Newer R/W head technologies have been developed to provide very high-density recording capabilities (15,000 fci).

Contact Versus Non-Contact Recording

Depending on the nature of the magnetic medium being read from or written to, the R/W head may either directly on the medium’s surface (**contact recording**) or float slightly above it on a thin cushion of air created by the moving surface (**non-contact recording**).

Hard disks, whether fixed or removable, must use flying heads, while flexible media (tapes and disks) generally use contact recording.

Hard disks must unavoidably use non-contact heads that fly over the medium. The extremely high speed of the medium, and the thin and fragile nature of its magnetic oxide coating, makes almost any contact between the R/W head and the disk surface a cause of considerable damage to both the head and the disk. Such contact is known as **head-to-disk interference (HDI)**, or simply a **crash**. Recent advancements—such as smaller and lighter R/W heads, and ever harder damage-resistant disk surfaces—have lowered the possibilities of damaging crashes somewhat. Because the medium is dimensionally stable and spins at a high rate of speed, the data density associated with hard disks is relatively high. Complex head architectures are not used with this medium.

Flexible media such as floppy disks and tape, expand and shrink with temperature and humidity variations. This causes the data tracks on the media to migrate in terms of track-location accuracy of the R/W head. To compensate for such shifting, the R/W heads ride directly on the media’s surface, the track density is kept low, and the heads are made more complex to create special zones in the track construction, which compensate for some of the misalignment due to shifting.

DISK-DRIVE OPERATIONS

The basic organization of both hard- and flexible-disk drives is similar in many respects. Both have drive spindles (which are actuated by precision synchronous motors) and one or more movable R/W heads that are positioned by a digital stepper motor, or voice coil. Both systems have intelligent control circuitry to position the R/W head and facilitate the transfer of information between the disk and the computer's memory. Figure 4 depicts the major components of the disk drive's electronics systems.

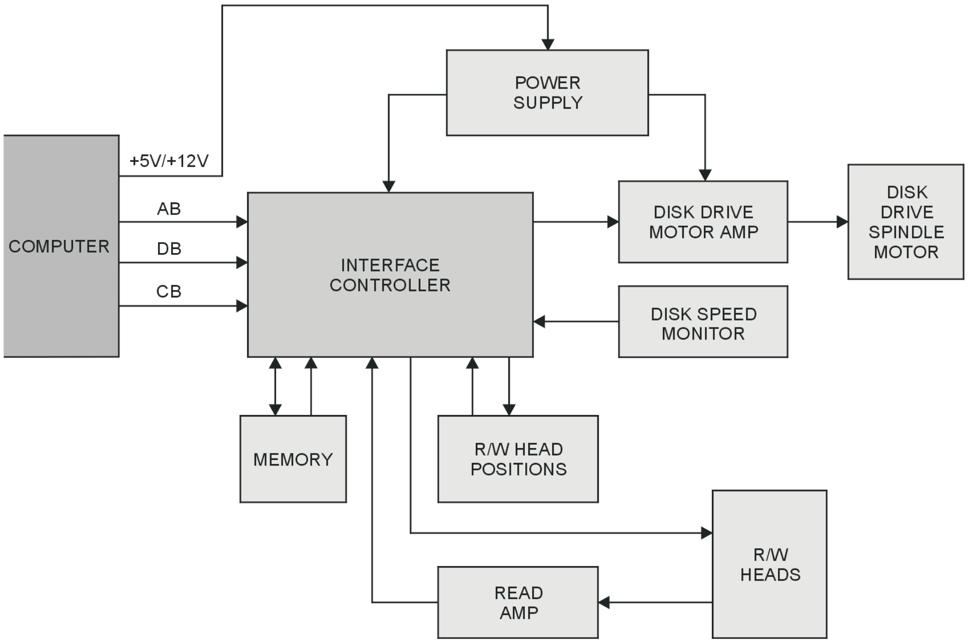


Figure 4: Disk Drive Components

The heart of the disk drive's circuitry is the **disk-drive controller**. The controller is responsible for providing the necessary interfacing between the disk drive and the computer's I/O channel.

This consists of decoding the computer's instructions to the disk drive and generating the control signals that the disk drive must have to carry out the instruction. The controller must also move back and forth between the parallel data format of the computer's bus and the serial format of the disk drive.

Furthermore, the controller must accurately position the R/W heads, direct the read and write operations, check and correct data received from the processor, generate error-correction codes for out-bound data, and mark the location of defective sectors on the disk, closing them to use. After all these responsibilities have been executed, the disk controller must also provide addressing for the tracks and sectors on the disk and control the transfer of information to and from the internal memory.

Due to the high data transfer rates associated with disk systems, a DMA controller is used to permit the disk drive to access the computer's primary memory without the intervention of the CPU. In this manner, the disk drive can exchange data with the computer's main memory while the microprocessor is busy at some chore that doesn't require the use of the address and data bus lines.

The extreme complexity of these responsibilities—and the speed at which they must be performed—usually dictates the presence of a specialized on-board **microcontroller** to control the operation of the drive. In this case, the has been designed expressly to control the operation of a floppy-disk drive. As a matter of fact, the first microprocessors were actually marketed as microcontrollers called **peripheral control units (PCUs)** for discrete computers. Only in their second generation were their full processing capabilities realized and brought to use as microprocessors.

In floppy-disk drives, and in hard drives designed around older interface standards, this controller is located on the drive's options adapter card. Drives that use newer interfacing methods have their controllers **embedded** directly on the disk drive unit.

With hard-disk drives, the presence of the onboard microcontroller is usually accompanied by onboard RAM and ROM, as well as any number of other support chips. The drive's RAM capabilities may be as simple as a single 8-bit buffer register or as complex as an array of RAM memory chips that can hold an entire track of information. This memory may be an integral part of the drive's intelligent controller, or it may consist of discrete memory devices.

Hard drives normally have read-only memories that contain an addition to the system BIOS program. This extension historically comes into play after the BIOS startup routine has checked for bootable files in floppy-disk drive A and has not found them. Other support devices in the drive might include clocks (timers), decoders, drive-motor controllers, read and write amplifiers, shift registers, counters, write-protect circuits (on floppy drives) and power supply circuits.

One piece of software is extremely important to the operation of both hard- and floppy-disk drives. This is the DOS **file management system** (**FMS**). Essentially, DOS keeps track of information stored on the disk by creating the file allocation table (FAT) on the disk. This table specifies where information is stored and what disk space remains available for new data. DOS matches the name assigned to a file with the blocks in the FAT and then refers to a map called a **directory**, which tells the operating system how to get to a specific piece of data.

In addition, the operating system provides disk utility programs to perform functions such as copying and editing files, keeping track of unusable (bad) sectors, and aiding in the formatting of new disks.

Groups of logically related sectors are typically linked into **clusters** (referred to as **allocation units** by DOS) on the disk. For example, if a file on a disk requires several sectors of storage space, DOS looks at the disk's FAT and tries to store the parts of the cluster in sectors that are logically sequential. This strategy takes into consideration the read and repositioning times involved in moving between sectors. This storage method improves the speed at which the total cluster of information can be accessed.

Initialization

A copy of the disk's directory and FAT are written into a dedicated portion of primary memory when the disk is booted to the system. To transfer a file from main memory to the disk (a write operation), the operating system sends the disk-drive controller a write command and initializes its control registers with parameters such as sector length and gap lengths. The system also specifies the track and sector number where writing will begin. It obtains this information by referring to the FAT and finding the address of the next available sector. Refer to Figure 5 for an illustration of this process.

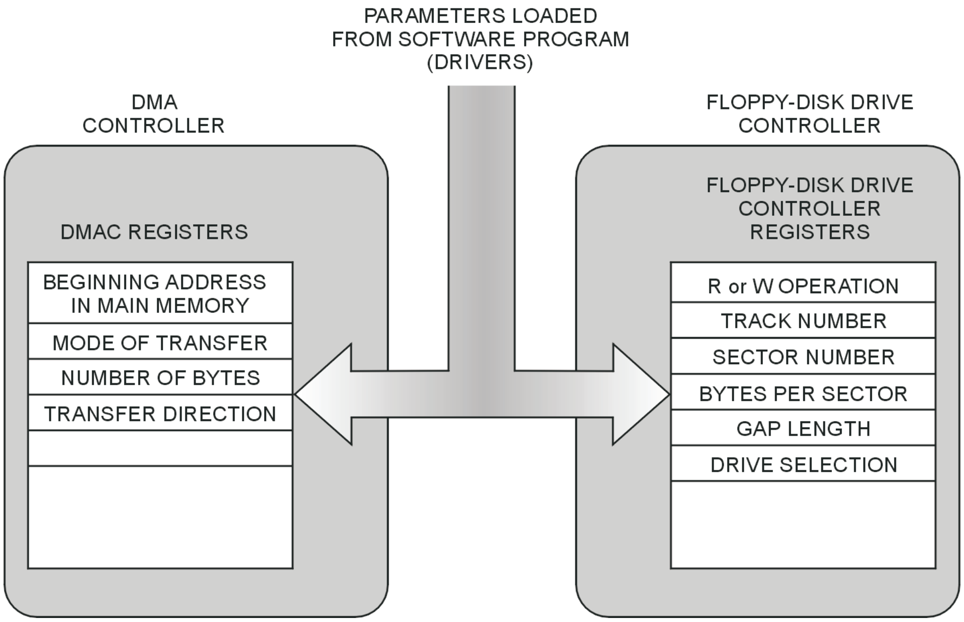


Figure 5: Initialization Process

The DOS software driver program plugs these parameters into the proper registers within the controller chip. The driver also sets up the DMA controller with information about where the starting address of the data to be written from main memory is, how long the block of data is (the number of bytes to be transferred), and in what mode the data transfer is to be performed (block or byte-by-byte). Once this has been accomplished, the drivers set up the DMA and disk-drive controllers for interaction and arms them to begin transferring the data.

Track Seek Operations

The controller enables the drive and produces a burst of step pulses to position the R/W head over the proper track. The controller accomplishes this by keeping a record of the current track location of the drive's R/W heads in one of its internal registers. The controller compares the contents of the current location register to the track number specified by the Write command and decides which direction the head must be moved. The controller then issues a direction signal to the drive on its **Direction** line and begins producing step pulses on its **Step** line. Each pulse on this line causes the drive unit to move the R/W heads one track over, in the direction specified by the direction signal. The value in the current track location register is also increased or decreased (depending on the direction of movement) by a factor of 1 for each step pulse. When the value of the present location register matches the track number specified for the Write operation, the step pulses cease and the R/W heads settle over the desired track. The positioning of the R/W heads is illustrated in Figure 6.

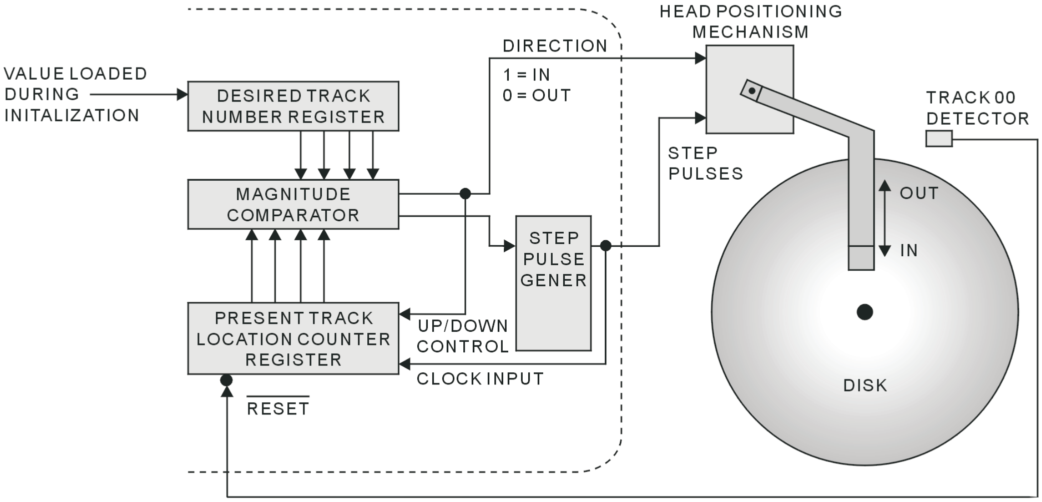


Figure 6: Track Seek Operation

Write Operations

When the head is over the track, the controller begins looking for the proper sector by reading the sector headers. The controller is actually waiting for a unique set of flux transitions to occur, which match the starting sector number stored in its registers. When the match is found, **write-splice** occurs (the controller changes from Read to Write operation) during the header gap, and data transfer and serialization begins.

The drive controller requests a byte of data (DREQ) from the DMA controller, which places the byte on the data bus and also sends a DACK signal to the disk controller. The disk controller obtains the byte (assuming byte-by-byte transfer) from the data bus, encodes it into the proper form, serializes it, and applies it to the Write channel of the R/W head, as illustrated in Figure 7.

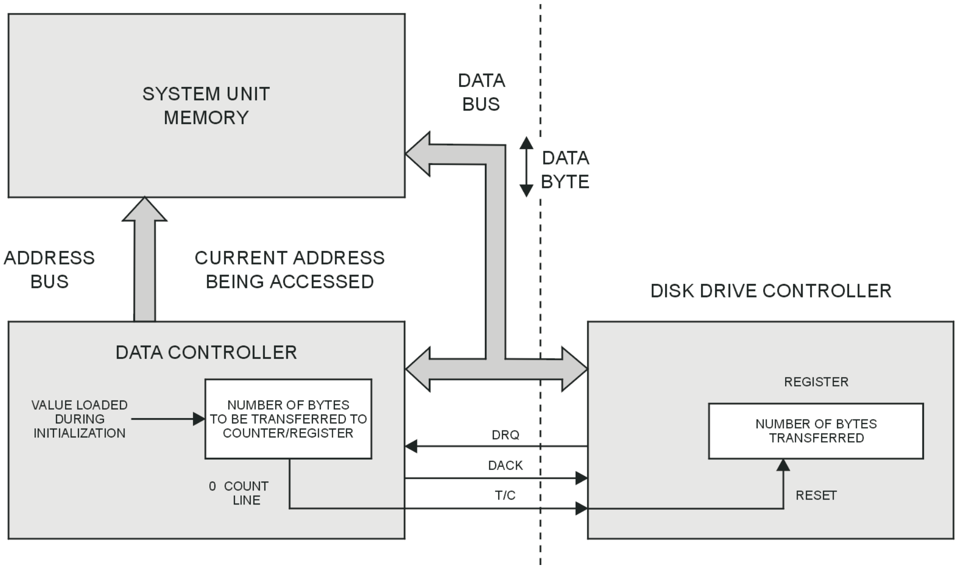


Figure 7: Data Transfers

The transfer continues byte-by-byte until the controller requests the last byte from the DMA controller. Counters in both the DMA controller and the disk drive controller count the number of bytes transferred. They do this by decreasing the initialized value of the number of bytes to be transferred by a factor of 1 each time a byte is transferred. When the counter in the DMA controller reaches its terminal count (0), it sends a **terminal count (T/C)** signal to the disk-drive controller. If the counter in the disk drive controller does not equal 0 at this time, an error flag is created in the disk drive controller, and the system must attempt to write the data again.

Data transfers in hard-disk drives must occur at a much quicker pace than those in floppy-disk drives. For this reason, hard-disk drives use onboard RAM buffers to enable transfers to occur in block form rather than byte by byte. The buffer is small, usually only large enough to hold one sector of data, although some manufacturers use larger memories. When the controller begins serialization, the data is obtained from the buffer rather than through the DMA channel.

In addition to writing the data, the controller must generate the sector postamble. At the end of the data field, the controller generates the postamble containing the error-detection and correction codes. The sector's **preambles** are written on the disk when it is formatted.

If the data from the computer requires more than one sector to be written, as it usually does, these logically related sectors may not be located sequentially on the disk. When data is transferred a block at a time, some time is required to process each sector of data. To give the drive time to process the information, logically sequential sectors are **interleaved** (separated) by a fixed number of other sectors. This way, the motion of the disk moves the second sector into position to be written (or read) while the drive is processing the previous sector of information. A common **interleaving factor** is 8 sectors between logically related sectors on a floppy drive, a factor of 3 on older hard-disk systems, and a factor of 1 on newer systems.

Read Operations

When a READ operation is performed, the operating system goes to the directory, either in main memory or on the directory track on the disk, to obtain the starting track/sector address of the data to be read. This address is loaded into the disk controller, and a Track Seek operation is performed. The R/W head is stepped to the desired track, and the proper head is selected by the disk-drive controller. After a few milliseconds delay (to allow the R/W head to settle over the track), the operating system gives the disk controller the command to read the desired sector. The controller begins reading the sector ID headers, looking for the assigned sector.

When the sector is identified, the preamble is read and **bit-sync** is established. As the preamble is read, the controller synchronizes the data separator with the incoming bit stream from the disk drive. At the beginning of the data field, a **data start marker** initiates **byte-sync**, which coordinates the first bit of the first data byte with the controller's internal circuitry. At this point, the controller begins dividing the incoming bit stream into 8-bit words for transmission to the system unit.

After byte-sync has been established, the drive begins reading the data through the controller. The controller decodes the bit stream from its coded form (such as FM, MFM, and GCR) and shifts it into an onboard shift register in 8-bit chunks. Hard-disk drives deliver the bytes to the onboard buffer memory; most floppies set up a DMA request and transfer the data bytes unbuffered into the computer's main memory. The transfer may continue over multiple sectors or tracks until an **end-of-file marker** is encountered, indicating that the entire file has been transferred.

FLOPPY-DISK DRIVES

The discussion of general disk drive operations applies to both hard and floppy drives alike. However, the physical construction and operation of the drives are quite different. The FDD is an exposed unit, with an opening in the front to allow the floppy disk to be inserted and removed. In addition, the R/W heads are open to the atmosphere and ride directly on the surface of the disk. Older 5-1/4-inch units used a locking handle to secure the disk in the drive. A spring-loaded assembly ejected the disk from the drive when the handle was rotated. Newer 3-1/2-inch units have ejection buttons. Table 1 provides a comparison of the two different floppy drive types using the disk formats possible for each.

Table 1: Comparisons of Floppy-Disk Drive Standards

DIAMETER	DENSITY	CAPACITY	TRACK	SECTORS
3 1/2"	DD	720kB	80	9
	HD	1.44MB	80	18
5 1/4"	DD	360kB	40	9
	HD	1.2MB	80	15

Data moves back and forth between the system's RAM memory and the floppy disk surface. Along the way, it passes from the system RAM to the **floppy-disk controller (FDC)** through the floppy drive signal cable and into the floppy drive's **analog control board**. The analog control board converts the data into signals that can be applied to the drive's read/write heads which, in turn, produce the magnetic spots on the disk surface.

In the original PCs and XTs, the FDC circuitry was located on the FDD controller card. In AT-compatible systems, it migrated onto a multi I/O card along with parallel, serial, game, and HDD control ports. With Pentium-based systems, all this circuitry has been integrated into the system board.

The circuitry on the floppy drive unit is usually distributed between two printed circuit boards: the analog control board and the drive's **spindle motor control board**.

FDD Controller

In older PCs, the FDD control function was performed by a **765 FDC controller** chip and a discrete **digital control port register**. In newer units, the floppy disk control function is provided by the FDC portion of the **integrated I/O ASIC**. This chip can be located on an MI/O card or on the system board. To remain PC-compatible, the FDC registers and signal definitions used must remain identical to those of the 765 FDC and digital control port register. In any case, the FDD controller provides a programmable interface between the system unit and the floppy-disk drive unit. Figure 8 depicts a block diagram of the floppy-disk drive control circuitry.

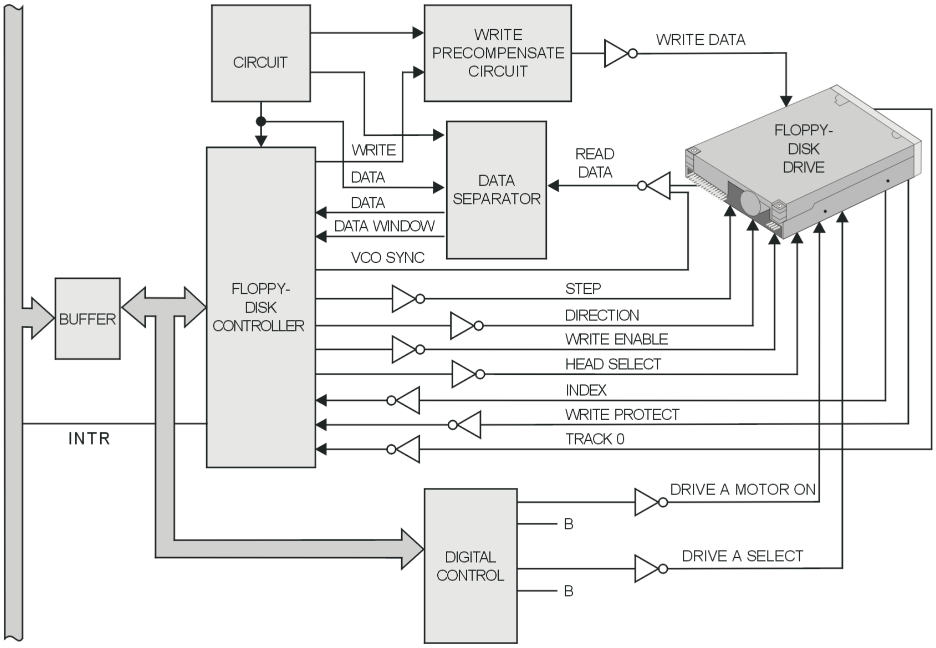


Figure 8: Floppy-Disk Drive Adapter

Under direction of the DOS system, the FDC divides the 3-1/2-inch floppy disk into 80 tracks per side, with 9 or 18 512-byte sectors per side. This provides the system with 737,280 (720 kB) or 1,474,560 (1.44 MB) total bytes of storage on each disk. Table 2 lists the operating specifications for a typical 3-1/2-inch floppy-disk drive unit.

Table 2: FDD Specifications

DRIVE UNIT PART	DSSD	DSHD
Tracks	80	80
Heads	2	2
Sectors per Track	9	18
Bytes per Sector	512	512
Formatted Capacity	720kB	1.44MB
Unformatted Capacity	1MB	2MB
Rotational Speed (RPM)	300	30
Recording Density (bits/inch)	8717	17,432
Tracks per Inch	135	135
Transfer Rate Unformatted (Kbps)	250	500

The 765-compatible FDC can carry out at least 15 different commands—such as **Read Sector**, **Write Sector**, **Read Track**, **Seek** and **Format Track—**under the direction of the disk operating system software. The FDC supports both double-density and high-density (MFM) recording formats and performs all the data-decoding functions for the drive. It has the capability to control two drives (drives A and B) simultaneously, with any mixture of sizes and densities. In addition, the FDC performs all data synchronization and error-checking functions to ensure reliable storage and recall of data.

The ASIC provides the additional circuitry required to complete the entire FDD controller/interface section. The additional circuitry consists of the address recognition circuitry, a clock signal generator to produce master and write clock signals, and a read data separator circuit to remove the data bits from the other information written in a sector. The FDC governs the operation of the data separator.

The ASIC also supplies interface signals that enable it to be connected to microprocessor systems with or without DMA capabilities. However, in most systems, the FDC operates in conjunction with the system's DMA controller and is assigned to the **DRQ2** and **DACK2** lines. In operation, the FDC presents a DRQ2 request for every byte of data to be transferred. In addition, the disk-drive controller is assigned to the **IRQ6** line. The FDC generates an interrupt signal each time it receives a Read, Write, or Format command from the system unit. An interrupt is also generated when the controller receives a READY signal from one of the disk drive units. Figure 9 depicts a block function diagram of the FDC.

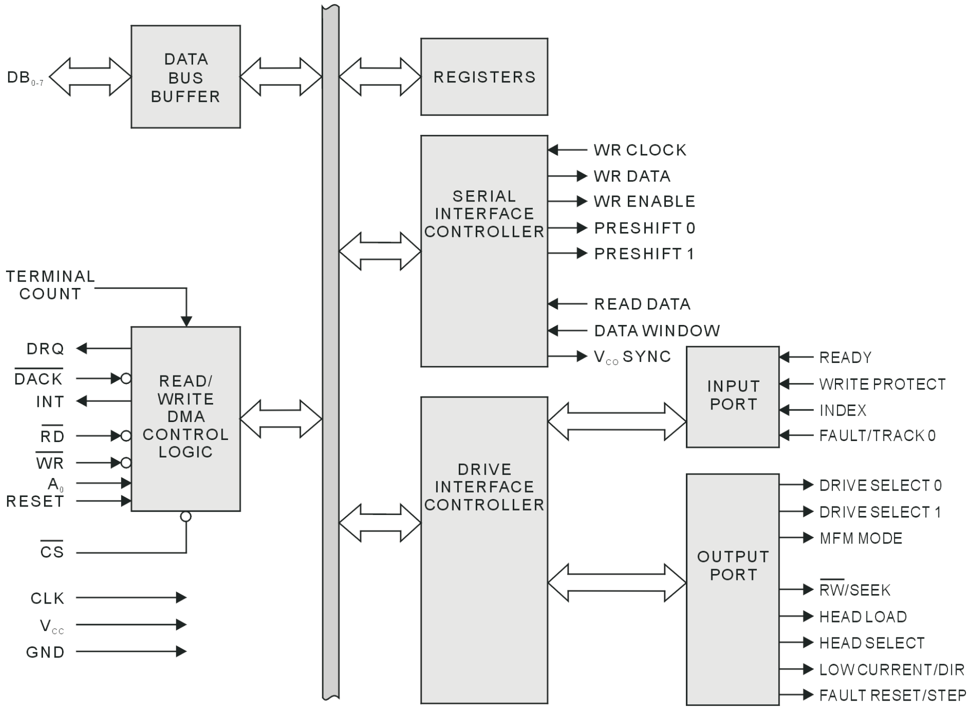


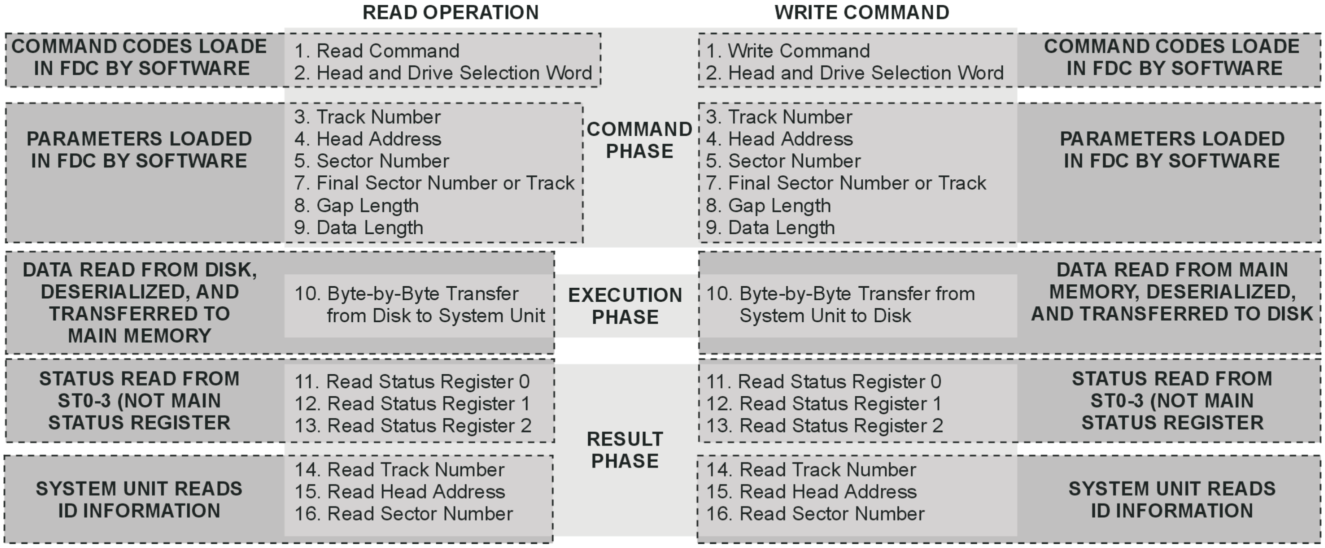
Figure 9: FDC Block Diagram

Of particular interest is the FDC's internal register set. This register set contains two registers that can be addressed by the system unit's microprocessor: an 8-bit **Main Status register**, and a stack of 8-bit registers called the **Data register**. Only one of these data registers can be accessed at a time. Data bytes are written into or read out of the data register to program the controller or obtain results after the completion of an operation. The Main Status register contains status information about the drive and can be accessed by the system at any time.

The operation of the FDC occurs in a three-phase cycle: a **command phase**, an **execution phase**, and a **result phase**.

During the command phase, the system unit initiates an action by writing a multi-byte instruction and all related data into the FDC's data register. At the completion of the command phase, the FDC enters the execution phase, in which the FDC generates the timing and control signals necessary for the drive to carry out the specified command. After the instruction has been carried out, the FDC enters the result phase, in which the disk drive's status information is placed in the Main Status register. Figure 10 shows the sequence in which information is moved into the FDC's Data register during Read Sector and Write Sector commands. Although the sequence is identical for these two commands, other instructions occur in different sequences and with other parameters.

Figure 10: FDC R/W Operations



READ OPERATION			WRITE COMMAND	
COMMAND CODES LOADE IN FDC BY SOFTWARE	1. Read Command 2. Head and Drive Selection Word	COMMAND PHASE	1. Write Command 2. Head and Drive Selection Word	COMMAND CODES LOADE IN FDC BY SOFTWARE
PARAMETERS LOADED IN FDC BY SOFTWARE	3. Track Number 4. Head Address 5. Sector Number 7. Final Sector Number or Track 8. Gap Length 9. Data Length		3. Track Number 4. Head Address 5. Sector Number 7. Final Sector Number or Track 8. Gap Length 9. Data Length	PARAMETERS LOADED IN FDC BY SOFTWARE
DATA READ FROM DISK, DESERIALIZED, AND TRANSFERRED TO MAIN MEMORY	10. Byte-by-Byte Transfer from Disk to System Unit	EXECUTION PHASE	10. Byte-by-Byte Transfer from System Unit to Disk	DATA READ FROM MAIN MEMORY, DESERIALIZED, AND TRANSFERRED TO DISK
STATUS READ FROM ST0-3 (NOT MAIN STATUS REGISTER	11. Read Status Register 0 12. Read Status Register 1 13. Read Status Register 2	RESULT PHASE	11. Read Status Register 0 12. Read Status Register 1 13. Read Status Register 2	STATUS READ FROM ST0-3 (NOT MAIN STATUS REGISTER
SYSTEM UNIT READS ID INFORMATION	14. Read Track Number 15. Read Head Address 16. Read Sector Number		14. Read Track Number 15. Read Head Address 16. Read Sector Number	SYSTEM UNIT READS ID INFORMATION

Digital Control Port Register

Operating in parallel with the FDC is the Digital Control Port register. This register is used to control the selection of drive units A and B; the drive motors; and the adapter's FDC Reset, DMA Request (DRQ2), and Interrupt Request (IRQ6) functions. This is a write-only register, selected when the system unit applies an address of 3F2 to the adapter, with the IOW line asserted. The various bits of the Control Port register are used for drive/motor selection, DRQ and IRQ enabling circuitry, and the FDC Reset bit. This register is depicted in Figure 11.

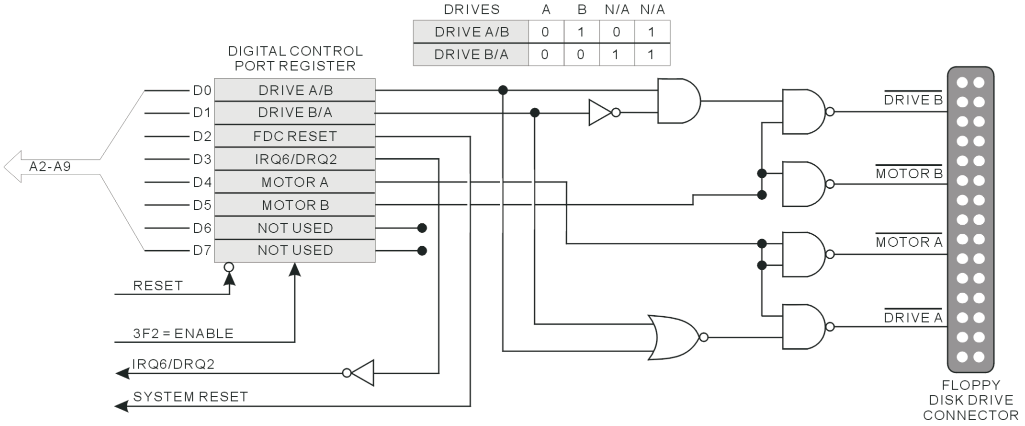


Figure 11: Digital Control Port Register

Read Circuitry

Data sent to the FDC from the drive is actually a combination of two signals: the data and the clocking information used to record it. This concept is easier to envision for FM-coded data, where each bit time is marked by the presence of a clock pulse. But MFM-encoded data possesses clocking information embedded in the serial bit stream. To properly read the data, it must be broken into two parts: the clocking information and the data bits. This is the function of the **Data Separator** circuitry.

The Data Separator must synchronize the rate of the serial bit stream coming from the drive unit and the clocking of the FDC's internal circuitry (bit-sync). After synchronization is achieved, the separator creates a **Data Window** timing signal, used by the FDC to differentiate between the clocking information and the data bits in the bit stream. The FDC uses this window to reconstruct the data bits from the bit stream.

In addition to recovering the data bits from the bit stream, the FDC must break the data bits into 8-bit words for transfer to the system unit (byte-sync). This requires that the system unit's DMA controller service the FDC approximately every 15 microseconds for MFM-recorded data. If this servicing does not occur, the FDC will detect that a Data Register overrun error has occurred and sets the **Overrun Error flag** in its Main Status register. This action terminates the Data Read command activities.

The disk drive adapter's Read Channel components are shown in Figure 12. The MFM-encoded data stream from the disk drive passes through pin 30 of the disk-drive connector. Next, the data stream is applied to the Data Separator, where it is combined with the FDC's clock signal. From the two inputs, the Data Separator produces two outputs: the serial bit stream and the **Read Data Window (RDW)** signal. The RDW signal causes the FDC to sample the **Read Data (RDD)** line. During the window, the FDC loads the data bit on its Read Data input into the internal Data Register.

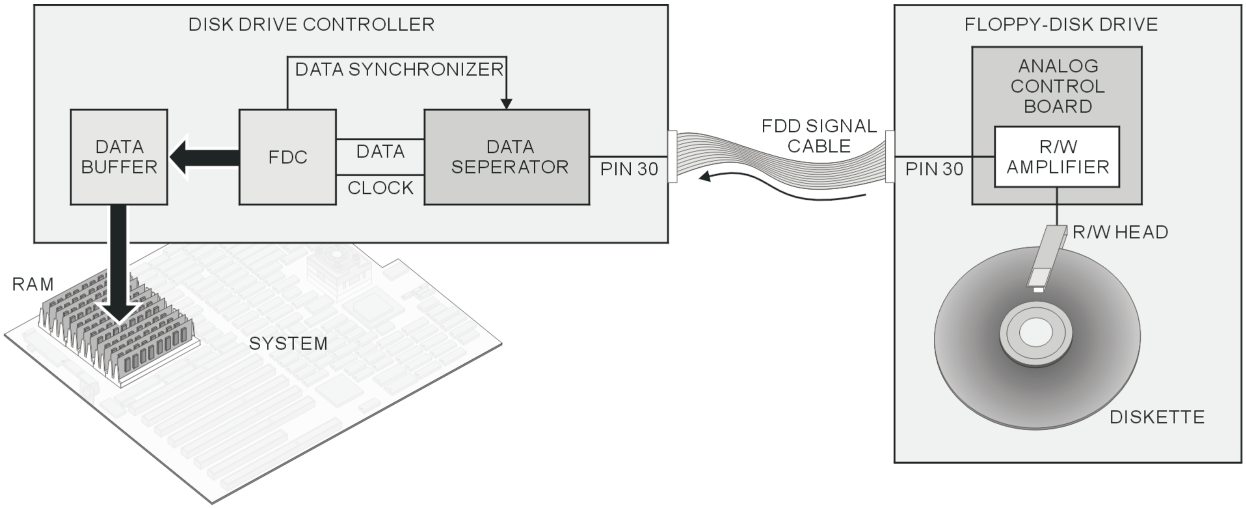


Figure 12: FDC Read Circuitry

Write Circuitry

The disk drive adapter's Write circuitry is depicted in Figure 13. When the system unit places data in the FDC to be written on the disk, the FDC serializes the data, codes the data in the designated form (FM or MFM), and adds error-detection and correction bits (CRCs) to the serial data stream. The data moves in encoded, serial format from the FDC's **Write Data (WD)** output through a Write Pre-compensate circuit to the adapter's FDD connector at pin 22. The cable carries the WD and **Write Enable (WE)** signals to the disk drive unit.

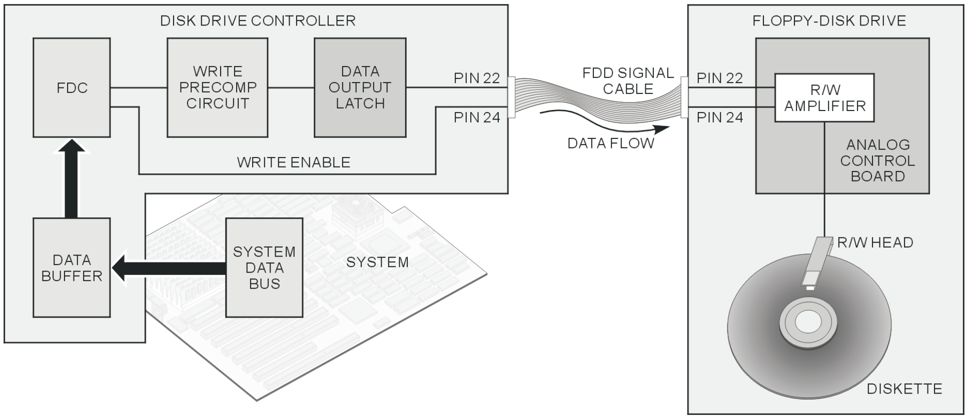


Figure 13:
FDC Write Circuitry

The **Write Pre-compensate circuit** acts as a time delay to correctly position the data bits for proper read-back. This is necessary because a certain amount of data shift occurs during the read-back process. This shifting is predictable, so the FDC is programmed to precompensate the data stream before it's written.

The precompensation circuitry is required only for MFM-encoded data. MFM-encoded data does not contain a clocking pulse in its bit cell, so it cannot tolerate large amounts of data-shifting when the data is read back from the disk.

Floppy-Drive Cables

A single ribbon cable is used to connect the system's floppy drive(s) to the disk-drive controller card. Generally, the cable has two **edge connectors** and two 34-pin, two-row BERG headers along its length.

The edge connectors enable the cable to be connected directly to the printed-circuit board of a 5-1/4-inch FDD; the BERG connectors are used for 3-1/2-inch drives. The other end of the cable terminates in a 34-pin, two-row BERG header. A small colored stripe normally runs along one edge of the cable, as illustrated in Figure 14. This is the **Pin #1 indicator stripe** that marks the side of the cable, which should be aligned with the #1 pin of the FDD adapter's connector and the disk drive's signal connector. The location of this pin is marked on the drive's printed-circuit board.

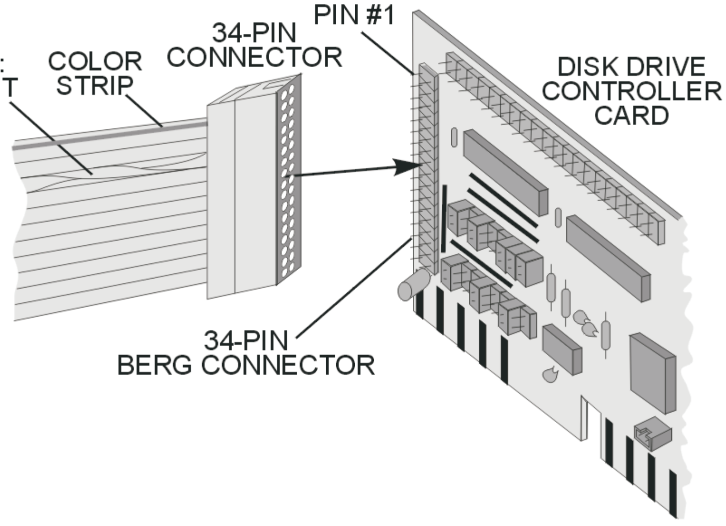
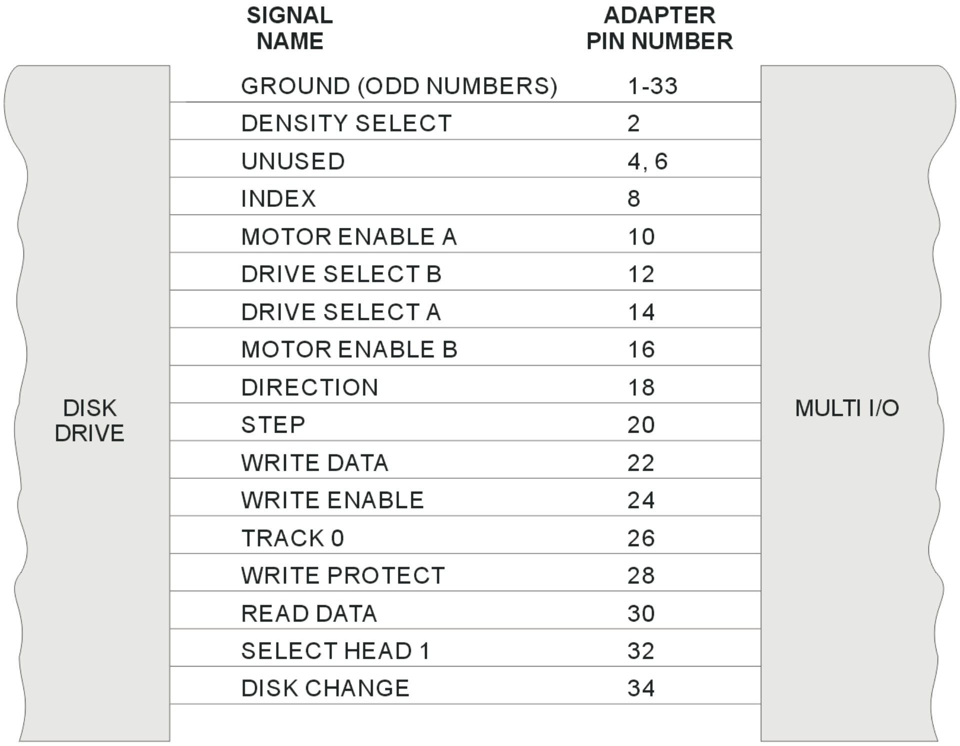


Figure 14:
FDD Signal Cable

The system assigns the **drive A** designation to the drive connected to the edge connector at the end of the cable. A floppy drive connected to the edge connector in the middle of the cable is designated as the B drive. A small twist of wires between the A and B connectors reroutes key lines that distinguish between the two drives.

The 34-pin interface connection enables the FDC to control two separate floppy-disk drive units. Figure 15 depicts the connections between the disk-drive adapter and the disk drive(s). The direction of signal flow between the drive(s) and adapter is indicated by the arrows.



SIGNAL NAME	ADAPTER PIN NUMBER
GROUND (ODD NUMBERS)	1-33
DENSITY SELECT	2
UNUSED	4, 6
INDEX	8
MOTOR ENABLE A	10
DRIVE SELECT B	12
DRIVE SELECT A	14
MOTOR ENABLE B	16
DIRECTION	18
STEP	20
WRITE DATA	22
WRITE ENABLE	24
TRACK 0	26
WRITE PROTECT	28
READ DATA	30
SELECT HEAD 1	32
DISK CHANGE	34

Figure 15:
FDD Cable Signal Definitions

All the adapter's signal lines are TTL-compatible. Furthermore, the functions of the lines are summarized as follows:

WRITE DATA (pin 22): For each high to low logic transition on this line, the disk drive stores a flux change on the disk. This action depends on the Write Enable line being activated.

READ DATA (pin 30): The selected disk drive places a pulse on this line for each flux change on the disk that passes under the selected R/W head.

WRITE ENABLE (pin 24): This line disables the drive's Write circuitry unless it is active.

INDEX (pin 8): The selected drive applies a pulse to this line each time the index hole on the disk passes the index sensor (1 pulse/revolution or approximately 300 pulses/min).

TRACK 0 (pin 26): When the R/W heads of the selected drive are positioned over track 0, the drive's track 00 sensor activates this line.

WRITE PROTECT (pin 28): The presence of a write protected disk in the selected drive activates this line.

DRIVE A (pin 14): This line enables the drive unit attached as drive A when active.

MOTOR A (pin 10): This line starts the drive A spindle motor when activated and stops it when the line returns to a high logic level.

DRIVE B and MOTOR B (Pins 12 and 16): These lines are identical to the drive A and Motor A lines, except they control the drive connected as drive B.

STEP (Pin 20): During a Seek operation, the FDC issues pulses on this line. The selected drive must move the R/W heads one track per pulse. The direction of movement is in accordance with the condition of the Direction line.

DIRECTION (Pin 18): When this line is high, the selected disk drive moves the R/W heads one track away from the center of the disk for each pulse on the Step line. When low, the drive moves the heads one track toward the center for each pulse on the Step line.

SLCT HEAD 1 (Pin 32): When this line is high, the upper R/W head (head 0) of the selected drive is activated. When low, the lower head (head 1) is activated.

DENSITY SELECT (Pin 2): This line sets the write current level used for double density or high density disks. The controller outputs a low when double density disks are detected, or a high when high density disks are detected.