$14.95

STREAMING

ARCHIVE
CORPORATION

STREAMING

Archive Corporation
3540 Cadillac Ave., Costa Mesa, CA 92626

STREAMING

International Standard Book Number: ISBN 0-9608810-0-X

Library of Congress Catalog Number: 82-072125.

TABLE OF CONTENTS

Mass-Storage Principles 1

Winchester Technology 2

Flexible Disks 3

Tape Storage 4

Streaming ¼-inch Cartridge 5

Streaming Software 6

System Considerations 7

Disk-Tape Storage 8

Streaming-Cartridge Drive 9

CHAPTER 1

Mass-Storage Principles

This book is about "streaming" magnetic tape—a fast, efficient, economical mass-storage medium.

More specifically, it is about the newer forms of streaming ¼-inch tape cartridges which represent by far the most economical media for storing data in capacity ranges that match the requirements of small-to-medium computer systems.

Such installations represent, in turn, the fastest growing segments of the computer industry: small business systems, distributed processing systems, local networks, personal computers, process control and instrumentation—plus word processing, business graphics, electronic mail, and all of the other applications that fall under the general heading of "office automation."

The handbook is directed to both the engineer/programmers who are responsible for the development of such systems and the specifier/users who must evaluate their worth in terms of function and cost. Sidewinder streaming-cartridge drives are featured in the final chapter, partly because they are products of the Archive Corporation, but more importantly, because they represent the most advanced examples of this new type of mass storage.

Streaming tape storage is, however, only one element in the system. It must be evaluated in conjunction with other system

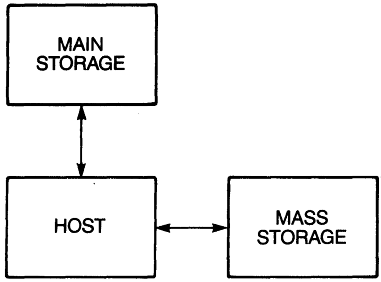


Figure 1-1 Programmed-I/O mass storage.

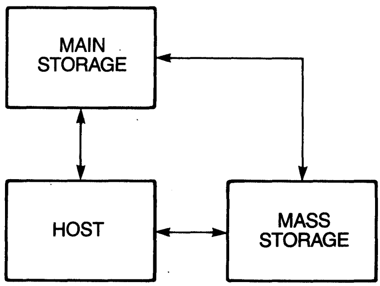


Figure 1-2 Mass storage with direct memory access (DMA).

components and in comparison with other mass-storage techniques. Moreover, as noted below, the current trend is toward a combined disk-tape mass-storage architecture, with streaming tape acting as archival "backup" for a faster but more expensive (per megabyte) random access Winchester disk.

MORE THAN BACKUP

"Primary" Winchester storage does not mean that streaming-tape storage should be evaluated only on the basis of its usefulness as a "secondary" medium for archival purposes. As systems designers learn to take advantage of the combined disk-tape architecture, major shifts in emphasis are likely to occur.

For example, file-management tasks can be routinely implemented off-line between disk and tape as the data is processed,

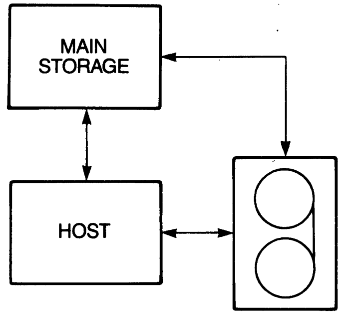


Figure 1-3 Serially recorded magnetic-tape storage.

as contrasted to an end-of-the-day "dump" of all the data stored on disk. Streaming tape is already established as a medium for data exchange and program distribution. It is not inconceivable that in many applications, users will eventually view streaming tape as the primary storage, with disks as simply fast-access caches for data and program segments currently being processed.

CHANGING ROLES

From the start, digital computers have required some form of data storage as an adjunct to their relatively sparse main-storage facilities. Punched cards and paper tape, for example, did double duty as mass storage and input/output media (Figure 1-1).

Mass storage, as presently conceived, did not become a reality, however, until faster-transfer, higher-capacity media became available and a direct link was established between the computer's main memory and the mass-storage device (Figure 1-2).

Magnetic tape storage was the principal storage medium during this transitional period (Figure 1-3). Its serial format, high capacity, and fast transfer rates lent themselves to the large-scale batch processing that dominated early computer applications. But within a few years, several interrelated developments relegated magnetic tape to a secondary role.

Rotating memories became the standard mass-storage device (Figure 1-4). Online capacities were comparatively limited, but

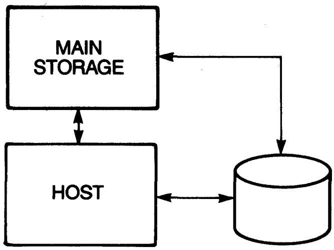


Figure 1-4 Random-access disk-based storage.

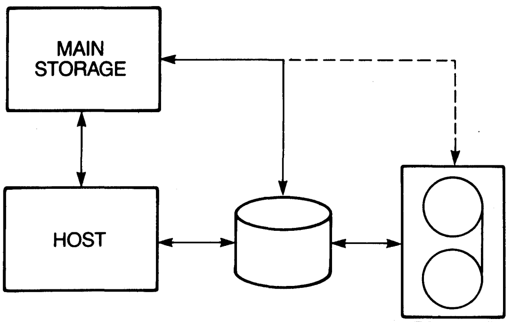


Figure 1-5 Disk storage with magnetic-tape backup.

more than adequate for the new generation of small computers. The combination of low-cost computer power and random-access disk storage led, in turn, to the proliferation of transactional-type applications which now characterize computer processing.

RESURGENT TAPE

Magnetic tape is now returning to the forefront, but in a new and different role. Replaced as an extension of main memory, it serves instead, as an extension of disk memory (Figure 1-5).

The trend back to tape started with Winchester disks. Winchester technology offers a number of advantages: device reliability, data integrity, faster transfer rates, and a broader range of capacities. But early Winchester disks were sealed in a controlled-environment enclosure, leading to the need for a removal-medium backup to protect against equipment failure, to offload data no longer needed on the disk, and to provide a low-cost, convenient vehicle for data exchange and program distribution.

Winchester drives with removable disks are now available, but disk storage will always be bulky and expensive in comparison with tape (see Chapter 4). For the foreseeable future, magnetic tape will continue to be an important mass-storage medium.

MAGNETIC RECORDING

The common denominator between disk and tape is the fact that both are based on magnetic-recording principles.

Cost and performance depend in both cases on compacting the largest amount of information onto the available recording surface. The smallest "bit" of information (not necessarily a digital bit) is a magnetic flux reversal between two magnetic states. The goal is to create a maximum number of flux reversals per inch as the medium moves under the "write" head and at the same time, take optimum advantage of the storage medium by tracing a maximum number of flux-reversal "tracks." The limiting factor is the ability to reliably retrieve the information by retracing the tracks with a "read" head—an electromagnetic challenge in the case of flux reversals along a track, an electromechanical challenge in the case of track-to-track alignment.

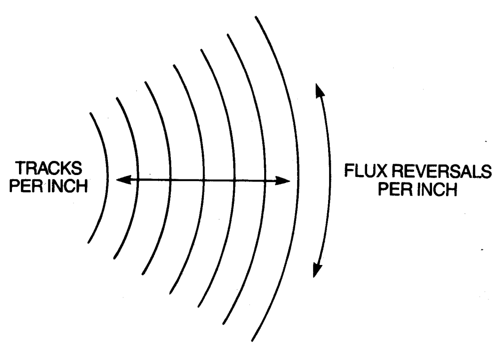


Figure 1-6 Bit density, disk storage.

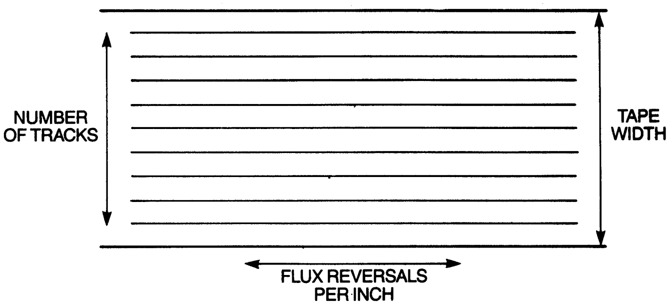


Figure 1-7 Bit density, tape storage.

These differences help to explain the relative packing densities of disk and tape devices (Figures 1-6 and 1-7). As noted below, flux-reversal density along a track is partly a function of the distance between the heads and the magnetic material. Flexible, slower moving magnetic tape can be pressed directly against the heads, resulting in reliable densities beyond 10,000 frpi (flux reversals per inch). To avoid damaging "crashes," rigid-disk heads must ride above the surface, reducing disk flux-reversal densities.

But the rigidity of the disk makes it simpler to relocate and follow a recorded track, resulting in disk track densities of hundreds per inch—compared to only 42 tracks on a typical 1-inch instrumentation tape or up to 17 tracks on a ¼-inch cartridge tape.

FLUX REVERSALS

Magnetic recording is based on the fact that a variety of materials, typically iron compounds or alloys, can be "permanently" magnetized by an external north-south magnetic field, and that this magnetism can be neutralized or reversed by subsequent exposure to another magnetic field.

There are a number of different theories on the nature of magnetism, but the following description should be sufficient for this discussion.

The orbit of an individual electron around an atomic nucleus

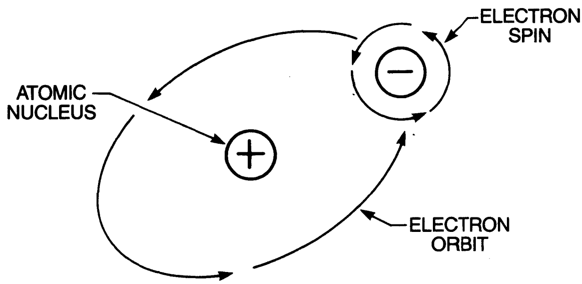


Figure 1-8 Electron spin as a source of magnetic force.

(Figure 1-8) is equivalent to current flow through a single winding of a coil. The resulting magnetic field is effectively neutralized, however, by the orbits of other electrons around the same nucleus.

An electron also "spins," creating another magnetic field. This, too, is neutralized by other electrons spinning in the opposite direction. But in a few cases, notably the iron atom, there is a net difference between the spins, making the atom a permanent magnet.

Individual fields of adjoining atoms tend to align themselves, forming microscopic "domains" which, in a non-magnetized material, neutralize each other (Figure 1-9). When placed in an external magnetic field, domains that are naturally aligned with the field tend to expand at the expense of other domains. There is also an "elastic" tilt of domains that are at an angle with the field. The result, when the external field is removed, is a net residual magnetism.

Figure 1-10 shows how current reversals through the windings of a write head can create alternating north-south magnets in the coated surface of a rotating disk or moving tape. A subsequent passage under a read head (Figure 1-11) will induce a voltage pulse each time the gap passes over a flux reversal and "cuts through" the magnetic flux entering or leaving the north-north or south-south magnetic poles.

Figure 1-12 indicates the spread of this coupling effect as a function of the width of the read-head gap and, equally important, the distance from the gap. The latter is, in turn, a function of both the head-tape separation and the depth of the flux reversal within the magnetic coating.

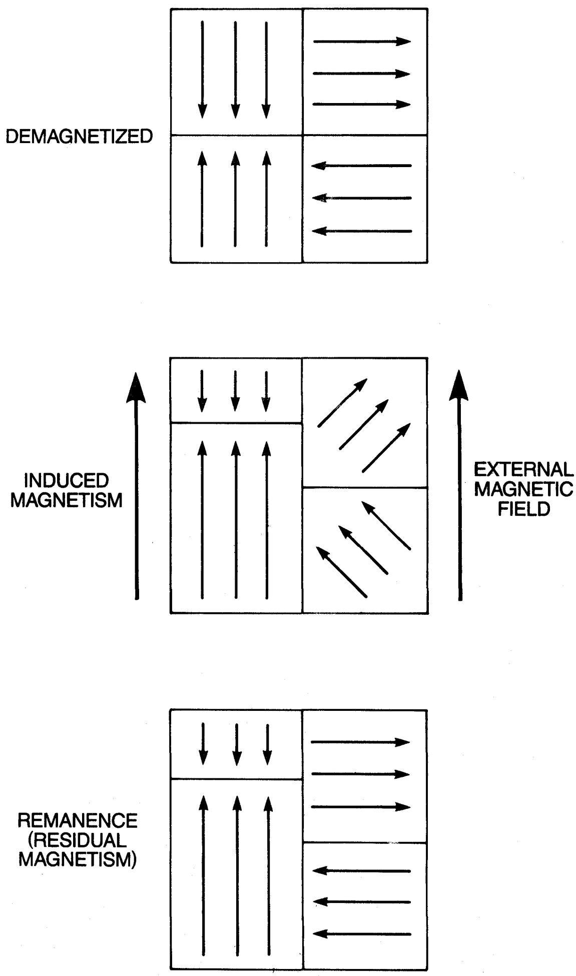


Figure 1-9 Magnetism as a function of domain size.

Disk coatings can be very thin (e.g., by applying thin-film or plating techniques), but tape coatings must be thick enough to resist wear. The result, as shown in Figure 1-12, is that flux reversals in domains close to a tape surface will result in sharp pulses, easily extracted from read-circuit noise, but reversals deep in the coating tend to widen the read-pulse output — limiting the number of detectable flux reversals per inch.

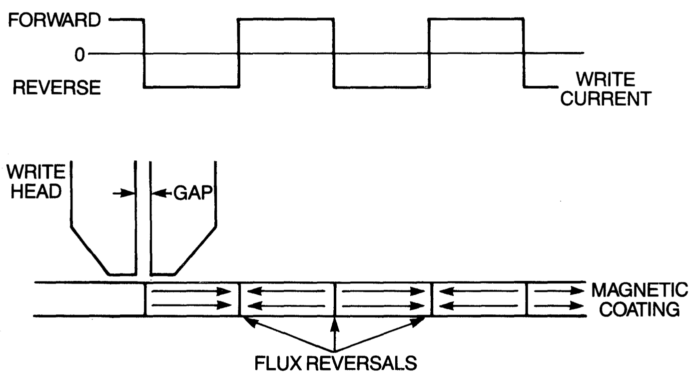


Figure 1-10 Flux reversals produced by write current.

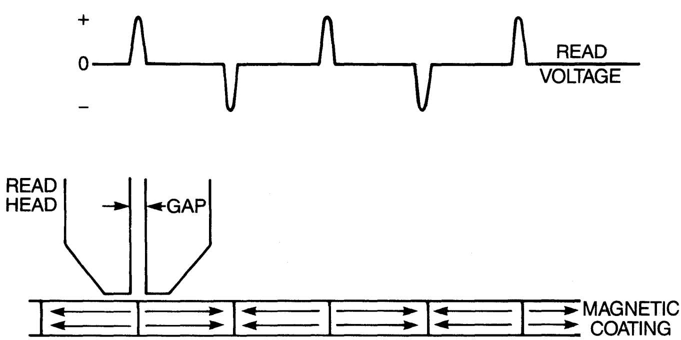


Figure 1-11 Read pulses generated by flux reversals.

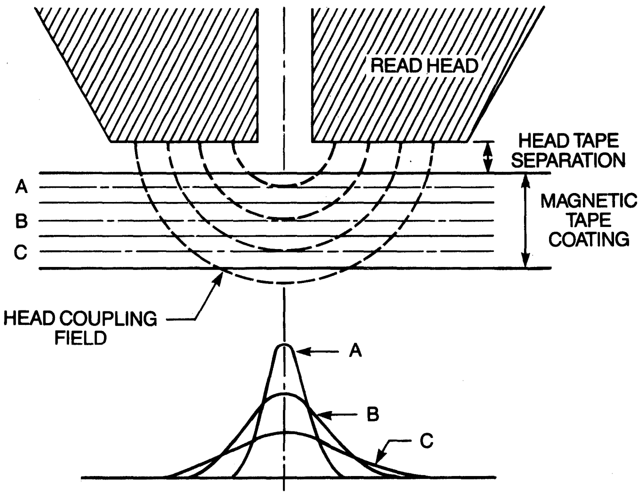


Figure 1-12 Output as a function of saturation.

SATURATED VS. UNSATURATED

The conventional way to record digital data has been to fully "saturate" the magnetic coating, magnetizing not just the surface but the full depth of the coating. Figure 1-13 shows the

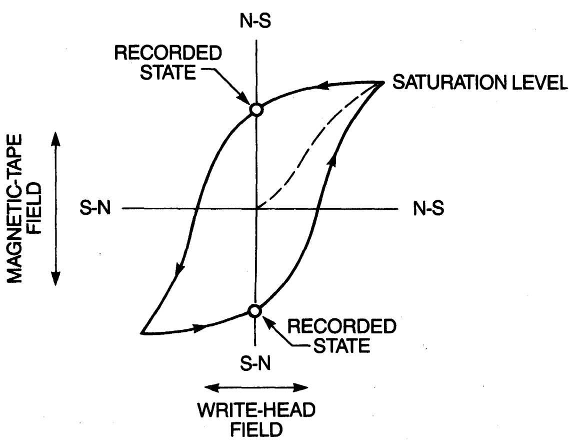


Figure 1-13 Conventional saturated recording.

hysteresis curve for a fully saturated recording. The only way to increase the flux reversal density under these conditions is to use a thinner coating or narrow the head gap (e.g, again by thin-film techniques) to limit the head-coupling field.

An alternate approach is to erase the tape to a neutral state and "lightly" write the flux reversal states in domains near the surface. This unsaturated-recording technique is shown in Figure 1-14, and has proven to be a reliable way to increase flux-reversal densities with conventional heads and industry-standard tape materials.

First, however, the full depth of the tape must be "erased" to a neutral state. Figures 1-15a and 1-15b illustrate how a strong high-frequency signal (on the order of 3.5 megahertz) can accomplish this objective. As the tape leaves the trailing edge of the erase-head gap, the signal attenuates with distance, "writing" a smaller and smaller hysteresis loop throughout the coating.

A secondary advantage of this AC-erase technique is that all flux-reversal states are recorded on a neutral medium and are therefore not "biased" in the direction of an "erased" state which, in the case of conventional digital recording, is fully sat-

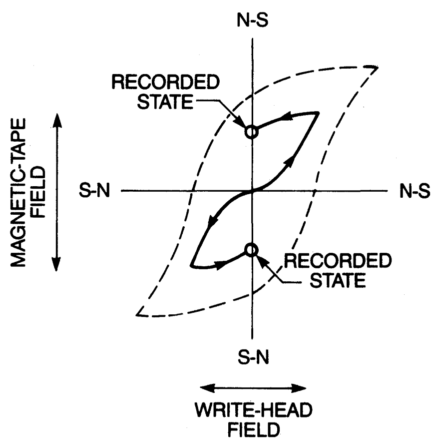


Figure 1-14 Unsaturated recording for higher densities.

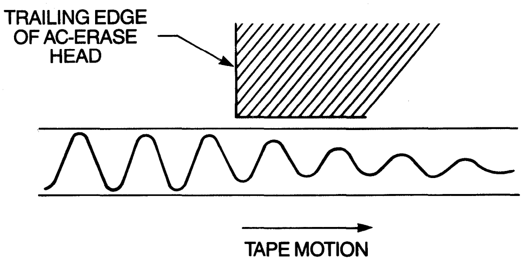


Figure 1-15a Diminishing AC-erase field.

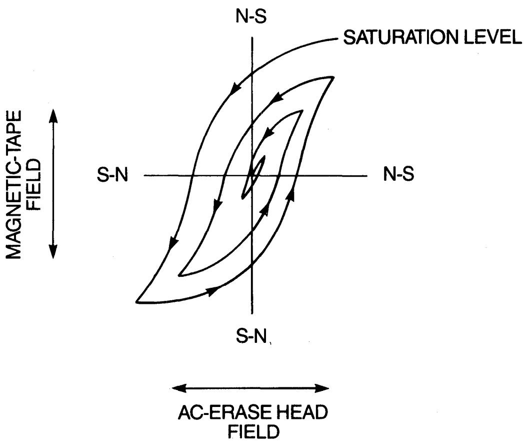


Figure 1-15b Effect of diminishing AC erase field.

urated in the direction defined by the standard for a particular recording techique.

DIGITAL CODES

Flux reversals recorded on disk or tape have no significance until interpreted by the system. They exist in a space domain along individual tracks but their function—assuming constant relative motion between the medium and head—is to create a meaningful signal in the time domain.

Nearly all digital-coding techniques start with a division of the time domain into "bit-cell" intervals. The significance of an individual flux reversal then depends on whether the reversal occurs at the beginning or mid-point of a bit-cell interval, and in some cases, the direction of the reversal.

Considering the demand for ever increasing data densities, it is understandable that the selection of a particular code is based largely on the efficiency with which flux reversals are converted into binary information, ZERO's and ONE's. In the ideal case, there should be fewer flux reversals than the number of data bits they represent. Ideally, too, the code itself should provide its own "clock" for identifying the bit-cell intervals. Lacking this feature, a separate clock track may be required—or an extremely accurate oscillator must be provided to maintain the bit-cell divisions during intervals without flux reversals.

The two requirements tend to be contradictory. An efficient code in terms of flux reversals will not be self-clocking. A self-clocking code will be wasteful of flux reversals. Nearly all of the widely used codes represent a compromise between these two extremes (see Figure 1-16).

NRZ (non-return to zero) is a telecommunication code and by far the most efficient. "Zero" refers to the transmission signal level. Instead of discrete pulses for each data bit, the signal rises or falls only when a ZERO bit is followed by a ONE bit or a ONE by a ZERO. NRZ coding reduces signal bandwidth by at least half. It also requires precise synchronization between source and destination in order to maintain bit-cell divisions during the transmission of long strings of ZERO's or ONE's.

NRZ can be used to transmit serial data to or from a magnetic-recording device, disk or tape. But the extended intervals which can occur between flux reversals limit its usefulness as a recording technique.

NRZI (NRZ, change on ONE's) is the next most efficient code. It is widely used for tape recording and to an increasing degree, disk recording. All ONE's are clocked, but special steps must be taken to compensate for the absence of flux reversals during strings of ZERO's. In the case of parallel-bit tape recording (Chapter 4), parity-bit ONE's serve as a clock when all other bits in the byte are ZERO. In the case of serial-bit recording, data can be converted to a "run-length limited" code (see below) which restricts the number of successive, unclocked ZERO's.

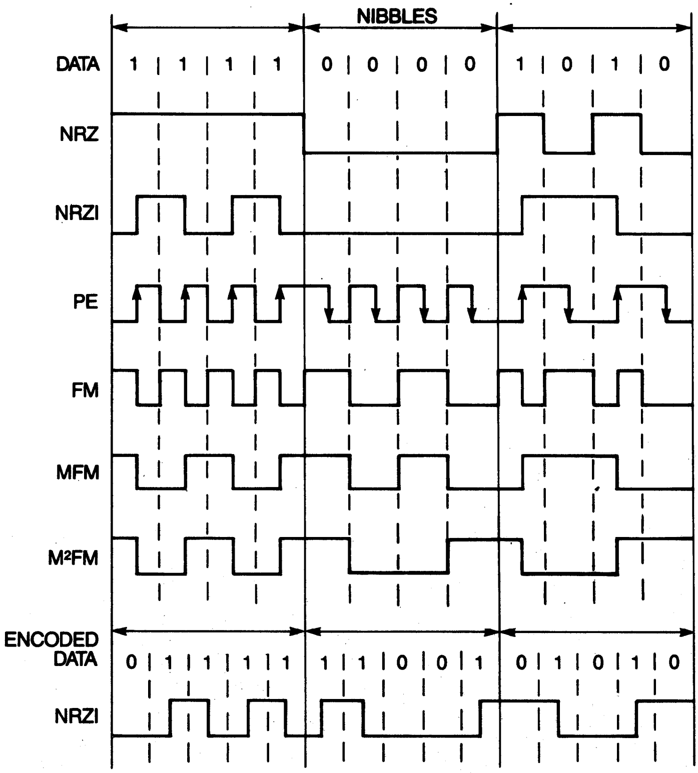
PE (phase encoded) is the least efficient of the coding methods but is completely self-clocking. The direction of a flux reversal at the middle of each cell indicates whether the bit is ONE or ZERO. Either one or two flux reversals occur, therefore, during each bit-cell interval. The effect is to shift the "phase" of the signal by 180° each time there is an NRZ-type transition between ZERO's and ONE's.

FM (frequency modulation) is equivalent to PE and was the first choice for early disk-recording systems. Every bit-cell interval is clocked by a flux reversal at the start of the cell. ONE's are marked by an additional flux reversal at the middle of the cell, doubling (modulating) the frequency of flux reversals for a series of ONE's compared to a series of ZERO's.

MFM (modified FM) is twice as efficient as FM or PE. It retains the middle-of-the-cell flux reversals for ONE bits but eliminates the cell-boundary clock reversal for a ZERO which is preceeded by a ONE. Clock-type flux reversals occur only when a ZERO is preceeded by another ZERO.

M^2FM (modified MFM) further reduces the number of clock-type flux reversals. Cell-boundary reversals are written only in the case of ZERO's which follow bit cells without a flux reversal—approximating the flux-reversal spacing of 0,2 run-length limited NRZI code but still requiring the system to discriminate between flux reversals that occur at the middle of a bit-cell interval (ONE's) and clock-type reversals at the start of a bit cell.

RUN LENGTH LIMITED (Figure 1-16) illustrates how a 4-bit data nibble (half of an 8-bit byte) can be converted to a 5-bit code in which no more than two ZERO's occur in succession, even when the converted codes are recombined into 10-bit "bytes." In run-length terminology, this is a 0,2 code, indicating the minimum and maximum number of ZERO's between ONE



DATA NIBBLE	RUN-LENGTH CODE	DATA NIBBLE	RUN-LENGTH CODE
0000	11001	1000	11010
0001	11011	1001	01001
0010	10010	1010	01010
0011	10011	1011	01011
0100	11101	1100	11110
0101	10101	1101	01101
0110	10110	1110	01110
0111	10111	1111	01111

Figure 1-16 Flux-reversal codes for digital recording.

bits—in this case, zero and two. (The remaining, non-data 5-bit codes—16 of the possible 32—can be used for formatting purposes.)

The 4-to-5 code conversion increases the amount of information that must be recorded by 20%. But NRZI recording more than compensates for this increase. All flux reversals occur in the middle of the bit cell. In most of the other coding techniques, flux reversals can occur at either the start or the middle of the bit-cell interval. There are, therefore, twice as many flux-reversal "windows" to be monitored by the data-recovery circuitry. For a given flux-reversal detection capability, run-length encoded NRZI allows twice as many recorded bits (60% more data bits) to be written.

DATA FORMAT

In addition to establishing the bit-value significance of the flux reversals, the system must "format" the recorded data so that (1) selected records can be located for subsequent reading or rewriting and (2) the accuracy of the record can be verified and corrected during a write operation and again when the data is read.

There is, in general, no direct correlation between the way a host computer organizes information and the way data is stored on disk or tape. Instead, a specified number of bytes, typically ranging from 128 to 2,048, is recorded as an identifiable unit, separated from the balance of the data by a distinctive "gap." In disk terminology, the unit is normally called a "sector." In tape storage, the unit is either a "block" or a "record" (not to be confused with a data-base record).

The terms are somewhat interchangeable. Thus, dating from the time when tape was the principal mass-storage medium, disk-oriented software may include "blocking" and "deblocking" utilities to facilitate the assembly and disassembly of sector-sized data segments.

To facilitate the search for a particular tape record, data can also be divided by distinctive "file-mark" blocks into arbitrary-length files (not to be confused with data-base files). Another important concept in streaming-tape operations is the "transmission"—a nearly continuous transfer of hundreds of data blocks at an average rate that will support uninterrupted tape motion.

The basic unit, however, is the data block (or sector) with its identifying code (e.g., tape sequential address or track/sector numbers), verification data, and intervening gap—all of which represent "wasted" storage capacity, independent of the length of the data block itself. Longer blocks reduce this overhead as a percentage of the total, but increase both the chance that an error will occur as the block is written or read and the time required for corrective action. The length of the block represents, therefore, a tradeoff between system throughput and efficient use of the storage medium.

DATA INTEGRITY

To be effective, a mass-storage system must be able to record and read back data that is essentially free of errors. Maximum uncorrectable error rates should be on the order of one error in every 10^{10} or 10^{12} bits.

Three separate error-processing steps are required in both the write and read modes. The first is to detect that an error has occurred. The second is to correct the error, if possible, without any intervention by the host computer or operator. If the attempt is successful, a "soft" error has occurred, affecting throughput but not the integrity of the stored data. If the error can not be corrected, the final step is to notify the host that the mass-storage system has encountered a "hard" error.

A variety of error detection and correction schemes can be implemented. Many use CRC (cyclic redundancy check) characters as the basic test for write-and-read accuracy. Appended to the end of a data block, the CRC characters represent the quantity that remains when the large binary number represented by all the bits (not bytes) in the block is divided by another large binary number, normally expressed as a polyonominal (e.g., $x^{16} + x^{12} + x^5 + 1$). When the CRC characters themselves are included in this division, the remainder should be zero. If not, an error has occurred.

Error correction is more complex and varied. It can take the form of repeated rewrites and rereads. Or the block format may include ECC (error-correction code) characters that help either the mass-storage system or the host computer to identify the particular bit or bits that have been incorrectly written or read.

ECC algorithms usually involve parity checks on overlapping groups of bits. There is a direct relationship, therefore, between the number of ECC bits and the number of separate or contiguous error bits that can be detected and corrected.

ECC bits represent a fixed data-storage overhead, reducing the amount of data that can be stored on a given disk or tape surface. If, as in the case of typical rigid-disk systems, single-bit errors predominate, a relatively efficient ECC scheme can be devised. But if the typical error is a "dropout" of dozens of adjoining bits, such as those caused by flaws in a streaming-cartridge tape, a more effective, tape-saving solution is to verify each block as it is written and simply rewrite the few blocks that fail a CRC error-detection test.

INTERCHANGEABILITY

Streaming tape has become an important mass-storage backup medium because (1) it represents the lowest cost alternative for storing large volumes of data and (2) it is removable.

Removability also applies, of course, to a variety of other storage media, such as start-stop tape, flexible disks, disk cartridges, and disk storage modules. In every case, removability implies a nearly unlimited capacity for detached, archival storage and, equally important, the ability to retrieve the stored data at any time by returning the media to the originating device.

Removability should also imply that the data can be recaptured by not only the originating device, but by a population of identical or similar devices—a much greater challenge. In brief, the storage medium should be interchangeable.

The importance of interchangeability will vary with the application. In a single-device installation that does not depend on "imported" software or data exchanges with other installations, the only concern would be for media compatibility as equipment is upgraded or expanded. The same could be true for well-defined user groups, such as the buyers of a particular type of word-processing or small-business system. Interchangeability becomes critically important, however, when commercial software, for example, is marketed through multiple distribution channels.

Interchangeability standards may be those of an individual company, applying only to its own products, or they may be industry wide, promulgated by either a standards organization or through the dominance of one or two suppliers. The extent of the interchangeability also depends on the maturity of the product. Industry-wide media and format standards exist, for example, for reel-to-reel ½-inch tape, and flexible disks. Similar standards are being established for newer media, such as streaming-cartridge tape. Meanwhile, users can generally be assured (or should assure themselves), that media is completely interchangeable between identical devices produced by the same company.

DEVICE INTELLIGENCE

The essential functions of a magnetic-recording device are to "write" flux reversals (not data), "read" flux reversals, and control the motion and relative positions of the media and write/-read heads. All of these actions are essentially analog. To record digital data, however, a high degree of "intelligence" must be incorporated into the storage system.

Device intelligence would include an ablity to:

Code binary data into a sequence of flux reversals during a write operation.

Decode flux reversals during a read operation.

Perform other data conversions (e.g., to and from run-length limited code).

Divide data into blocks or sectors.

Format the medium (e.g., by generating inter-block gaps, sync marks, and identification codes).

Detect errors during both write and read operations.

Correct errors by rewriting, rereading, or performing an error-correction algorithm.

Perform parallel/serial data conversions.

Report status, as it relates to both medium and device.

Buffer data to accommodate differences between host-device transfer rates and the device write-read rate.

COMBINED INTELLIGENCE

Combined disk-tape mass-storage architectures impose their own intelligence requirements—particularly if the backup storage is streaming tape.

Streaming efficiency and throughput depend on continuous tape motion. This, in turn, requires precise coordination between disk and tape to compensate for transfer rates that may differ by factors up to ten-to-one. The host computer can provide the coordination and the necessary buffering, but at a high cost in terms of computer resources and processing time.

Three different levels of host-computer control over the combined mass storage can be defined. Both Winchester disk and streaming-cartridge tape, for example, record data in a serial format. Both are driven by electromechanical assemblies which require control signals and generate status reports. Separate formatter/controller circuits are therefore required.

The simplest configuration—in terms of interface hardware—is illustrated in Figure 1-17. Formatter/controller circuits for two non-intelligent devices are embedded in the host computer or packaged as separate modules.

The most important result of this design decision would be that all data transfers to and from the disk and streaming-cartridge drives must be processed by the computer program, interrupting its normal operations, even though the transferred data would be, in most cases, an exact copy.

Computer main memory would also have to serve as a buffer to compensate for differences in disk and streaming-cartridge transfer rates. Disk read and write rates range from 500 Kbytes to over 2 Mbytes per second, compared to a typical 30 Kbytes or 90 Kbytes per second for streaming-cartridge drives.

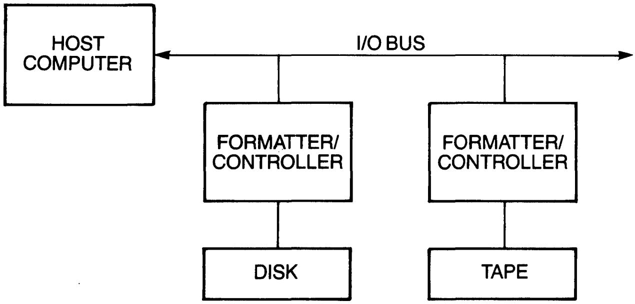


Figure 1-17 Non-intelligent devices with separate formatter/ controllers.

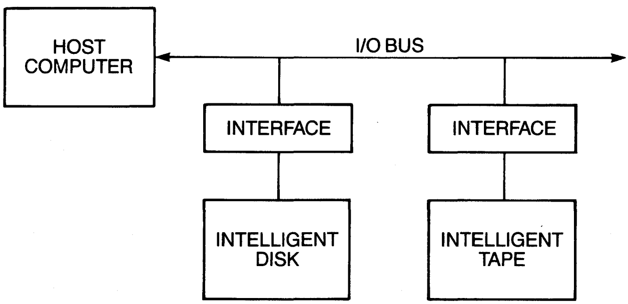


Figure 1-18 Intelligent devices with separate host interfaces.

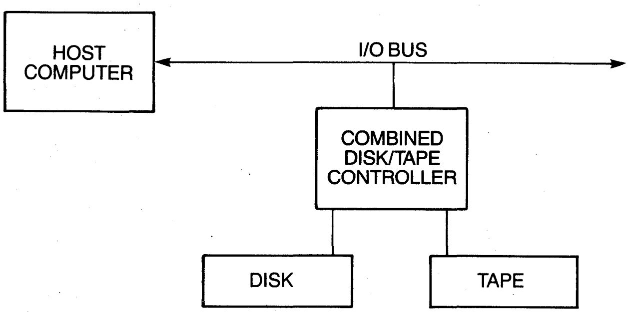


Figure 1-19 Combined disk-tape mass storage system.

These transfer-rate comparisons can be misleading, however, particularly if the disk is assembling or distributing files. Track-seek and rotational-latency times (see Chapter 7, System Considerations) can sharply reduce the effective transfer rate so that host-computer buffering is actually required for a slower disk rather than the nominally slow streaming cartridge. To sustain efficient streaming, the computer would have to accumulate tens or hundreds of Kbytes.

Figure 1-18 shows a more sophisticated configuration. Intelligent versions of the two mass-storage devices have taken the place of the basic drives. Interfaces are reduced, typically, to host-adapter cards and connecting cables. Device intelligence has assumed most of the control and formatting functions. Equally important, the devices usually include buffers, typically for error-processing purposes, which help to match the transfer rates and assure a steady stream of data to or from the streaming-cartridge drive.

But the host computer still serves as a conduit and must supervise the transfers, even when the flow is from one device to the other without any alterations in the data.

The effect of this overhead burden on the efficiency of the total system will depend, of course, on the computer's load factor. The complete transfer of a 20-Mbyte disk record can take up to four minutes, an excessive amount of time in an interactive, real-time environment, but acceptable during hours when the computer is inactive.

All of these device-to-device penalties can be eliminated by the configuration shown in Figure 1-19. The connection between the host and the Winchester/streaming-cartridge system is reduced to a single data-and-control interface. Controller and formatter functions for both devices are independent of the host, and a direct data channel is provided for device-to-device transfers, including the necessary buffering facilities. With this configuration host to device data transfer cannot occur with both devices simultaneously.

An example of the architecture shown in Figure 1-19 is described in Chapter 8. Connection with the host computer is through a combined disk controller that interfaces, in turn, with a disk drive and an intelligent streaming-cartridge drive. The concept, however, remains the same. Device-to-device transfers are totally off-line and "transparent" to the host—once the required transfer commands have been issued. As a bonus, even the fact that the disk is "busy" with a disk-tape transfer is transparent to the host software. The host's operating system can still access the disk at any time by interrupting a lower-priority backup operation.

SHARED MASS STORAGE

Off-line transparency becomes critically important in a multiple-user or local-network environment. Major benefits result when all of the mass-storage facilities, both primary and backup, are shared. But every user should still have near-instant access to the shared resources.

In the case of digital-PBX and similar star-type networks (Figure 1-20), the storage system would typically interface with the central control unit. Storage for all the users on a loop-type network (Figure 1-21) could be appended to the message-routing master controller. Centralized disk/tape storage for a token-passing ring network (Figure 1-22) or a bus-type configuration (Figure 1-23) could be treated as just another network node—interfaced with the network through either a microprocessor control module or hardwired circuits to make the serial-parallel conversions and generate the required commands and control signals for both on-line accesses and off-line disk/tape data transfers.

"Economies of scale" provided by such shared-resource configurations can be impressive. Assume, for example, a relatively small network with just eight user-station nodes, each consisting of a microprocessor-based terminal running under CP/M. Directly addressable storage for each station is limited to approximately 8 megabytes, well within the capacity of a typical small Winchester drive. One alternative, then, would be to equip each station with its own disk storage, allowing it to directly access its own database, operating-system modules, and application programs. But at a cost of $X for each station, the total primary-storage investment would escalate to $8X.

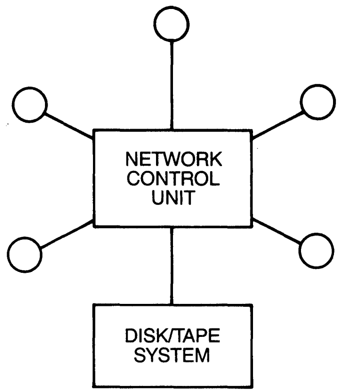


Figure 1-20 Star-type network with shared mass storage.

By comparison, a single Winchester drive with an unformatted capacity of 80 megabytes and a price tag of $2X can be easily partitioned into separate 8-Mbyte logical disk units—one for each user—with storage left over for such tasks as assembling files for backup storage in a high-throughput, off-line mode. The result is an immediate $6X reduction in network costs, without any noticeable loss of system performance from the users' point of view. Each station would still have its own dedicated storage, independent of other users on the network.

Similar savings can be achieved by using ¼-inch streaming-tape cartridges as the removable media for archival storage, program loading, and data exchange. Streaming-cartridge drives can serve as direct replacements for the flexible-disk drives now typically installed at each station. The higher capacity of the streaming cartridges (e.g., 45 megabytes) would multiply the number of megabytes of attached backup storage and reduce the per-megabyte storage cost to a fraction of its current value (see Chapter 5).

In terms of network costs, however, even greater savings could be realized by combining Winchester disk and cartridge drives in a single disk/tape storage system. In the example above, only one 45-megabyte cartridge drive (with one media change) would be required to provide complete backup for the

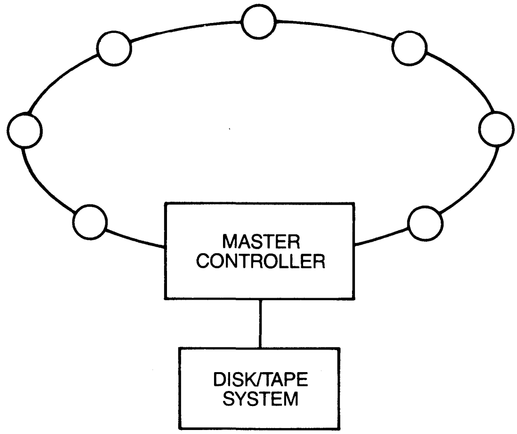


Figure 1-21 Loop-type network with shared mass storage.

80-megabyte Winchester, compared to the eight drives required when each station is equipped with its own backup device.

Balanced against these hardware savings, of course, would be such factors as the increase in network traffic and added contention between users—not only for access to the network but to the disk/tape system itself. In fast-growing applications such as office and laboratory automation, however, network utilization is generally low and intermittent, and most of the system resources are "idle" most of the time. Centralized data storage would increase system efficiency, therefore, and offer the system designer a number of other tangible benefits.

Logical disk units and tape files, for example, could be dynamically altered to meet the needs of individual users, instead of being fixed by network hardware. Logical units could also be both dedicated to single users and simultaneously shared by all the users on the network—assuming adequate controls over data security and integrity. Operating-system utilities, conversion tables, and historical data could be stored only once within the system, yet accessible to any user. Data exchanges between user programs could even be accomplished by copying records from one logical unit to another without any network transfers.

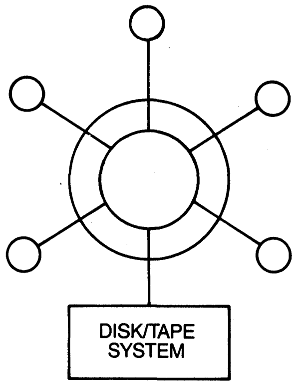


Figure 1-22 Ring-type network with shared mass storage.

Most of these benefits would accrue, of course, with any combination of Winchester disk and backup media. All backup alternatives should, in fact, be evaluated.

The starting point in this evaluation process will usually be the Winchester disk itself, selected on the basis of such parameters as size, capacity, and transfer rate. The following chapter traces the history of rigid disks and summarizes the factors that must be considered.

The next chapter examines two backup alternatives: flexible disks and magnetic tape.

Flexible disks have the advantage of small size, low equipment costs, and reasonably low media costs. But even the larger flexible disks are limited in terms of their transfer rates and the amount of backup capacity they can provide without frequent media changes.

Reel-to-reel tape, by comparison, offers high backup capacities, fast transfer rates, and reasonably low per-megabyte media costs. But the hardware costs and physical size of ½-inch tape drives are generally not appropriate for small-to-medium computer applications.

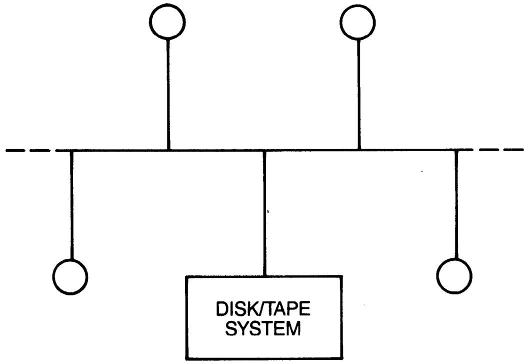


Figure 1-23 Bus-type network with shared mass storage.

The balance of the book concentrates on the most viable of all the backup options: ¼-inch tape cartridges which, in the streaming mode, combine very high capacities—approaching those of reel-to-reel tape—with the small size and convenience of flexible disks.

Comparative cost figures for these various alternatives are given at the start of Chapter 5. They indicate that at the present stage of development, ¼-inch streaming-cartridge tape represents the optimum choice for nearly every application in which the requirement calls for capacities in the range of 10 to 50 megabytes.

With streaming-cartridge tape as backup, the entire contents of a Winchester disk in this capacity range can be recorded on a low-cost, removable medium in just a few minutes—without a single media change.

CHAPTER 2

Winchester Technology

Looking ahead, streaming tape may someday be viewed as a primary mass-storage medium for a variety of applications. Present wisdom, dictates, however, the selection of an appropriate disk, followed by the "backup" device.

Disk selection is not an easy task. Few areas of computer technology are undergoing such rapid and dynamic change, both revolutionary and evolutionary. A vast array of rigid-disk choices face the system designer and user. Figure 2-1 lists the major milestones in the history of rigid disks and illustrates the accelerating pace of new-product introductions.

Most of the recent disk developments have centered on "contamination-free" Winchester technology. Economical disk storage requires high flux-reversal densities. As noted in the previous chapter, one way to increase such densities is to reduce the distance between magnetic-recording head and medium to the microinch level. This, in turn, means that disk and head must be completely free of dust, lint, and finger prints (Figure 2-2).

COUNTERCURRENT TRENDS

Two opposing trends help to explain the proliferation of disk-storage types and sizes.

First is the constant demand for higher and higher capacities within the form factors established by existing disk units. The majority of disks attached to minicomputers and microcomputer are used in business-type applications with expanding databases. As new, more interactive applications develop, and as the number of users and terminals multiply, there is an escalating requirement for more on-line storage.

Fortunately, most of the hardware costs of a disk drive are in the motor, heads, head actuator, electronics, and housing. Incremental increases in capacity through additional disk surfaces and higher data densities can be obtained, therefore, at minimal cost.

The other trend, in the reverse direction, is toward smaller-sized units with limited capacities that match the storage requirements of smaller, less costly computer systems. This has been, until recently, the province of the flexible disks described in the next chapter. But even at the entry level, such systems can benefit from the increased reliability and higher capacities offered by fixed-disk units.

Of course, once the new, smaller disks have established footholds in these new, high-volume applications, the upward trend toward ever higher capacities immediately begins, blurring the distinctions that originally existed between the disk-size categories.

WINCHESTER ROOTS

Disk technology started a quarter century ago with the introduction of a large, cumbersome, fixed-disk unit with fifty rotating surfaces, 24 inches in diameter, a single write-read head assembly, 600-millisecond seek time, and a modest capacity of approximately 5 megabytes.

A half decade later, capacities had increased by ten-fold. Multiple head assemblies, one for each surface, introduced the concept of a "cylinder"—simultaneous access to multiple tracks, one above the other, with a single head movement.

Removable modules based on 14-inch-diameter disks appeared. Packing densities increased sharply; a single disk-pack could

1957	24-inch, 50 disks, single head
1962	24-inch, 50 disks, head-per-disk
1963	14-inch removable pack
1966	Disk storage drops below attached-tape storage costs
1970	Operating systems based on virtual-storage concept
1973	14-inch removable Winchester
1976	14-inch non-removable Winchester
1979	8-inch non-removable Winchester
1980	8-inch Winchester cartridge
1980	5¼-inch non-removable Winchester
1981	5¼-inch Winchester cartridge
1982	Gigabyte-per-spindle Winchester
1982	3.9-inch non-removable Winchester

Figure 2-1 Milestones in the history of rigid disks.

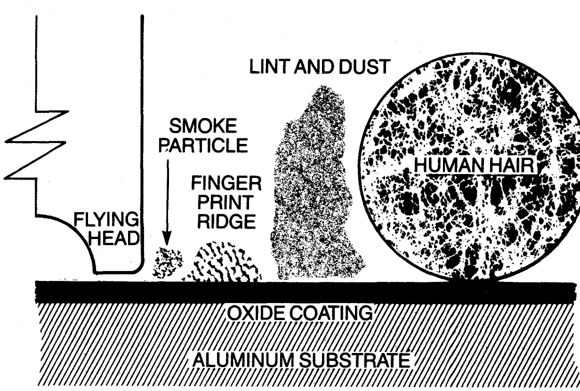


Figure 2-2 Sources of disk contamination.

store up to 100 megabytes. "Flying" head height dropped from a thousandth of an inch to 50 microinches. Bits per inch increased from 100 to over 4,000; tracks per inch doubled to nearly 200.

New operating systems evolved to take advantage of the "virtual" memory provided by disk storage. The distinction between main memory and disk memory became "transparent" to most users. (A new type of input device, a small flexible diskette, was used to load the microprogram that controlled the new disk systems.)

Contamination-free Winchester technology arrived in 1973. The original design goal had been a 30-30 configuration: 30 megabytes of removable storage, 30 megabytes of fixed. The first offering was all-removable, but not cost effective. Heads and disks were sealed in expensive modules that could store only 35 or 75 megabytes.

But the direction had been set. In addition to a controlled environment, Winchester innovations included lightly loaded heads, flying within 20 microinches of the disk surface, an oriented iron-oxide coating to support higher flux-reversal densities, and a silicone or wax coating that permitted heads to slide directly on the surface during "takeoff" and "landing"—eliminating the need for complex head-loading mechanisms.

By the early 1980's, fixed-disk 14-inch Winchester capacities were approaching 600 megabytes and headed toward the present commercially available limit of 1.25 gigabytes per spindle. Drives with capacities of 3 to 6 gigabytes are on the immediate horizon.

SCALED DOWN WINCHESTERS

Winchester innovations also served as the springboard for miniaturized rigid-disk systems. First were compact single or double-platter, non-removeable 14-inch units with capacities down to 10 megabytes. Then in 1979, the 8-inch Winchesters appeared. A 5¼-inch unit followed within a year.

Today there is an almost continuous spectrum of small-to-medium Winchester sizes and capacities: 3.9, 5¼, 8, 9, 10½, 14 inches. Capacities range from five megabytes to over 600 megabytes—with an equally continuous spectrum of data-storage capabilities as the capacity of individual drives continue to increase and overlap those of the next larger units.

WINCHESTER CARTRIDGES

Equally important has been a trend toward removable Winchester storage, in both removable-only and fixed-removable configurations. Winchester cartridges have capacities in the range of 2 to 15 megabytes and in principle can serve a backup role. But costs are high compared to other offloading techniques (Chapter 5). Their principle applications, therefore, will be to expand disk capacity at minimum cost in multiple-program applications and to serve as "user-dedicated" disks in multiple-user environments.

PHYSICAL SIZE

With overlapping capacities between sizes, physical form factors can easily become the first and most important selection criteria.

In fact, as computer systems become more "personal," emphasis is shifting toward such features as size, appearance, and operator convenience—often at the expense of device performance and cost effectiveness.

Aiming at the flexible-disk upgrade market, the first 8-inch drives were deliberately configured so that they would fit into an existing 8-inch flexible-disk slot on the computer front panel. Following suit, 5¼-inch Winchesters have been designed to fit into 5¼-inch flexible-disk slots.

These user-oriented, marketing decisions make sense, of course, in the case of removable-cartridge Winchester drives (and the tape cartridge described in Chapter 5). A more general rule, of thumb, however, would be that 5¼-inch (and smaller) disk drives would be the first choice for personal, professional and other low-cost computer systems, 8-inch drives for more sophisticated desktop computers, 14-inch drives for "furniture-sized" systems—packaged in desks or rack-type enclosures.

STORAGE CAPACITY

Data capacity is, perhaps, the most difficult decision to make in the selection process. It represents, at best, a moving target. Tomorrow's storage requirements are certain to be greater than today's. But by how much? Should the growth route be through additional drives, higher-capacity drives, or removable disk cartridges?

And what will be the impact of new backup technologies, such as streaming-cartridge tape? Will it be more cost effective, or technically feasible, to hold disk storage at a constant level and expand system capacity by adding backup? If so, what will be the logistics of the backup? In a multiple-user environment, will the storage be in a central location or distributed at each work station?

All of these questions, present and future, must be answered in the context of the application. With backup considerations included in the equation, parameters affecting the decision include processor cycle time, memory size and speed, size of the user database (or databases), structure (segmentation) of the system software, size and number of application programs (and frequency of use), number and location of users. Capacity is even a function of the way the computer system organizes data into records and files.

Figure 2-3 shows a Winchester disk format. Typically, the length of the sector can be set at 128, 256, 512, or 1024 data bytes. Larger sectors increase the effective capacity of a given disk by reducing the amount of space devoted to formatting codes. But this advantage is lost if records are short and most of the sectors are padded out with "filler" data.

UPWARD MOBILITY

A fail-safe option, of course, would be to select a drive design with enough potential capacity to meet any future storage requirements.

Drives in each size category have been doubling in capacity each year. Eventually, however, plateaus must be reached. There is a limit on the number of disk surfaces that can be added without exceeding form-factor limits. Further capacity increases must then take the form of increased data densities (e.g., flux-reversals per inch, tracks per inch, more efficient codes).

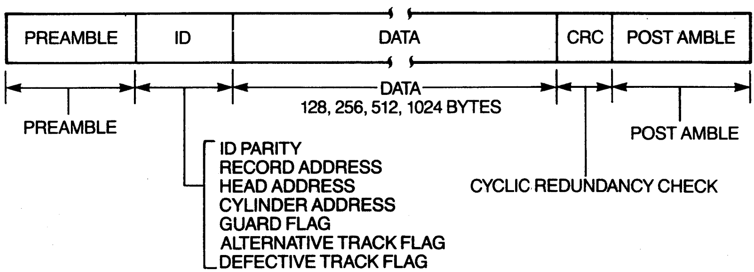


Figure 2-3 Sector format, Winchester disk.

There is considerable room for growth. Thin-film heads with narrower gaps (see Chapter 1) can be combined with thinner magnetic coatings—sputtered or plated—to increse linear densities to over 20,000 flux reversals per inch. The same techniques can be used to narrow the heads—and tracks—to raise track densities to over 1,000 per inch.

Higher track densities will accelerate the move away from head-positioning stepper motors and toward solenoid-type "voice coil" actuators with theoretically infinite track-following resolution.

Disk manufacturers are also shifting to a 2,7 run-length limited code which increases data density by 50% compared to conventional MFM.

And then there is the potential of new magnetic-recording technologies, such as "vertical" recording with north-south magnetic poles perpendicular to the disk surface instead of end-to-end along the track.

Adding up these possibilities, it is safe to look ahead to 5¼-inch drives with capacities in the range of 100 plus megabytes, 8-inch drives in excess of 150 megabytes, and 14-inch drives to over a gigabyte.

TRANSFER RATES

All of these developments can also influence the transfer rate, another critical disk-selection criterion. The transfer rate directly affects system throughput and indirectly (see Chapter 7) the efficiency and speed of streaming-tape backup and restore operations.

It is the average transfer rate that counts, and again this is a function of the application. If write-read accesses are scattered (e.g., because different users are accessing the disk, or file-management algorithms require frequent assembly or disassembly of individual records), track-seeking and sector-searching delays will reduce the effective transfer rate to a fraction of the theoretical specification-sheet value based on data density and rotational speed.

A series of application-dependent cost-performance tradeoffs must be individually evaluated. Higher rotational speeds reduce the "latency" time as the system waits for a desired sector to pass under the write/read heads. Multiple heads reduce both the number of head repositions and the distance that must be traveled. Lower-cost stepper-motor actuators are normally "open loop"—moving the heads from track to track at a constant, relatively slow rate (Figure 2-4). Voice-coil actuators are more expensive but inherently faster, accelerating and decelerating in response to feedback signals from a closed-loop servo system.

The servo system can take two forms. As shown in Figure 2-5, a complete recording surface may be dedicated to the head-positioning function. Tracks are prerecorded at the factory and the whole head assembly moves as a unit in response to signals from a read-only head which identifies the tracks during track-to-track positioning and then "follows" any track variations

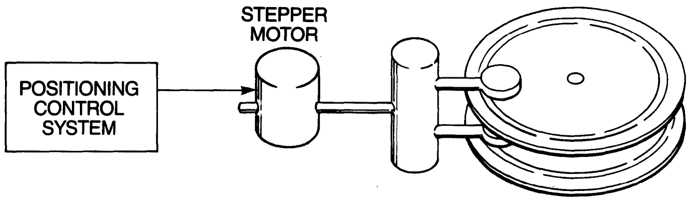


Figure 2-4 Open-loop head positioning.

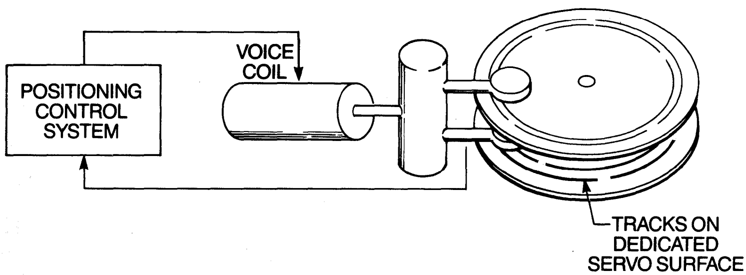


Figure 2-5 Closed-loop with dedicated servo surface.

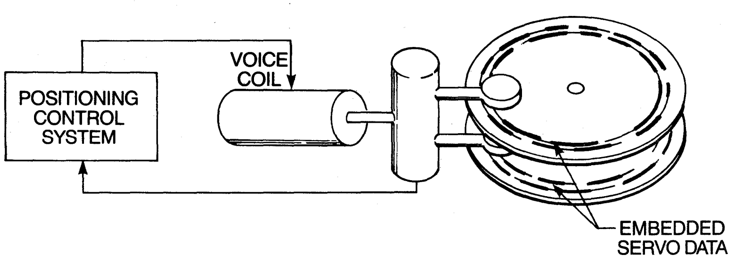


Figure 2-6 Closed-loop with embedded servo code.

which may result, for example, from changes in temperature or faulty spindle bearings. The other technique, illustrated in Figure 2-6, is to embed the servo information in each data track, typically at the start of each sector.

Again, there are tradeoffs—primarily the loss of a complete recording surface versus a reduction in the capacity of each data track. Accuracy of the dedicated-surface servo technique also depends on the stability of the mechanical assembly that links the servo-head arm to the write/read heads. The embedded-servo method eliminates this potential problem and in principle lends itself to higher track densities. But the disks are hard-sectored by servo data prerecorded at the factory, reducing their formatting flexibility.

CHAPTER 3

Flexible Disks

By whatever name—diskettes, floppies, minifloppies, or microfloppies—flexible disks have earned a reputation as the storage medium with nine lives.

Flexible-disk drives were first developed as read-only input devices for loading microcode into rigid-disk systems. When a write capability was added, flexible disks rapidly became a widespread medium for data exchange and program distribution. Then, as small computers came to the fore, flexible-disk drives were transformed, yet again, into random-access mass storage devices.

Today, with the introduction of small-size Winchesters, the adaptable, flexible disk has found a new role as backup for the non-removable disk that threatens to replace it.

Streaming-tape cartridges may take away even this niche. But they also open the way for perhaps the brightest future of all: economical mass-storage systems with streaming tape as bulk storage and flexible disks serving as random-access storage for data being processed. The combination would represent the lowest cost, most flexible system for the small computer systems that are experiencing the most rapid growth—word processing, office automation, and personal computing.

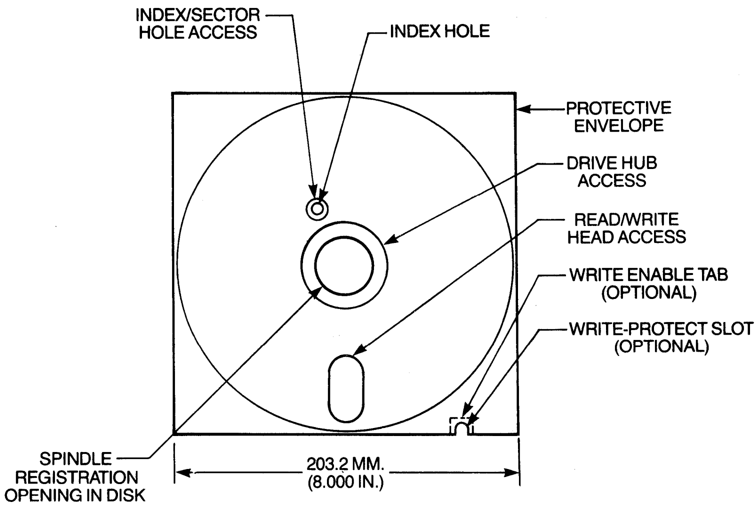


Figure 3-1 8-inch flexible-disk

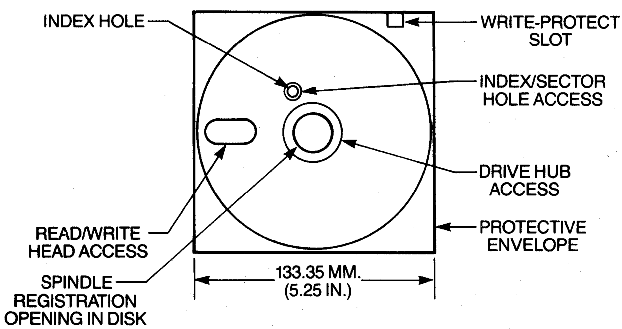


Figure 3-2 5¼-inch flexible disk

DISK-TAPE CHARACTERISTICS

Flexible disks combine the characteristics of rigid disks and magnetic tape. Its shape, of course, is that of a disk. The organization of the data into sectors on concentric tracks is identical to that of other disks. An actuator moves the write-read heads from track to track, providing exactly the same random-access characteristics.

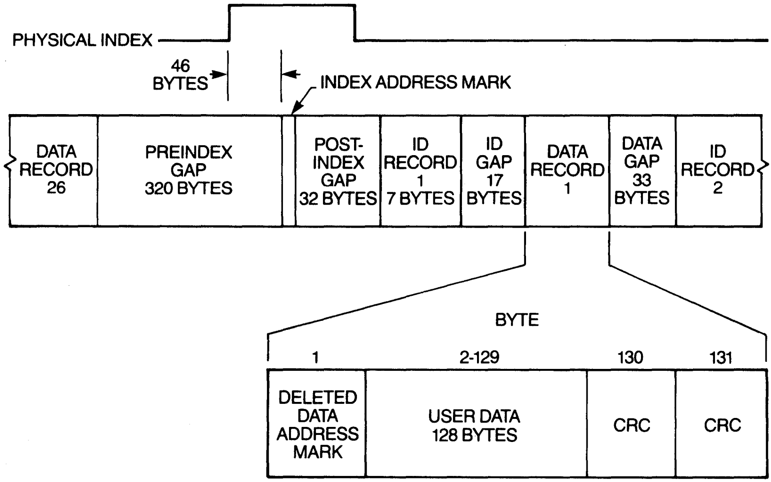


Figure 3-3 Sector format, 8-inch flexible disk.

But flexible disks also have a number of magnetic-tape attributes. In fact, the first disks were stamped from magnetic tape stock and mounted on a foam pad. The substrate is now 3-mil mylar, but is still relatively soft and pliable. At 300 or 360 revolutions per minute—a fraction of the rotational speed of rigid disks—surface speeds also approximate those of magnetic tape. Flexible media and moderate speeds allow the write-read heads to press directly against the recording surface. High linear flux-reversal densities can be achieved, but transfer rates are relatively low, under 100 Kbytes per second. And again like tape, media instability limits track densities, typically to less than 100 per inch.

SHRINKING SIZES

In one important respect, however, flexible disks are identical to their rigid counterparts. They are experiencing the same trends toward smaller sizes and higher capacities.

The original flexible disk, introduced in 1971, had a diameter of 8 inches (Figure 3-1). This was followed by a smaller, 5¼-inch disk in 1976 (Figure 3-2). The two sizes still dominate the field. But the move toward smaller sizes is accelerating. Diameters now range down to 3½ inches.

Flexible-disk drives are also shrinking. Half-high 8-inch and 5¼-inch drives allow two disk units to be installed in the space that had been occupied by a single unit.

But countering these trends has been a steady increase in flexible-disk capacities. An early shift from FM to MFM encoding doubled the data density for a given number of flux reversals per inch. Two-sided recording again doubled the capacity. And the number of tracks per inch has steadily progressed from an original standard of 48-per-inch. Flexible disks are now approaching the lower end of the rigid-disk range. Figure 3-3 illustrates a typical flexible disc sector format.

EVOLVING TECHNOLOGY

For a relatively mature technology, flexible disks are continuing to evolve at a rapid rate.

Flexible-disk access times, for example, have been historically slow. But innovations like the band-type actuator have reduced the track-to-track access time from 10 milliseconds down to 3 milliseconds, average access time from 300 milliseconds down to 90 milliseconds.

Another innovation is a "submerged" servo system in which a low frequency signal is read by a separate servo head that controls, in turn, the position of the write-read heads. And in another development, bridging Winchester and flexible-disk technologies, the disk literally "flies" inside a cartridge-type package, producing very high bit and track densities and potential capacities up to 10 megabytes on a single 8-inch surface.

CHAPTER 4

Tape Storage

Reel-to-reel, ½-inch tape was the original mass-storage medium, and in terms of volume, it is still the premier form of archival storage. Millions of miles of digitally recorded magnetic tape fill vaults, warehouses, and computer rooms around the world.

Industry-standard start-stop or streaming ½-inch tape formats are viable backup alternatives for Winchester disks when the application requires both high capacity and very high (up to a Mbytes per second) transfer rates.

Reel-to-reel systems can achieve these high rates because, unlike disks and most smaller-width tapes, ½-inch tape records data in a byte-wide format instead of serially, bit by bit. For a given speed and linear data density, therefore, ½-inch tape will outperform any other storage medium by ratios up to 8-to-one.

In many respects, however, reel-to-reel tape represents a case of arrested development. Its start-stop format dates from the era when tape was replacing punched cards and papertape because it allowed the host computer to "randomly access" large volumes of stored data without operator intervention. Forced into a role that was not appropriate to its serial nature, tape was easily preempted by rotating disks and relegated to a secondary role—carrying with it the start-stop, search-and-update characteristics that had been so valuable at an earlier time.

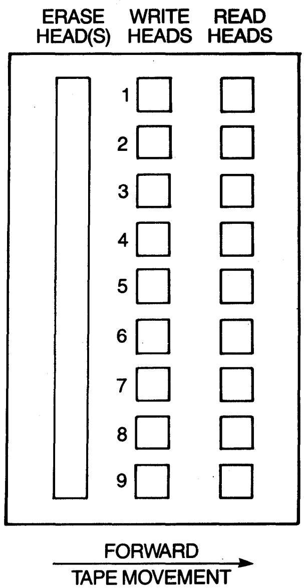


Figure 4-1 Nine-track head for recording on ½-inch tape.

PARALLEL RECORDING

Reel-to-reel tape capacities vary over a wide range. Standard reel diameters are 7, 8½, and 10½ inches. The tape itself can have a thickness of 1 or 1½ mils. Added to these variables, however, are the data density and format. A 10½-inch, 1½-mil, 2,400-foot reel can store approximately 19 Mbytes at 800 bytes per inch (BPI), 30 Mbytes at 1600 BPI, and over 90 Mbytes at 6250 BPI.

The only common denominator is the nine-track parallel record "written" by multiple write-read heads (Figure 4-1). Eight-bit byte and parity bit are recorded simultaneously across the width of the tape (Figure 4-2).

The parity bit ensures that at least one bit will be a ONE to "clock" the bytes when the data is recorded in an NRZI format (see Chapter 1). But if tape or heads are even slightly "skewed," errors can result—limiting NRZI data densities to 800 BPI.

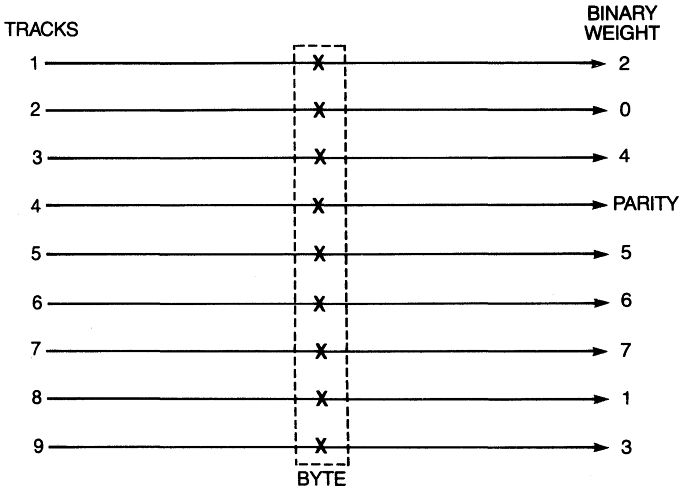


Figure 4-2 NRZI and PE recording, ½-inch tape.

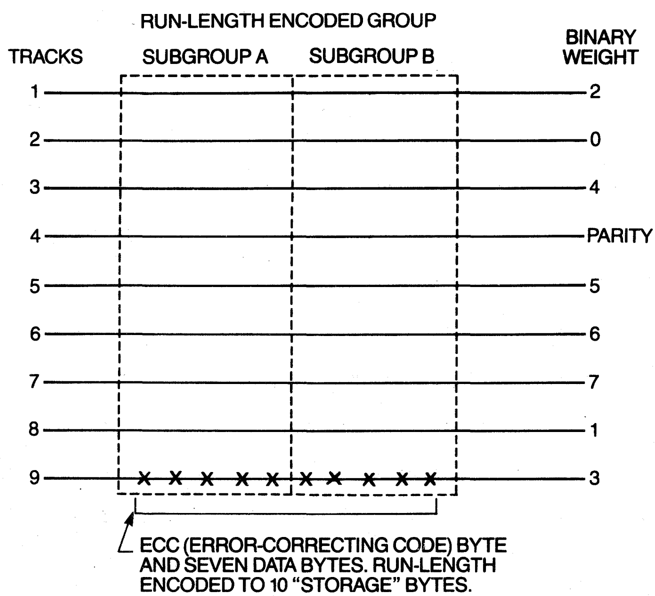
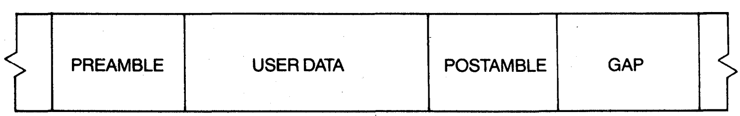


Figure 4-3 GCR (group-coded recording) ½-inch tape.

A phase-encoded (PE) record is less efficient in terms of flux reversals, but each track is self-clocked by the code, eliminating the tape-skew problem. Standard density for PE-encoded tape is 1600 BPI.

The highest-density, 6250-BPI recording method combines the efficiency of run-length limited NRZI encoding with an elaborate GCR (group-coded recording) format that is able to correct read-back errors. Seven data bytes and an ECC (error-correcting code) byte are divided into two sub-groups. Corresponding bits in each subgroup are then subjected to a 4-to-5 code conversion. Ten bytes are recorded for each group of seven data bytes, but flux-reversal densities—and tape speeds—can be very high. In fact, a GCR tape running at 200 inches per second can record or read data at rates over 1 Mbyte per second—approaching those of even the highest-performance disk.

Figure 4-4 summarizes the characteristics of the three tape formats. In each case, there is gap between data blocks to allow a conventional start-stop drive with a low-inertia capstan (and tape stored in mechanical buffers) to halt between blocks and wait for the next write or read command (Figure 4-5). Alternatively, if the command is received on time, the tape will continue through the gap and write or read the next block "on the fly."



CODE	DENSITY	PREAMBLE	POSTAMBLE	GAP
NRZI	800 BPI	—	CRC & LRC	0.6 INCH
PE	1600 BPI	41 BYTES	41 BYTES	0.6 INCH
GCR	6250 BPI	80 BYTES	CRC, ECC, & 80 BYTES	0.3 INCH

Figure 4-4 Standard ½-inch tape formats.

STREAMING HALF-INCH TAPE

Simpler, lower cost, ½-inch "streaming" drives duplicate this format in order to maintain compatibility with the large population of conventional tape transports. Most of them can also record in a start-stop mode, but at low tape speeds—down to 12½ ips compared to the normal 75 to 200 ips—to compensate for their high-inertia drive mechanisms.

Continuous tape streaming, by comparison, occurs at rates up to 100 ips, producing "overshoots" of up to 15 inches if the flow of data is interrupted (Figure 4-6). The tape must then be reversed and again brought up to speed before additional data can be written or read—a process that can take over a second.

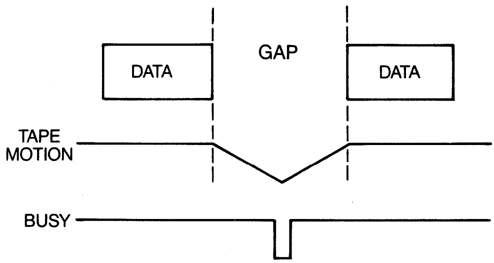


Figure 4-5 Conventional start-stop operation, ½-inch tape.

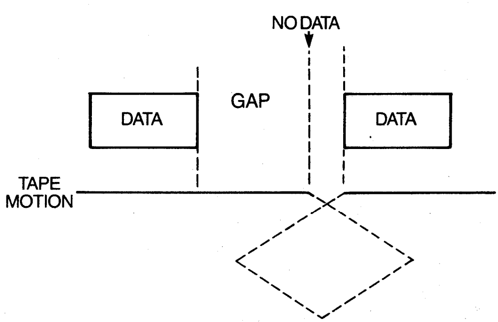


Figure 4-6 Interrupted streaming, ½-inch tape.

SIZE AND COMPLEXITY

Similar "underrun" penalties apply to all streaming-tape devices and are discussed in detail in Chapter 9. The major objections to ½-inch tape drives are the size and complexity of the equipment itself—compared to the ¼-inch cartridge drives described in the next chapter.

CHAPTER 5

Streaming ¼-Inch Cartridge

Tape storage also includes any type of "pre-packaged" tape with self-contained reels and tape path. The storage medium is "mounted" by simply inserting the package into an appropriate drive mechanism.

Typically called cartridges, packaged tapes give system designers a wide choice of backup options. Data-storage capacities range from a few Kbytes to nearly 70 Mbytes—rivaling the storage capabilities of reel-to-reel ½-inch tape but occupying only a fraction of the space.

The tradeoff is transfer rate. Except for a few specialized forms (e.g., continuous-loop cartridges), packaged tape is recorded in a bit-serial or nibble-serial format, compared to the byte-wide reel-to-reel formats. At equal tape speeds and bit-per-inch densities, ½-inch reels outpace other tapes by a ratio of 8-to-one.

SELECTION CRITERIA

Cartridges and reel-to-reel tape—start-stop and streaming—are not, however, the only backup alternatives. Like tape a decade earlier, flexible disks have already made the transition from primary to secondary storage in a number of applications (see Chapter 3). Removable disk packs and Winchester cartridges can also serve a backup role.

BACKUP DEVICE	FORMATTED CAPACITY (M BYTES)	DEVICE COST (QTY. 500)
8" Floppy Disk Drive	.8-1.3	\$300 - \$400
Archive ¼" Streaming Cartridge Tape Drive	20-45	\$950/1100
¼" Start/Stop Cartridge Tape Drive	8.6-15	\$1100
Fixed/Removable Disk Drive (5¼",8")	5-10 (removeable)	\$1500 - \$2200
Removable Cartridge Drive (5¼", 8")	5-10	\$1200 - \$1800
½" Tape, Start/Stop Drive 75 IPS, 6250 BPI (2400 ft)	15-160	\$3000 - \$5000
½" Tape, Streaming Drive 100 IPS, 1600 BPI (2400 ft)	15-42	\$2200 - \$2800

Figure 5-1 Comparative backup equipment costs (1982).

Three different criteria can be applied to the selection of the most practical, cost-effective backup media for a given application. One, of course, is the installed cost of the backup device, normally expressed as the cost per megabyte of removable storage (averaged over total capacity if storage is both fixed and removable). Figure 5-1 summarizes equipment costs for the principal options.

Comparative hardware costs are a major concern for system suppliers but not necessarily the most important consideration for users. Equally important would be the second selection criterion: cost of storage.

Storage costs include the media itself, again compared on a per-megabyte basis. Added to this, however, are system costs based on the time it takes to transfer data to and from the backup media. Operator time and attention also add to the cost when frequent media changes must be made.

DEVICE	FORMATTED CAPACITY (M BYTES)	NO. OF MEDIA FOR 20/45 M BYTES	TOTAL MEDIA COST 20/45 M BYTES	RECORDING TIME (MIN) 20/45 M BYTES	OPERATOR INVOLVEMENT
8″ Floppy Disk Drive	8-1.3	16/32	$80/160	34/68	Multiple Media Insertions
Archive ¼″ Streaming Cartridge Tape Drive	20-45	1/1	$30/30	4/9 (90 ips) 12 (30 ips)	One Media Insertion
¼″ Start/Stop Cartridge Tape Drive	8.6-15	3/6	$90/180	60/120 (30 ips)	Multiple Media Insertions
Fixed/Removable Disk Drive (5¼″,8″)	5-10 (removeable)	2/4	$200/400	4/8	Multiple Media Insertions
Removable Cartridge Drive (5¼″, 8″)	5-10	2/4	$200/400	4/8	Multiple Media Insertions
½″ Tape, Start/Stop Drive 75 IPS, 6250 BPI (2400 ft)	15-160	1/1	$10/10	6/12	One Media Insertion
½″ Tape, Streaming Drive 100 IPS, 1600 BPI (2400 ft)	15-42	2/3	$20/30	11/18	Multiple Media Insertions

Figure 5-2 Comparative backup media costs (1982).

Figure 5-2 tabulates these factors, using a 20-Mbyte transfer as the basis for comparison. Their relative importance will vary according to the application. If large volumes of data are destined for archival storage, media costs become a principal concern. If backup and restore operations occur during periods of heavy system usage, transfer times take precedence. If an operator is distracted from critical tasks, or is unavailable, the number of media insertions may be the deciding factor.

Closely associated with these concerns is the third selection criterion: the physical and functional requirements of the application. Included here would be such factors as existing panel space, cosmetic appearance, operator convenience (and skills), space for storing archival records, location of users, and the ease with which the media can be shipped or mailed. Any one of these requirements may be more than sufficient to override economic considerations based on equipment or media costs.

Standard ¼-inch cartridges score high on every basis of comparison, both economic and functional. They offer a unique combination of low media costs, high capacities, adequate transfer rates, compact size, and virtually foolproof operation. And they are available in a variety of tape capacities. Industry-standard 4-inch by 6-inch cartridges with 450 feet of tape can store up to 45 Mbytes of data in a streaming format. With a 600-foot cartridge, capacities of 60 Mbytes are possible.

Figure 5-3 Standard ¼-inch tape cartridge.

Figure 5-4 shows the internal construction of the standard ¼-inch cartridge and its interface with drive components such as tape-hole sensors, file-protect "safe" switch, drive motor and write/read heads. Not shown is the protective door which covers the exposed tape when the cartridge is removed from the drive. Figure 5-5 illustrates how the door automatically opens as the cartridge is inserted. Figure 5-6 indicates the safe and unsafe positions of the write-protection plug.

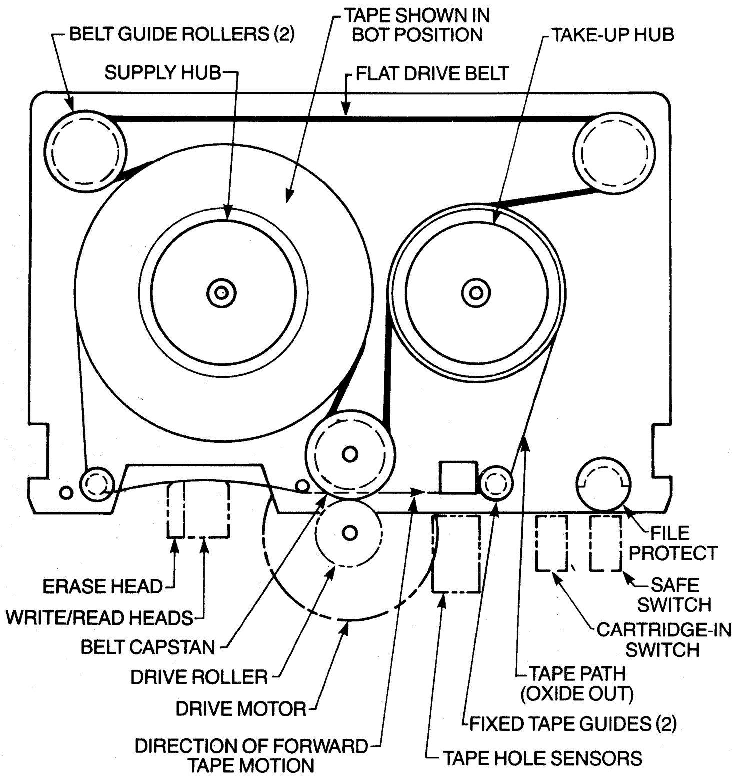


Figure 5-4 Internal construction, ¼-inch tape cartridge.

Tape motion is controlled by an external drive-motor to cartridge-capstan arrangement. The single capstan drives an isoelastic belt that in turn drives the tape itself. Tape speed is independent of both the direction of movement and the current tape position.

The standard ¼-inch cartridge tape has holes to mark beginning of tape, end of tape, a "load point" for recording data at the start of the tape and an "early warning" near the end. (The latter two reverse functions when the tape is recorded in a reverse

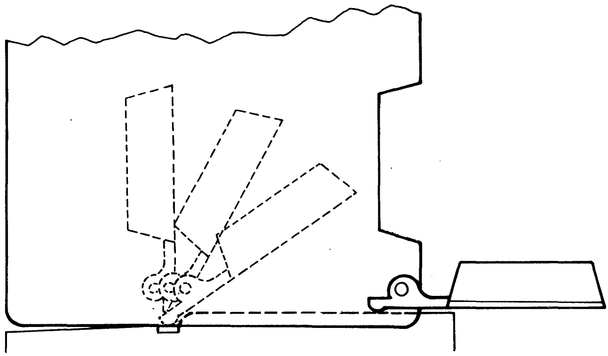


Figure 5-5 Protective door swinging open as cartridge is inserted.

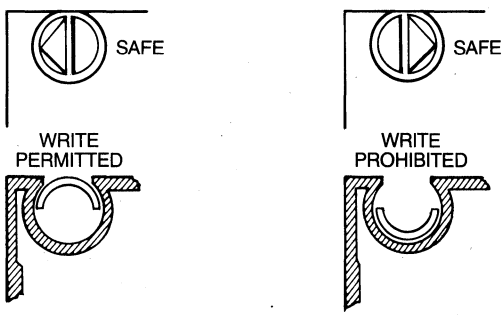


Figure 5-6 Safe and unsafe positions of write-protect plug.

direction.) Figure 5-7 shows the location of these holes and how they are "coded" on two levels. Light from an external LED is reflected by a mirror in the cartridge and detected by two tape-hole sensors.

PARALLEL TRACKS

Figure 5-7 also indicates how data is recorded in parallel tracks extending from just beyond the load and early-warning points. The four tracks are typical of many of the systems now in use. The way the tracks are recorded, however, has varied greatly from one type of system to another.

The original method—still widely used—was to record the data in one direction only, with a rewind between tracks (Figure 5-8). Performance was limited by a 300-foot tape length, PE encoding, and a 3200-frpi flux reversal density. Unformatted capacity was only 2.7 Mbytes. With 1.2-inch gaps between

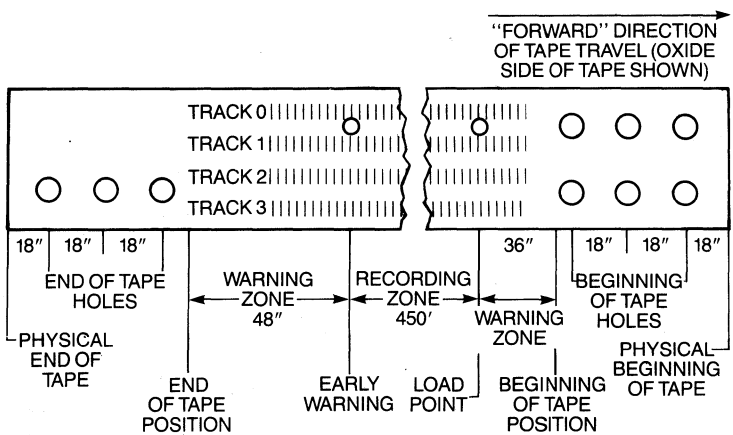


Figure 5-7 Tape-position holes, ¼-inch cartridge tape.

blocks to facilitate start-stop operations and a tape speed of 30 inches per second, transfer rates were held to 3 Kbytes per second.

An eight-fold improvement in transfer rates, up to 24 Kbytes per second, was achieved by FM encoding and parallel recording of 4-bit nibbles (Figure 5-9). But the multiple read/write heads were expensive, and precise alignment of tape and heads had to be maintained. Most start-stop cartridge systems have therefore returned to the sequential track-recording technique

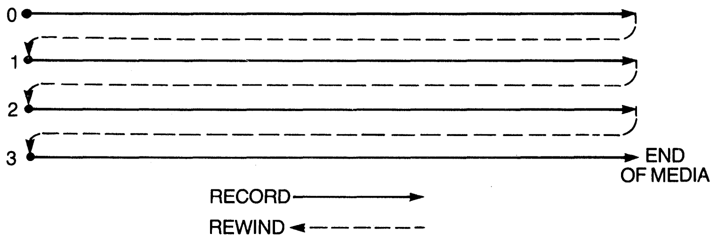


Figure 5-8 Four-track, forward-only recording.

accompanied by significant increases in capacity. Flux-reversal density has been raised to 6,400 frpi; MFM encoding matches bit density to the number of flux reversals; and longer blocks have reduced the number of inter-block gaps (still set at 1.2 inches).

SERPENTINE TRACKS

"Serpentine" tracks (Figure 5-10) take much fuller advantage of the cartridge's bidirectional capabilities. Instead of rewinding the tape at the end of each track, the write-read heads are logically or mechanically switched to a different position on the tape and another track is written or read in the reverse direction. The process continues until, in the case of a four-track tape, the heads have returned to the physical beginning of tape.

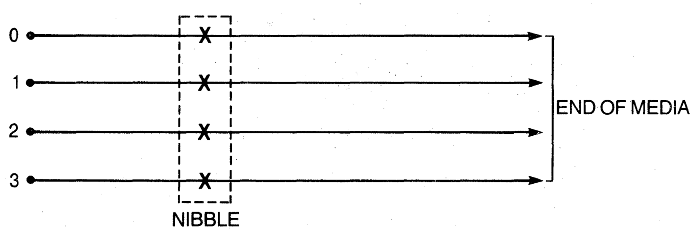


Figure 5-9 Four-track parallel recording.

Streaming systems, designed for almost continuous tape motion, have universally adopted the serpentine track format to avoid time-consuming rewinds. With the high frpi's and efficient coding described later in the next chapter, up to 5 Mbytes can be stored on each track for a total of 20 Mbytes for a four-track cartridge. Nine serpentine tracks (Figure 5-11) increase the capacity to 45 Mbytes, using industry-standard cartridges from

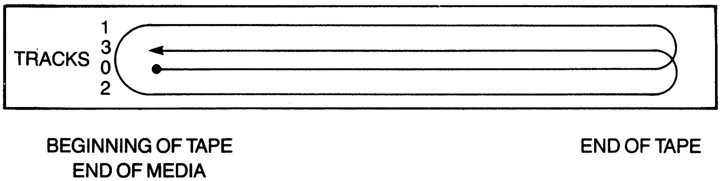


Figure 5-10 Four-track serpentine recording.

any of several suppliers. (A 16-track, 67-Mbyte system requires a factory-formatted 600-foot cartridge to achieve the higher track density.)

STREAMING DRIVE

Figure 5-12 illustrates the rugged simplicity of a streaming-cartridge drive. Other than safety switches and sensors, there

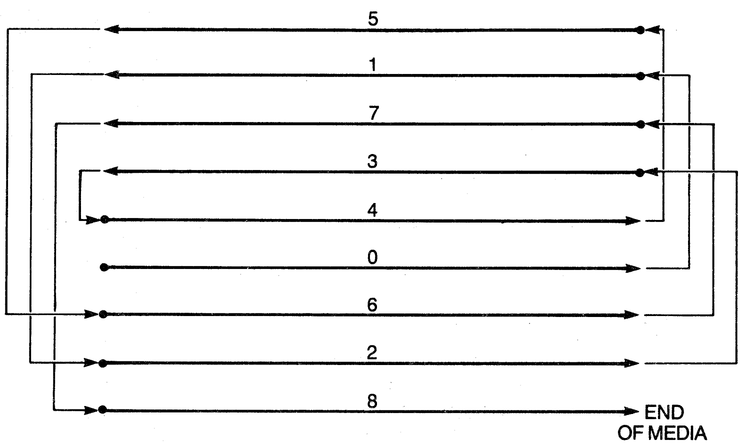


Figure 5-11 Nine-track serpentine recording.

are only two electromechanical assemblies: (1) a capstan motor with its associated drive circuits mounted on a board below the molded-polycarbonate frame, and (2) a head carriage assembly actuated by a stepper motor. (The balance of the electronics is mounted on a board above the cartridge.)

Two sets of write-read heads are required to trace a serpentine, bidirectional track pattern and simultaneously provide a read-after-write error detection capability. Figure 5-13 shows how dual write and read heads can be electronically switched to a second track level when the tape is reversed, reducing by half the number of head repositions.

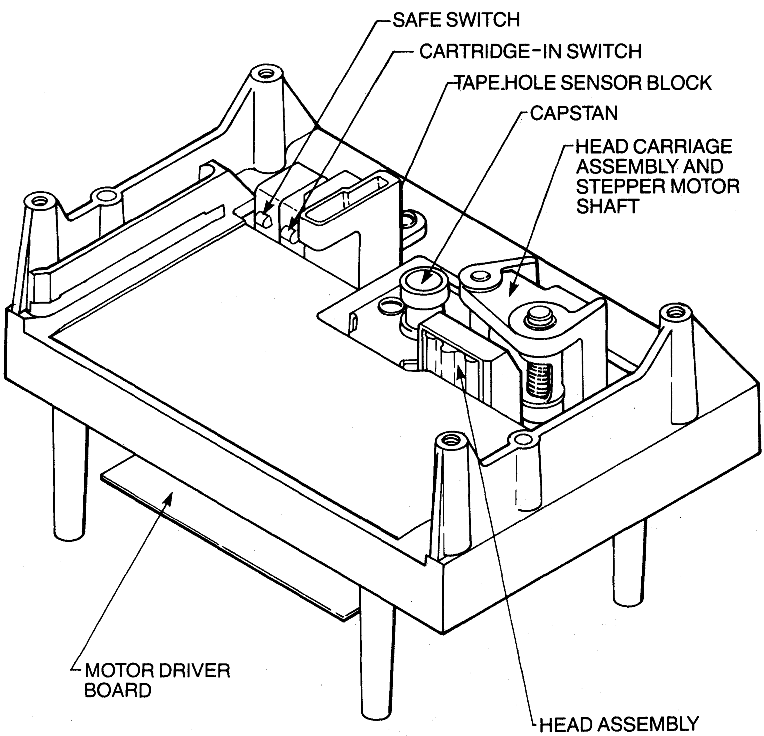


Figure 5-12 Archive streaming-cartridge drive subassemblies.

Unsaturated recording (Chapter 1) allows up to 10,000 flux reversals per inch to be written on conventional cartridge tape. To return the magnetic coating to a neutral state, a single tape-wide AC-erase head is automatically switched on whenever data is written on the first track, clearing the entire tape of all previously recorded information.

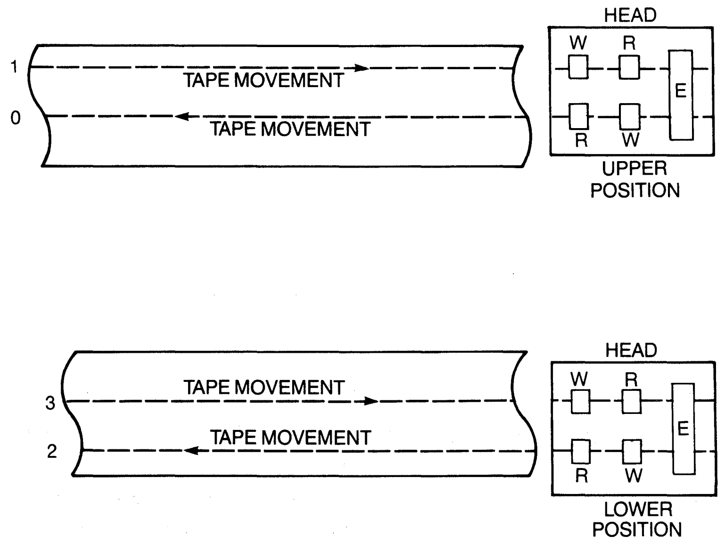


Figure 5-13 Dual-head positions for 4-track recording.

To assure media interchangeability from one drive to another, the head-positioning stepper motor and control circuitry are "recalibrated" each time a drive is powered up or reset. Calibration is accomplished by retracting the head assembly until a factory-set collar on the positioning lead screw reaches a mechanical stop (see Figure 5-14). The head is then raised a fixed number of stepper-motor increments until it reaches the correct level for recording or reading track 0. Subsequent adjustments in the head position are made against this benchmark.

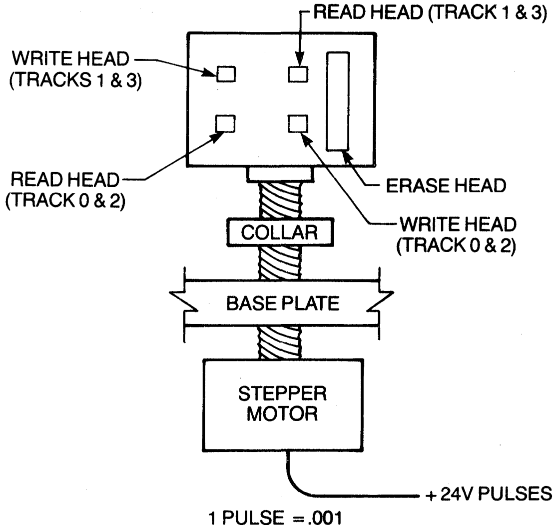


Figure 5-14 Head-positioning mechanism.

STREAMING INTELLIGENCE

The electro-mechanical components of a streaming-cartridge drive are designed to perform just three functions. Their first task is to control the motion of the magnetic medium as it passes under the device's write-read heads. Their second and third functions are to "write" flux reversals on the medium and "read" flux reversals that have been previously recorded. The significance of the magnetic record is established by the intelligence built into the drive.

Streaming-tape intelligence has several distinguishing features. Streaming-tape throughput is based, for example, on continuous tape motion which requires, in turn, an uninterrupted, precisely coordinated flow of data to and from the streaming device. Organization of the recorded data should also enhance tape's potential as an extremely low cost storage medium (iron oxide on a plastic substrate, produced in a continuous roll-coating process). Finally, as an archival record, streaming-tape data should be as error-free as technology will allow.

Intelligence to achieve these streaming-tape objectives is now at an evolutionary stage, with streaming-cartridge devices as the focal point of the development efforts. Streaming ½-inch, reel-to-reel tape has a longer history, but the need for compatibility with a large population of start-stop ½-inch drives has limited the inventiveness of designers. In effect, streaming cartridges are starting where the older technology has left off.

EFFICIENT ENCODING

An example of this fact is the almost universal use of NRZI run-length limited encoding—similar to that of the highest-performance ½-inch tape systems—as the most efficient way to take advantage of the high flux-reversal densities produced by such other innovations as AC erasing followed by unsaturated recording.

In the case of the Archive intelligent drive which will serve as our model for this discussion, the combination produces a linear data density of 8,000 data bits or 1,000 data bytes per inch. (NRZI encoding of 10,000 flux reversals per inch results in 10,000 recorded bits, but the 4-to-5 run-length code conversion reduces this figure to 8,000 bits of data actually recorded.)

One fortunate byproduct of this data density is an easy-to-calculate relationship between stored data and linear tape length. Each byte occupies a thousandth of an inch along the tape. The length of a 512-byte block is therefore approximately a half inch.

MAXIMUM TAPE USAGE

Increased data densities mean that more information can be stored on a cartridge of a given tape length, reducing the per-megabyte media cost, increasing the amount of attached backup storage, and minimizing the need for frequent media exchanges when large amounts of data are recorded or read back onto disk. Higher data densities also increase the write and read transfer rates for a given tape speed, reducing the time required to make disk-to-tape and tape-to-disk transfers and potentially reducing the requirement for buffer storage between disk and tape.

	STREAMING	START-STOP		
Data Block	512	512	1024	2048
Preamble	—	5	5	5
Postamble	—	5	5	5
Block Address	1	—	—	—
Sync Mark	0.5	—	—	—
CRC	2	2	2	2
Gap	13	960	960	960
Total/Block	528.5	1,484	1,996	3,020
Tape Usage	97%	35%	51%	68%

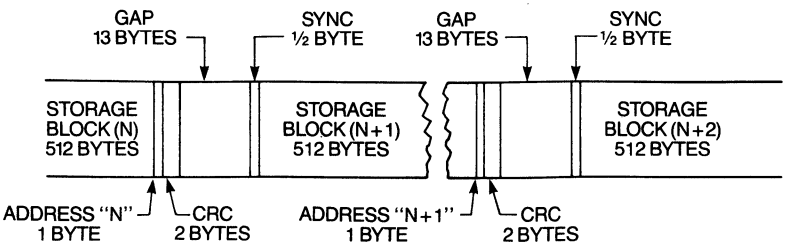
Figure 5-15 Tape usage, streaming versus start-stop.

All of these benefits also result from a more efficient format that reduces "wasted" tape between data blocks.

Data is organized into blocks for two primary reasons: to facilitate the search for a particular record or file, and to isolate write or read errors to specific sections of tape which can be individually verified and corrected. A third reason would be to provide convenient areas for starting and stopping the tape at selected locations along the tape.

Both of the primary objectives can be fulfilled by very short "gaps" between blocks if it is assumed that (1) the system will rarely need to access records stored in an individual block, and (2) error processing can be effectively performed "on the fly" or, alternatively, by trading off a relatively inefficient but infrequent error-processing procedure for the repetitive advantage of a short inter-block gap—too short for the tape to start or stop within the gap.

Start-stop tape formats, ½-inch and cartridge, make the opposite assumptions. The start-stop format was developed at a time when tape was the only available medium for storing large volumes of transactional and historical data in a form that could be "randomly accessed" for reference with minimum operator



CODES	NON-DATA RUN-LENGTH CODE
GAP	11111
SYNC	00111
FILE MARK	11100

CRC POLYNOMINAL: $X^{16} + X^{12} + X^{5} + 1$

Figure 5-16 Streaming-cartridge tape format.

intervention. Both drive and tape were optimized for frequent reversals and advances. Unless instructed otherwise, the tape stopped at the end of each block and waited for the next command. Inter-block gaps had to be long enough to allow the tape to decelerate to a stop and reaccelerate before the start of the next block.

All of these characteristics have been carried forward to the present-day tape cartridge when recorded or read in a start-stop mode. Figure 5-15 compares the industry-standard start-stop format with the streaming-cartridge format shown in Figure 5-16. The start-stop gap is set at 1.2 inches, equivalent to 960 bytes at a data density of 800 bytes per inch. The size of the start-stop data block is user selectable, but even with a very long block (increasing the odds that an error will occur and the block must be rewritten or reread), tape usage is far below the 97% achieved by the streaming format.

STREAMING FORMAT

"Bytes" shown in Figure 5-16 are 10-bit units that match the size of data bytes after a 4-to-5 run-length conversion of half-byte nibbles. Thus the short, 13-byte gap is recorded (not erased, as in the case of a start-stop gap) as 130 ONE bits, an illegal run-length limited code. Illegal codes are also used for "file marks" and the 5-bit sync mark that signals the start of user data. A file mark is a full block of 512 non-data bytes that identifies, for example, the start or end of a user file. It can also be used to divide a file into smaller segments to facilitate the search for a particular record.

CRC characters and "address" are converted 8-bit bytes. Both play a part in the error detection and correcton procedures described below. The block address sequence, repeated every 256 blocks, also helps to guarantee that blocks transmitted to the host or disk are in the correct order.

ERROR CORRECTION

Write or read errors are inevitable. They may be caused by faults in the magnetic coating, by contamination, by slippage as the tape is unreeled within the cartridge, or simply by the wear that occurs with repeated passage over the write-read heads.

Most of the errors are "dropouts"—a loss of data as the information is recorded or recovered. Less frequently the error will be a "drop-in," a spurious flux reversal that throws off the precise syncronization required to separate 4096 data bits into the 512 bytes contained in each block.

Even a cartridge that has been factory-qualified for digital recording is likely to have five to ten blocks in error (BIE's) among the 10,000 recorded along each 450-foot track. The challenge is to efficiently detect each of these occurrences, correct the error, or report the error if it is uncorrectable.

Error detection is straightforward. Nearly every magnetic-recording format, disk or tape, includes CRC characters that indicate, to a high degree of confidence, whether a block or sector has been correctly recorded or read (see Chapter 1).

By comparison, error correction can take a variety of forms, each representing a different tradeoff between the frequency of hard errors and such cost-performance parameters as storage capacity, transfer rate, efficient useage of the media, and processing time on the part of the host, formatter-controller, or the device itself.

One approach is to add an error-correction code (ECC) to each block or sector. The code may consist, for example, of parity bits for each of a number of different bit combinations within the block, allowing the system to identify and correct a specific bit in error. The size of the correction code and the complexity of the calculation increase, of course, when multiple error bits must be identified.

ECC proves to be a practical way to correct Winchester-disk errors. The factory-prepared recording surface and contamination-free environment limits typical disk dropout errors to a few bits. But as shown in Figure 5-17, an equivalent dropout in data recorded on cartridge tape extends over several thousandths of an inch and would require the reconstruction of well over a hundred bits.

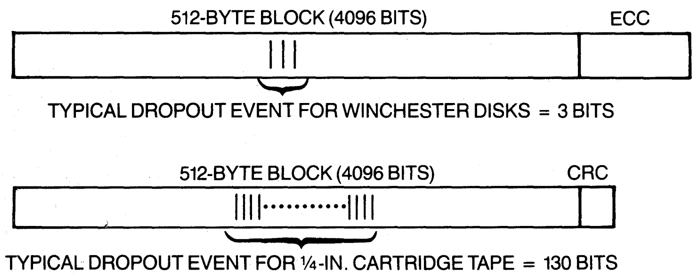


Figure 5-17 ECC (error-correcting code) versus CRC (cyclic redundancy check).

Any effort to use ECC correction becomes, in this case, cumbersome. One commercially available streaming-cartridge system, for example, dedicates a full third of its potential data-storage capacity to the correction task. Corresponding bits in two contiguous sub-blocks are exclusive-ORed to form a third sub-block. Each data bit is recorded, in effect, one and a half times.

A different set of tradeoffs applies to the error-correction technique shown in Figure 5-18 and described more fully in Chapter 9. Tape is dedicated to rewritten blocks only when

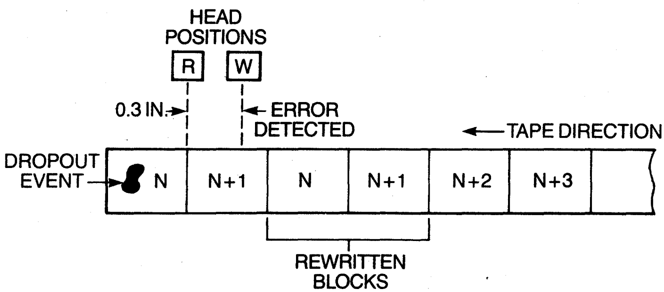


Figure 5-18 Correction by rewriting BIE (block in error).

errors have actually occurred during a write operation. The tradeoff during a read operation is a reduction in the tape-to-host transfer rate.

A read-after-write check for errors is performed as each block is recorded. The physical spacing between write and read heads (combined with the short inter-block gap shown in Figure 5-16) means that the next block will have been partially written before the CRC characters of the previous block have been read and verified. If the read-check indicates that a write error has occurred, block N+1 is completed and then the BIE (block N) is rewritten. To maintain sequence, the following block (N+1) is also rewritten. If the rewritten BIE still contains an error, the process is repeated. A hard error is reported if, after 16 attempts, block N continues to be in error.

Block addresses identify the rewritten blocks during subsequent read operations. Only verified, sequential blocks are transmitted to host or disk. An incorrectly written block N would not meet the first test; the following block address, N+1, would break the sequence. No transfers occur until a correct block N has been read.

An identical procedure protects against read errors. Again an unverified block N would inhibit the transfer of blocks N and N+1. If the next block address is N+2 instead of N, block N was not rewritten during the write operation. The tape is stopped, backed up, and reread. Up to 16 read attempts can occur before a hard read error is reported. Hard or soft, read errors interrupt the continuous motion of the tape, reducing the transfer rate.

The error-correction technique shown in Figure 5-18 does require, however, buffering at the device level. The previously recorded block must be retained in memory until it is verified. The resulting three-block buffer (Figure 5-19) is a bonus during read operations, providing an extra margin of safety against "buffer underruns" that would otherwise stop the tape because host or disk can not accept the data at the full streaming rate. Chapter 7 discusses the requirement for additional buffering to maintain streaming.

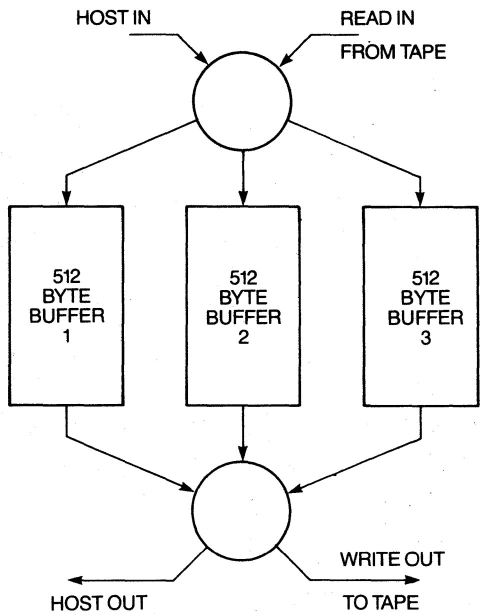


Figure 5-19 Three-block buffer for write and read operations.

STATUS AND CONTROL

An intelligent streaming-cartridge drive should be able to perform all of the functions described in this chapter in response to a minimum number of host commands. It should also be able to report a maximum amount of status information so that the host can coordinate the data transfers and counteract any conditions that might reduce system throughput. And all of this should be accomplished across a simple, easy-to-implement status and control interface.

Figure 5-20 summarizes the host interface for the intelligent drive shown in Figure 5-21. All of the lines, including the byte-wide bus, are negative true. With -ONLINE true, -TRANSFER and -ACKNOWLEDGE handshake byte-by-byte data transfers during read and write operations. -REQUEST and -READY form a handshake for transfer of command and status bytes. -DIRECTION indicates whether the drive is ready to receive or transmit information; -EXCEPTION notifies the host that a condition exists which requires its attention;-RESET initializes all of the drives connected to the system.

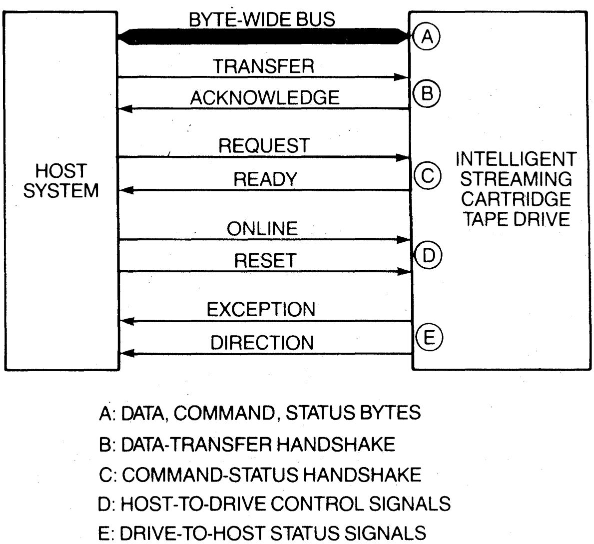


Figure 5-20 Interface signals, intelligent drive.

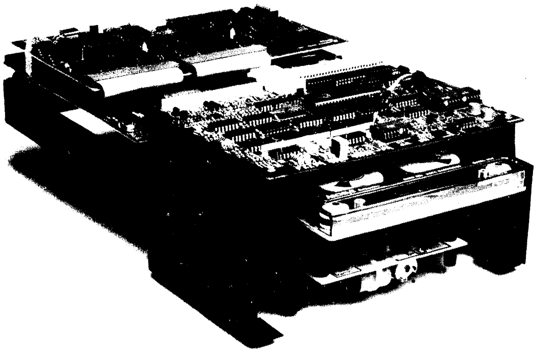


Figure 5-21 Intelligent streaming-cartridge drive.

All of the streaming-drive functions are initiated by single-byte host commands:

SELECT identifies the drive to be controlled or accessed.

BEGINNING OF TAPE, RETENSION, and ERASE are tape-positioning commands that end with the cartridge rewound to beginning of tape.

WRITE initiates a transfer of data to tape.

WRITE FILE MARK directs the drive to generate and record a File Mark block.

READ indicates that data is to be read from tape.

READ FILE MARK tells the drive to advance to the next File Mark without transferring any data.

READ STATUS initiates the transfer of status bytes maintained by the intelligent-drive microprocessor.

Six bytes are returned to the host in response to the READ STATUS command. Two provide single-bit status indications, such as the lack of a cartridge or the receipt of an illegal command. The remaining bytes indicate the number of rewrites or read errors and buffer underruns.

CHAPTER 6

Streaming Software

Specialized software for systems with streaming-tape storage can be generally divided into four categories:

Streaming-cartridge driver or handler

Mass-storage utilities

Application programming

Diagnostics

The division between these categories will depend, in turn, on the system configuration and whether an off-line data channel is provided between the streaming-tape device and its associated Winchester disk (see Chapter 8, Disk-Tape Storage).

HOST ADAPTER

Drivers and utilities, in particular, are affected by the sophistication of the host adapter. Using the intelligent streaming-cartridge drive described in the previous chapter as an example, it can be assumed that in most applications the host can readily generate the -ONLINE, -REQUEST, and -RESET interface signals by a programmed-I/O output to register latches and drivers incorporated into the interface adapter. Similarly, device-generated signals such as -READY and -EXCEPTION can be relayed to the host through a programmed-I/O input.

The host adapter should also include implementation of -READY and -EXCEPTION interrupts to avoid prolonged tie-ups while the host computer is waiting for the device to complete a command or the device itself has encountered an -EXCEPTION condition.

Whenever possible, a DMA channel should be used to transfer the read and write data to insure an adequate throughput rate for continuous streaming. The principal function of the -DIRECTION status signal is to enable the host's bus drivers during DMA write operations.

LOWER LEVEL CALLS

Command transfers can be easily accomplished by loading the command into a register connected through drivers to the bidirectional bus. The required control-signal protocol could then be executed by the device driver. Status input and the reading and writing of files can be similarly implemented. The flow diagrams included in this chapter are intended as a guide to creating a set of lower level calls for these purposes.

Figure 6-1 shows the procedure for sending a command across the data bus. As described in Chapter 9, either -READY or -EXCEPTION must be set by the streaming-cartridge device before a command can be transmitted. If -EXCEPTION is set, READ STATUS is the only valid command that will be accepted by the device.

Status information can be requested at any time. Figure 6-2 indicates how the READ STATUS command initiates the transfer of six status bytes maintained by the drive microprocessor.

The host tests whether an operation has been completed by polling the device's -READY and -EXCEPTION lines (Figure 6-3). The former indicates a successful completion; the latter could mean that an illegal command had been transmitted, an error had occurred, or the operation was aborted.

A system reset is accomplished by asserting -RESET for a minimum of 13 microseconds (Figure 6-4).

Six commands (Figures 6-5 through 6-7) require no further actions on the part of the host computer. Single-byte commands

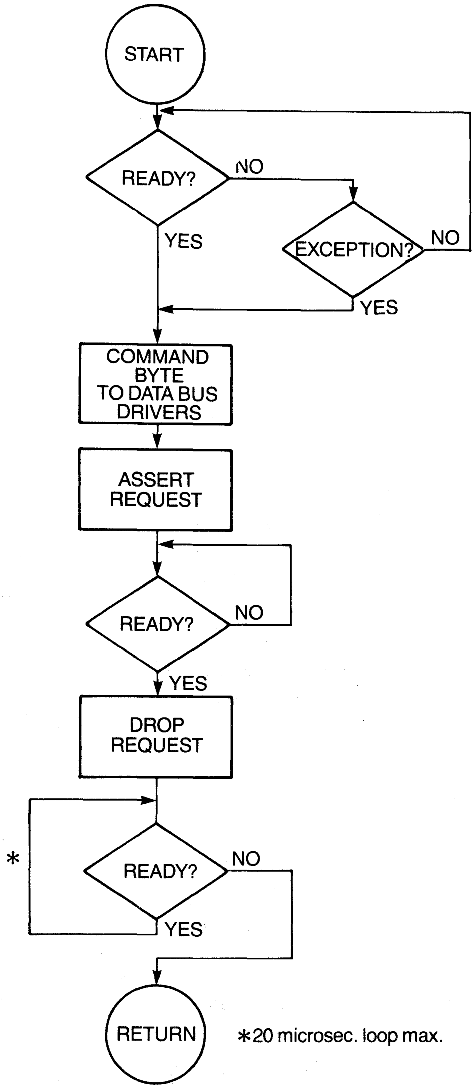


Figure 6-1 Flow diagram, send command.

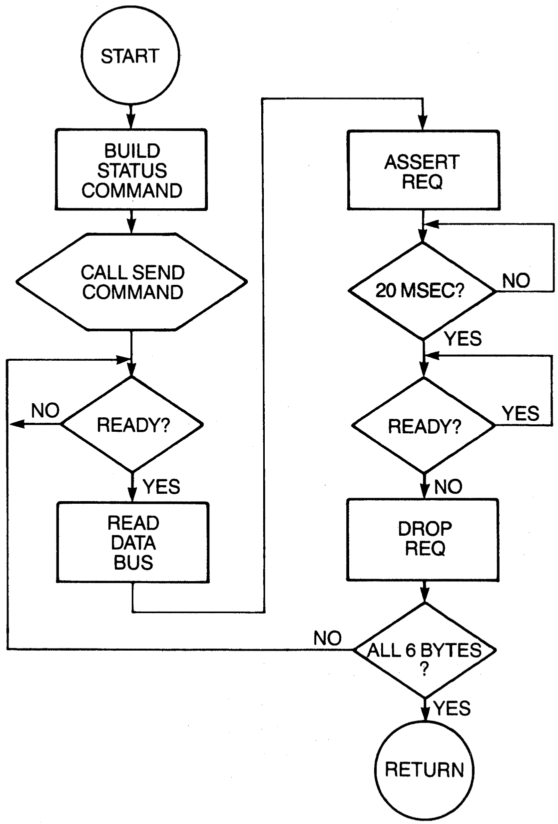


Figure 6-2 Flow diagram, read status.

direct the device to select a drive in a multi-drive system, rewind the cartridge to beginning of tape, retension or erase the tape, write a File Mark, or advance to the next File Mark.

A file is written (Figure 6-8) by transmitting a 512-byte data block and testing -READY before the next block is sent. After all the data is written, a File Mark is recorded to indicate the end of the file.

Data is appended to previously recorded files by reading successive File Marks until a "no data" condition is encountered (Figure 6-9). Write File is then called.

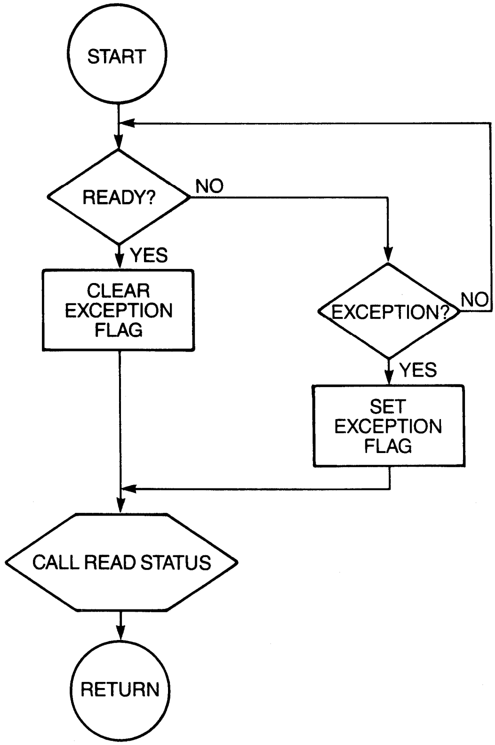


Figure 6-3 Flow diagram, host done.

A file is read by again transferring 512-byte blocks and testing -READY before each block transfer (Figure 6-10). A specific file is read by advancing from one File Mark to the next (Figure 6-11), counting the files and issuing a READ command when the tape has advanced to the desired file.

Specific files can also be accessed by recording a unique identification code in the first block following each File Mark. The host builds an identification block in memory (Figure 6-12) and compares this with the first block in successive files. If there is a match, the file is read. If not, the device is instructed to advance to the next File Mark without transmitting any data.

DISK-TAPE DRIVERS

The software driver that generates streaming-cartridge commands and supervises the storage system operations can take a variety of forms, depending on the computer, operating system, and objectives of the system programmer.

The driver can, for example, completely supplant the existing fixed or flexible disk driver incorporated into the operating system. Or it could be "attached" to the established storage-device driver software, adding to the capabilities of the system without reducing the hardware flexibility.

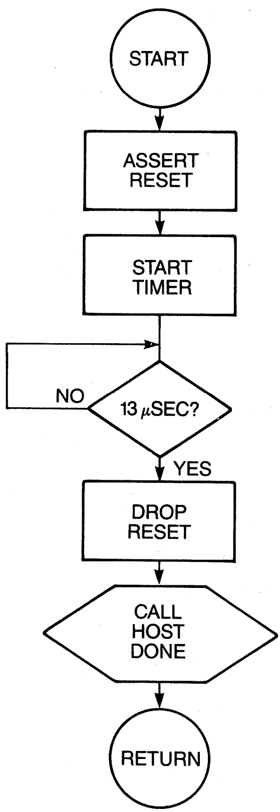


Figure 6-4 Flow diagram, reset drive.

Both approaches have been taken by commercially available software packages. In one case, applicable to systems running under CP/M 2.2, customized software is created by integrating the user's existing console, printer, and flexible disk drivers into a skeletal BIOS. The BIOS supports both cold and warm boot operations. Conditional assembly switches allow the user to configure the storage-system segments to the known capacity of the disk drive (e.g., 20, 40, or 80 megabytes). Logical 8-megabyte disks are defined by the driver. The user must also reconfigure computer memory to allow for the expanded system software.

DISK-TAPE UTILITIES

Streaming-cartridge software could also include backup-and-restore software that allows the user to transfer logical disks and selective files to and from the streaming-cartridge drive.

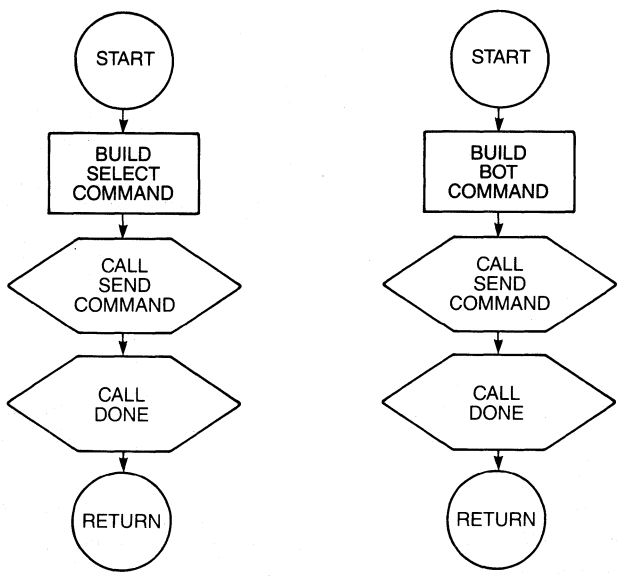


Figure 6-5 Flow diagrams, SELECT and BEGINNING OF TAPE commands.

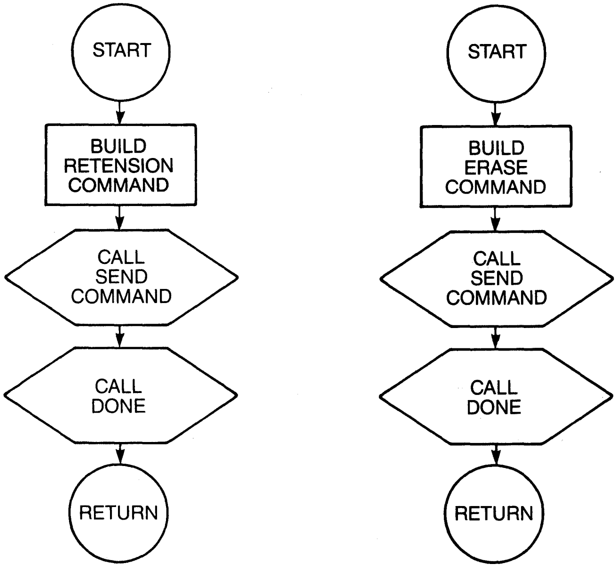


Figure 6-6 Flow diagrams, RETENSION and ERASE commands.

Typically, the utility would be a menu-driven program prompting the user at each decision point with appropriate questions and valid answers. On entering the program, for example, the user would be given the following choices to select the type of operation:

BACKUP

RESTORE

VERIFY

STATUS UPDATE

The first option allows transfer from disk to tape, while the second reverses the process. During a backup or restore operation the user can use the status update option to read the status of the operation. Verify performs a tape-disk comparison assuring data integrity.

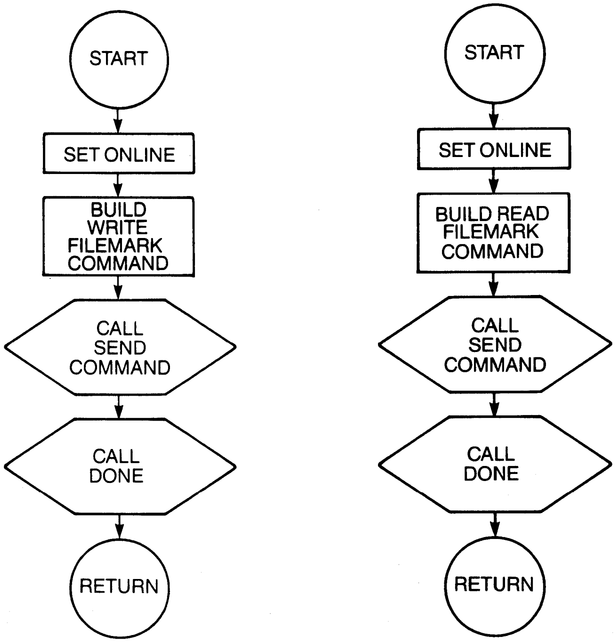


Figure 6-7 Flow diagrams, WRITE FILE MARK and READ FILE MARK commands.

With the selection of the three data transfer options, the program might respond with:

ERASE TAPE

REWIND TAPE

RETENTION TAPE

NONE OF THE ABOVE

If a restore or verify command was initially selected, the next option presented to the user could be:

ADVANCE TO A NEW FILE ?

ENTER Y or N

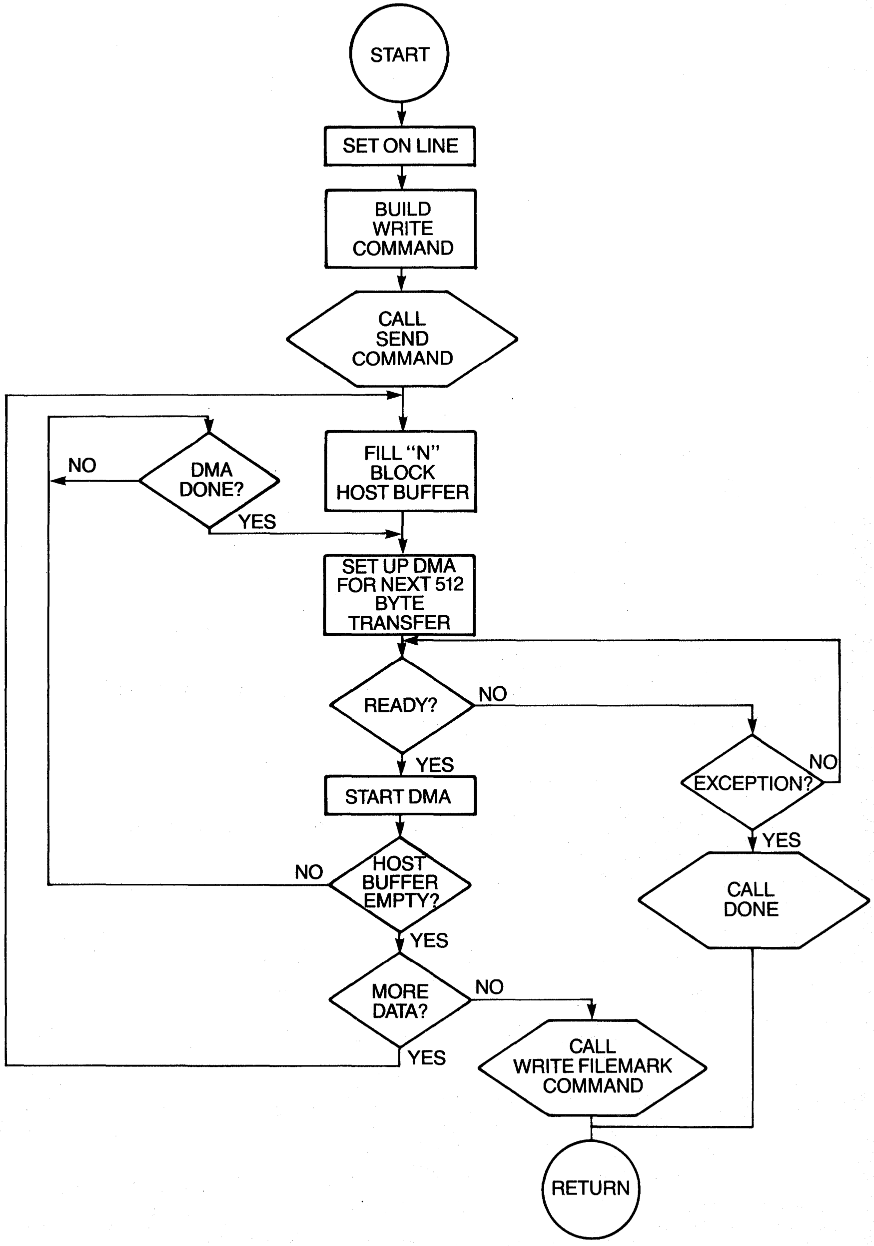


Figure 6-8 Flow diagram, write file.

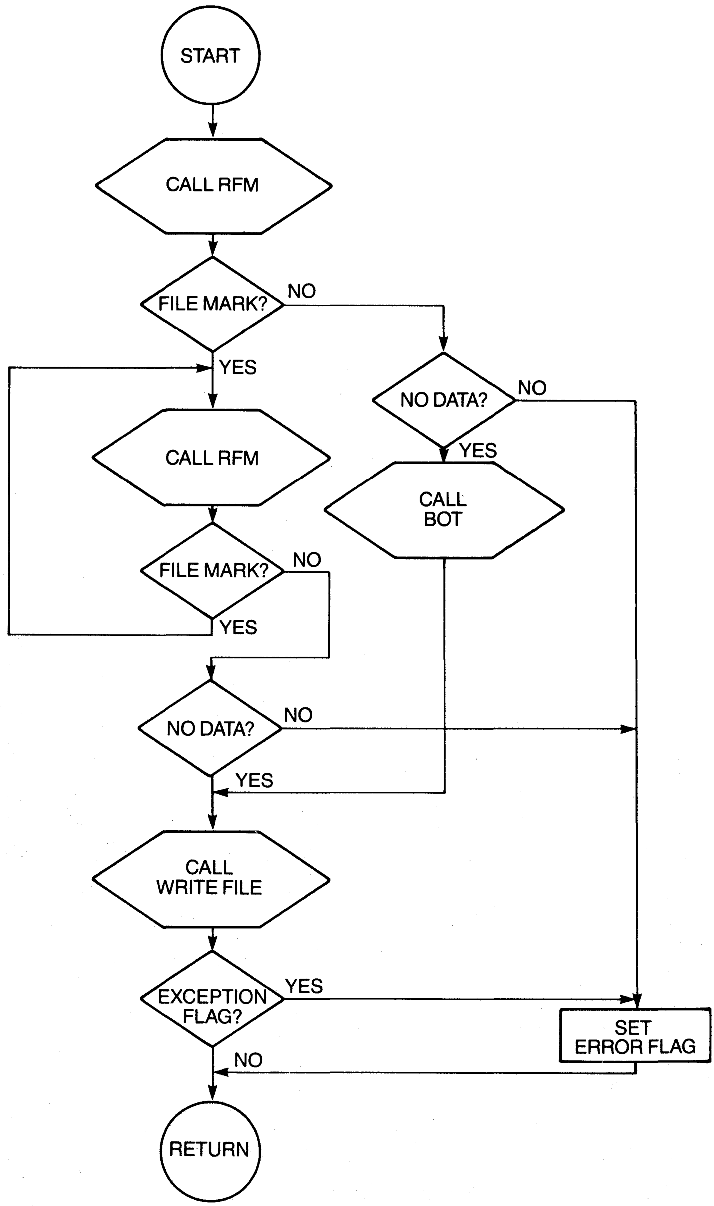


Figure 6-9 Flow diagram, append a file.

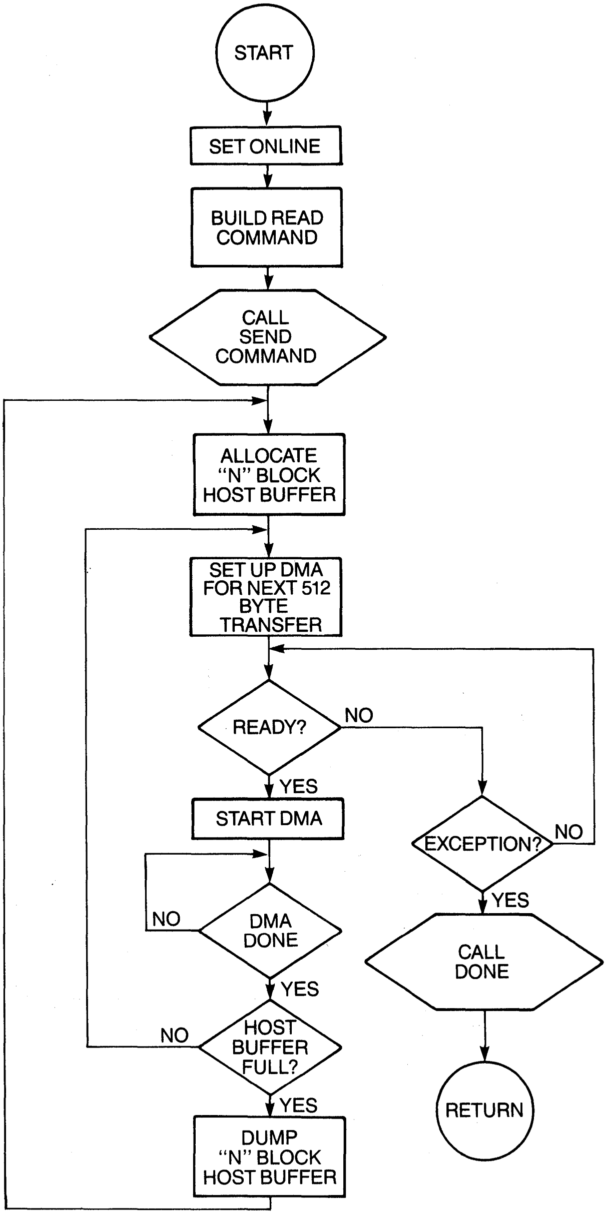


Figure 6-10 Flow diagram, read file.

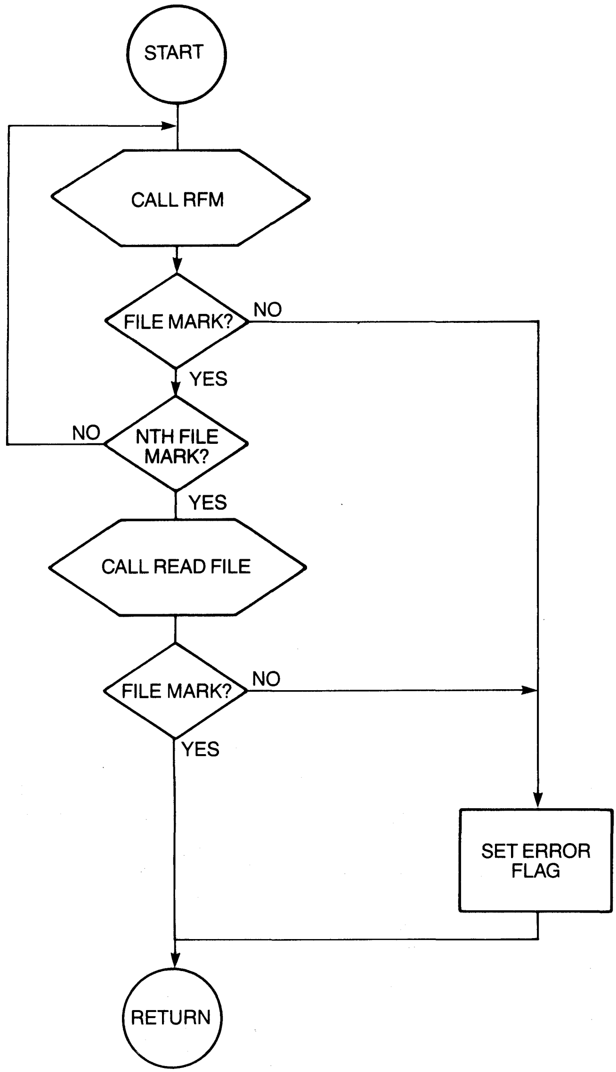


Figure 6-11 Flow diagram, read Nth file.

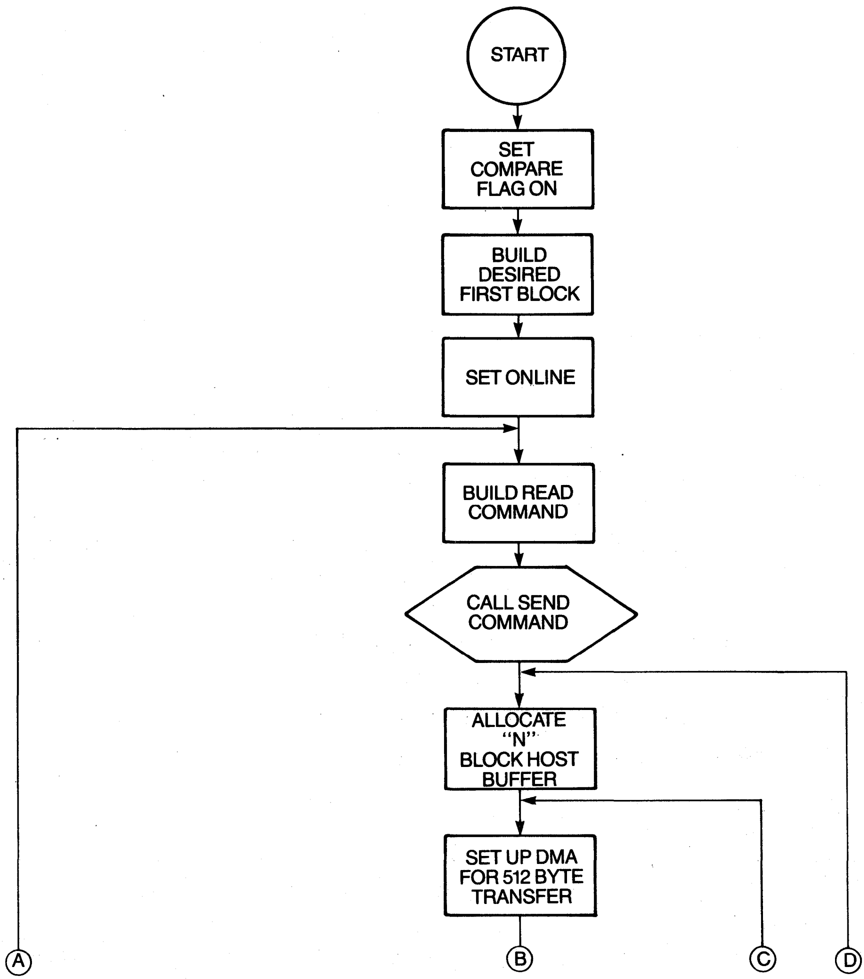


Figure 6-12 Flow diagram, read file with first-block ID.

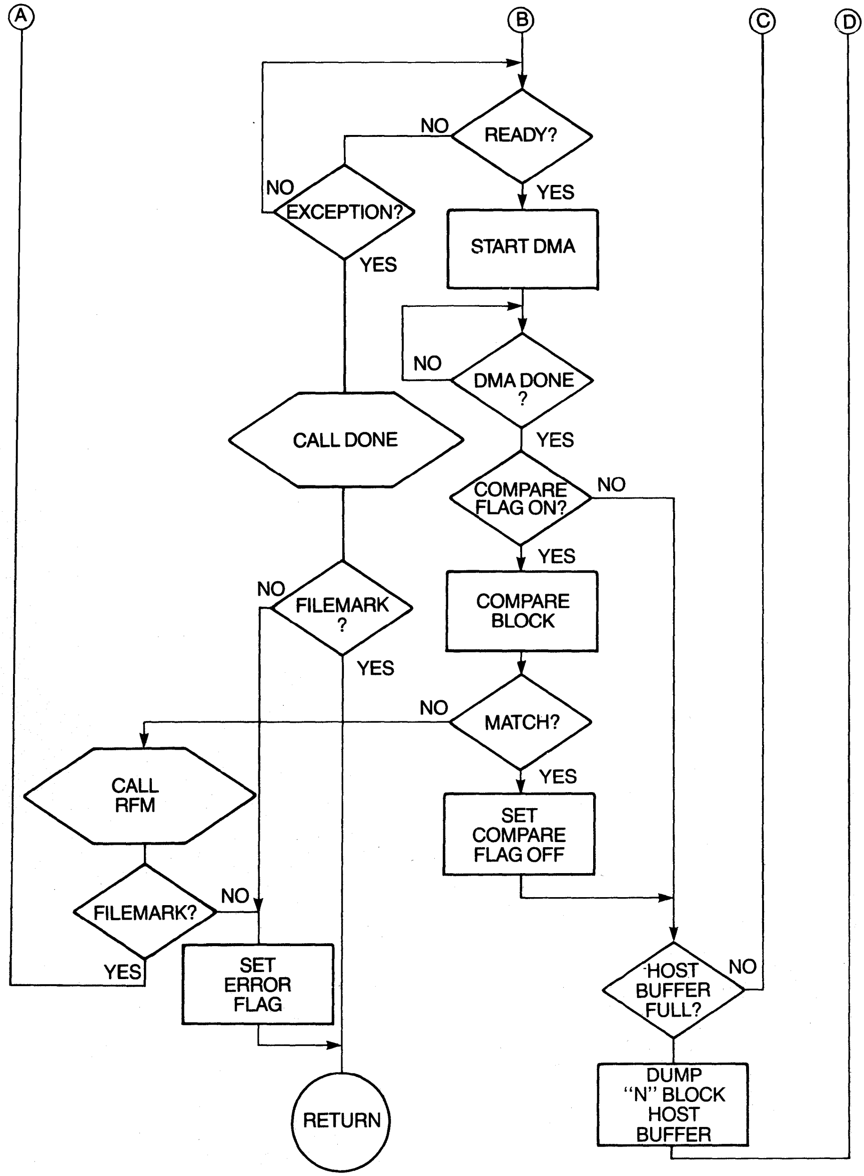


Figure 6-12(con.) Flow diagram, read file with first-block ID.

A yes response might result in the following question:

NUMBER OF FILES TO ADVANCE FROM CURRENT
TAPE POSITION (1 to 9)

ENTER 1 TO 9

Once the entry is made, the tape would begin to move forward looking for File Marks. If a 1 was entered, the tape would stop at the first File Mark and the next request would be for a logical disk number.

Additional prompts and responses could allow the user to select a specific logical disk, an on-line or off-line mode, and the level of priority for off-line transfers.

CHAPTER 7

System Considerations

Streaming-cartridge drives are storage-system components which must communicate efficiently with the host computer and, directly or indirectly, with one or more disk units. System considerations include, therefore, not only streaming-cartridge characteristics, but also the capabilities of the host and disk drives.

DEVICE CAPACITIES

Of major importance, for example, are the absolute and relative capacities of the disk and streaming-tape devices. Chapter 2 considered this variable as it applies to Winchester disk storage. There is no limitation, of course, on the amount of detached, archival streaming-tape storage—except for media costs. The question, then, is the appropriate amount of attached tape capacity.

The decision is highly dependent on the application. In multiuser environments, for example, designers may decide to take advantage of the per-megabyte savings of a high-capacity disk shared by all of the users. It would seem logical, then, to back up the disk with a high-capacity cartridge drive. The complete contents of the disk could be unloaded without a media change.

In many instances, however, it may be important for users to have physical possession of their data and application programs. One or more smaller-capacity cartridge drives would provide this capability.

WINCHESTER DISK PARTITIONING

In many applications the Winchester disk is partitioned into a number of logical units. A logical unit can be defined as an area of disk space (cylinders and tracks) which is recognized and treated by system software and/or hardware as one device. As an example, CP/M operating systems typically define the capacity of a disk unit to be less than 10 Mbytes. To meet this requirement, a physical 40-Mbyte Winchester disk can be divided into several logical units, each appearing as a unique disk to the operating system.

Logical partitioning of a relatively large Winchester disk provides for efficient use of both disk and tape storage in a multiuser environment. One obvious advantage is the ability to assign logical units to various system users as they are requested. When a particular user wishes to go on-line, he merely loads his data base from streaming-cartridge tape onto his assigned logical disk unit. Once loaded, the programs can be worked on and transferred between logical disk units via standard operating system programs such as PIP (peripheral interchange program) or COPY.

When the user has completed his task and wishes to go offline, his data base is again unloaded to streaming-cartridge tape. The data base is now resident on tape, and the logical disk unit is available to other users.

The use of logical disk units thus allows tape cartridges to function as primary storage, reducing the size of the disk to a capacity that will support only the number of users who, on average, are on-line and actually accessing the disk.

Partitioning of the disk into logical units provides other advantages. Logical-disk capacity can be selected to match the streaming-cartridge drive capacity, simplifying the backup function. Logical disk units can also be used as "scratch pad" areas to stage segmented data prior to transfer to streaming tape.

TRANSFER RATES

Most of the other system considerations relate, directly or indirectly, to backup and restore transfer rates. Speed is critically important if the transfers occur in an interactive environment or involve host resources that are normally committed to other tasks. The transfers may not have to be as fast or efficient if their primary purpose is an end-of-the-day archival store or a start-of-the-day program load.

Three separate transfer rates are involved: host, streaming-cartridge drive, and Winchester disk. In all three cases the burst or maximum transfer rate is not significant. Instead, only operation dependent rates—often a fraction of the maximum rate—should be considered.

HOST TRANSFERS

In general, the host computer will be the fastest element in the system—if host resources are available for the transfers and no interruptions occur.

This is assuming, too, that a DMA channel can be used for at least the host-disk transfers and ideally for transfers to and from the tape device. The highest rate would be a burst DMA transfer that preempts all other host activities. More realistically the transfers would be in a cycle-stealing mode and subject to interruption by higher priority tasks.

System architecture may also limit the host transfer rate. In a bus-structured system in which system main memory is used for buffering disk and tape data, a large quantity of data at a relatively fast transfer rate must be moved over the system bus—requiring a bus structure which has adequate bandwidth to permit the maximum required transfer rate. Contention for the bus must also be considered because the data transfer must occur without lengthy interruptions. The architecture should allow all devices to access system resources as they require and still provide adequate data transfer to and from the tape drive to maintain streaming.

The host could also limit the overall transfer rate by restricting the size of the average "transmission"—an uninterrupted stream of data to or from the tape device. Transmission size has a non-linear effect on the average rate at which a streaming-cartridge drive can write or read data. If host memory is the principal buffer between disk and tape, the amount of memory allocated to the task will, in many instances, put a ceiling on the effective rate of the streaming-tape transfers.

TAPE TRANSFERS

The nominal transfer rate of a streaming-tape device is the tape speed multiplied by the linear data density. At 90 inches per second (ips), 97% tape utilization, and a recording density of 1,000 bytes per inch, the theoretical write-read rate is 87,300 bytes per second. With the speed reduced to 30 ips, the rate is 29,100 bytes per second.

But these figures are based on continuous tape motion. If the system is unable to supply or receive data for more than a few milliseconds (typically less than 6 milliseconds for a 90-ips drive, 17 milliseconds for a 30-ips drive), drive buffers will "underrun" and the tape will stop.

The high speed of a 90-ips drive becomes a liability under these conditions. As in the case of cars on the highway, the higher the speed, the longer it takes to come to a safe stop—and again regain momentum when motion is resumed. Moreover, streaming tape, with its short interblock gaps, is certain to overshoot its restart point, requiring another acceleration and deceleration as the tape backs up. Chapter 9 discusses in detail the timing associated with tape repositioning.

The following calculations show tape throughput—assuming no write-error rewrites, read-error retries, or track switching. The effective tape transfer rate is dependent on transmission block size, tape reposition time, data transfer time and host overhead time. For the following calculations it will be assumed that the host is ready to tranfer data before tape repositioning is complete and therefore host overhead will be zero.

$$\text{Tape Transfer Rate} = \frac{\text{Transmission Size}}{t_{transfer} + t_{reposition} + t_{host}}$$

where

Tape Transfer Rate	=	Data transfer rate in Kbytes per second.
Transmission Size	=	Amount of data in Kbytes written or read from tape without encountering a tape reposition.
$t_{transfer}$	=	Time in seconds to write or read the transmission block to/from the tape.
$t_{reposition}$	=	Time in seconds to completely reposition the tape, including the time to stop, back up and restart the tape.
t_{host}	=	Time in seconds when the tape drive is stopped and waiting for the host.

Example: 90-ips tape operation and 16 Kbyte transmission

$$\text{Tape Transfer Rate} = \frac{16}{32\,(5.872\text{x}10^{-3}) + 1.5 + 0} = \frac{16}{1.689}$$

$$= 9.47 \text{ Kbytes/sec}$$

Buffer underruns become far less costly—and 90-ips drives come to the fore—as the amount of data transferred without interruption increases toward 100 Kbytes. Figure 7-1 plots transmission size against the average transfer rate—assuming that for system or disk-related reasons, backup operations are intermittent.

The curves indicate that large transmission blocks are required for efficient streaming operation and even when streaming is achieved, system constraints on data throughput may limit the operation to 30 ips. Also, due to different data transfer rates and tape repositioning times, a 30 ips drive may outpace a 90 ips drive for small transmission sizes.

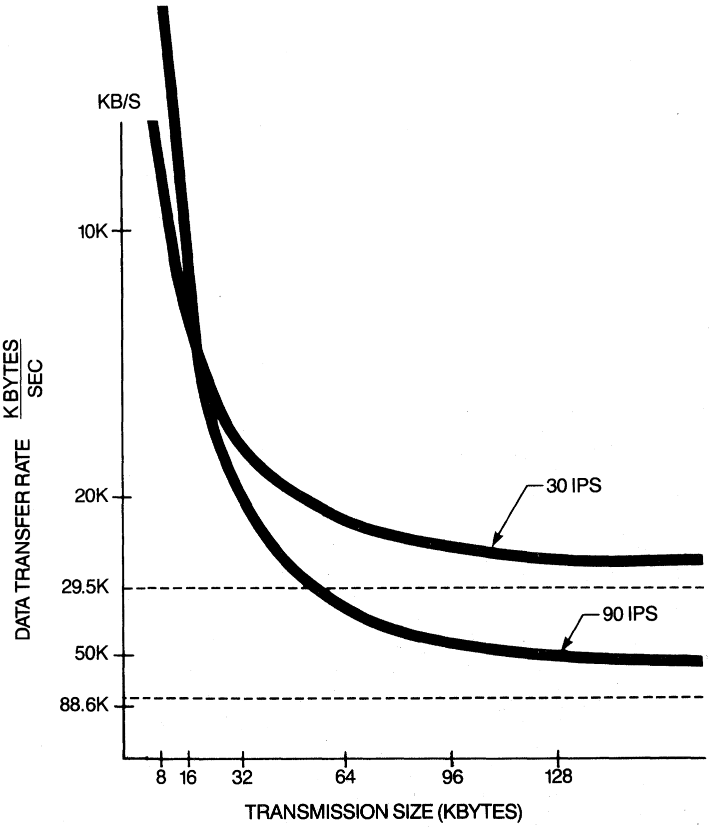


Figure 7-1 Effect of transmission size on transfer rate.

DISK TRANSFERS

Disk "average" transfer rates are normally several times that of even a 90-ips streaming-cartridge drive. But the way data is organized on a disk makes it a challenging task to smooth the flow and match the transfer rate required for continuous streaming.

This is particularly true when data is read from disk. The information may be scattered on different tracks, requiring frequent head-positioning delays. Latency (rotational) delays can also cause a problem.

Most Winchester controller/formatters support multi-sector accesses—a first requirement for adequate-sized transmissions. But such accesses are of little value if the tape is not able to take the data as contiguous sectors are read. Instead, only enough sectors would be read to fill the available buffers. The system would then have to wait until the disk makes a complete revolution before the next set of sectors can be read. At 3600 rpm this delay approaches 17 milliseconds—during which a 90-ips tape drive can write three 512-byte blocks. Additional buffering is therefore necessary to maintain streaming.

Sector interleaving may reduce the size of the required buffer. By assigning sequential addresses to alternate sectors (2-to-one interleaving) or every third sector (3-to-one interleaving) the transfer rate for multi-sector accesses is reduced by approximately half or two-thirds—approaching the transfer rate of a continuously streaming cartridge.

Streaming must also be sustained during the time it takes to move the disk heads from one track (or cylinder) to the next. Here the designer faces a tradeoff between buffer storage and the higher cost of a faster (i.e., voice coil) head-positioning actuator.

Head positioning between widely separated tracks could take more time than the amount of buffering can accommodate. The task, in this case, is to maximize the amount of data that can be transmitted between interruptions. One approach is to select multi-surface and multi-head disk drives with high cylinder capacities. A second alternative would be to organize data on the disk specifically for streaming transfers. If this is not possible, a set of contiguous tracks or cylinders could be reserved as a staging area for data to be transferred. (File-management functions performed during backup could be accomplished as the data is copied into the reserved space.)

An effective way to evaluate these alternatives is to start with a "typical" Winchester disk with the following operational parameters:

Rotational Latency	
maximum	20 msec
average	10 msec

Access Time

track to track	15 msec
average	65 msec
maximum	100 msec

Formatted Capacity

sector	256 bytes
track	8 Kbytes
cylinder	64 Kbytes

Sector Transfer Time: 0.625 msec.

When one cylinder of data is to be transferred and the disk head assembly is positioned over the required data, the burst transfer rate will be given by:

$$\text{Burst Transfer Rate} = \frac{\text{Transfer Size}}{{}^{t}\text{transfer}}$$

where ${}^{t}\text{transfer}$ is the time to tranfer the data and is given by

$${}^{t}\text{transfer} = \frac{\text{Transfer Size}}{\text{Sector Size}} \left(\begin{array}{c}\text{Sector Transfer}\\ \text{Time}\end{array}\right)\left(\begin{array}{c}\text{Interleave}\\ \text{Factor}\end{array}\right)$$

$$= \frac{64000}{256} (.625 \times 10^{-3}) \; (1)$$

$$= 0.15625 \text{ seconds}$$

Therefore:

$$\text{Burst Tranfer Rate} = \frac{64000}{.15625} = 409.6 \text{ Kbytes/sec}$$

This figure can be very misleading, however, because in many typical operations, the disk drive must also reposition the head assembly, adding significant time delays which result in a much lower effective transfer rate. In an image copy mode of operation, for example, a complete cylinder of data is transferred and the head assembly is then moved to the adjacent cylinder. This will typically add, in addition to the track-to-track delay, one rotational delay during which no data will be tranferred. Data transfer rate calculations for this case, assuming a 10 Megabyte operation will be:

$$\text{Effective Transfer Rate} = \frac{\text{Transfer Size}}{t_{\text{position}} + t_{\text{transfer}}}$$

$$\text{where } t_{\text{position}} = \text{Total time taken to position the disk head assembly}$$

$$= \frac{\text{Transfer Size}}{\text{Cylinder Size}} \text{ (Maximum Latency Time)}$$

$$= \frac{10{,}000{,}000}{64{,}000} (20 \times 10^{-3}) = 3.13 \text{ seconds}$$

$$\text{and } t_{\text{transfer}} = \frac{\text{Transfer Size}}{\text{Sector Size}} \left(\text{Sector Transfer Time}\right)\left(\text{Interleave Factor}\right)$$

$$= \frac{10{,}000{,}000}{256} (.625 \times 10^{-3}) \ (1)$$

$$= 24.41 \text{ seconds}$$

Therefore:

$$\text{Effective Transfer Rate} = \frac{10{,}000{,}000}{3.13 + 24.41} = \frac{10{,}000{,}000}{27.54}$$

$$= 363.1 \text{ Kbytes/sec}$$

Sector interleaving can further reduce the effective transfer rate of the Winchester disk. For example, for an image-copy mode of operation the effective transfer rate is reduced to 192.49 Kbytes/sec by an interleave factor of 2 and to 130.96 Kbytes/sec by an interleave factor of 3.

Effective data transfer rates become even less predictable in the file mode of operation. In this mode the data to be transferred does not reside on adjacent tracks. As a result, considerably more head repositions to virtually any area of the disk surface may be required to process the same amount of data. As an illustration, assume an average of 4K bytes of data are transferred between each seek operation, and each seek travels an average of half the disk surface. For a 2-Mbyte transfer the rate will be:

$$\text{Effective Transfer Rate} = \frac{\text{Transfer Size}}{{}^{t}\text{position} + {}^{t}\text{transfer}}$$

$$\text{where } {}^{t}\text{position} = \frac{\text{Transfer Size}}{(\text{Average File Size})}\begin{pmatrix}\text{Average Seek Time} \\ \text{Average Latency}\end{pmatrix}$$

$$= \frac{2{,}000{,}000}{4{,}000}(75 \times 10^{-3}) = 37.5 \text{ seconds}$$

$$\text{and } {}^{t}\text{transfer} = \frac{\text{Transfer Size}}{\text{Sector Size}}(\text{Sector Transfer Time})$$

$$= \frac{2{,}000{,}000}{256}(.625 \times 10^{-3}) = 4.88 \text{ seconds}$$

Therefore:

$$\text{Effective Transfer Rate} = \frac{2{,}000{,}000}{37.5 + 4.88} = \frac{2{,}000{,}000}{42.38} = 47.2 \text{ Kbytes/sec}$$

Such calculations indicate the wide variation in disk transfer rates which can be expected under various operating modes. In general disk transfers are characterized by bursts of data with a high transfer rate separated by periods of no data transfers. The effective transfer rate can vary from greater than 400 Kbytes to less than 50 Kbytes per second, separated by periods of no data which can be as long as 120 milliseconds.

To maintain a streaming tape operation, two basic conditions must be satisfied. The first requirement is an effective disk transfer rate which is greater than the average tape transfer rate. If this condition can not be met, the tape will overrun disk data transfers. Solutions to this problem could be a slower streaming tape drive or the "staging" of disk data on spare disk cylinders (or in buffer memory) prior to transfer to tape.

The second requirement for streaming is an adequate sized buffer to maintain tape data tranfers during disk latency time.

DATA BUFFERS

The two preceding sections indicate the wide range of variables that can affect the data transfer characteristics of streaming-cartridge drives and Winchester disks. The key to efficient system integration of streaming-cartridge drives is the matching of these two dissimilar and variable rates. One way to facilitate this task is by designing an appropriate data buffer architecture.

The input/output characteristics of such a buffer must be capable of handling the maximum burst transfer rate of the Winchester-disk and streaming-cartridge drives. Simultaneous disk and tape data transfers must also be supported. Possible solutions for this buffer architecture are FIFO, ring, or alternating multiple-memory buffers.

The size of the data buffer is dictated by the maximum disk latency that must be supported in a streaming-cartridge mode of operation. As indicated above, disk latency will vary with the type of operation being performed. Since the 90-ips cartridge drive will tranfer 512 bytes of data in approximately 6 msec, a general guideline would be to provide a minimum of 1K bytes of buffer storage for each 10 msecs of disk latency. Overhead timing for data buffer management may add to this requirement.

The actual location of the data buffers can vary depending on system architecture. One alternative could use host memory. Figure 7-2 shows this architecture and the resulting data paths.

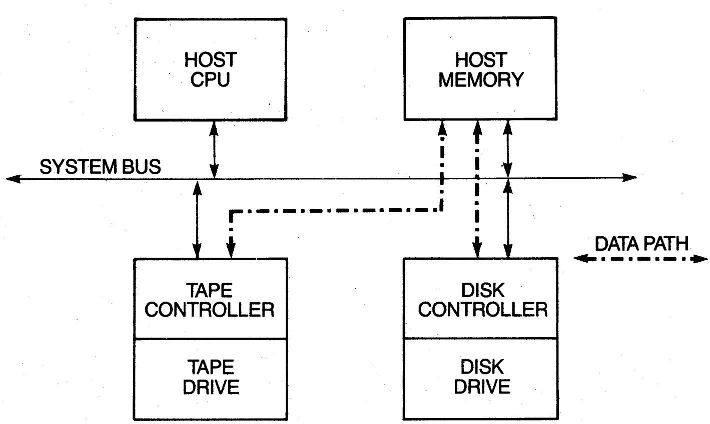


Figure 7-2 Data buffer in host memory.

The advantage of this approach is that it utilizes an existing memory resource. A disadvantage is that a large portion of host memory might be required for the buffer application and this could limit the tape operation to stand-alone functions or to inefficient on-line streaming. The approach could also result in increased system bus contention problems since each byte of data is tranferred twice (once into host memory and once out of host memory).

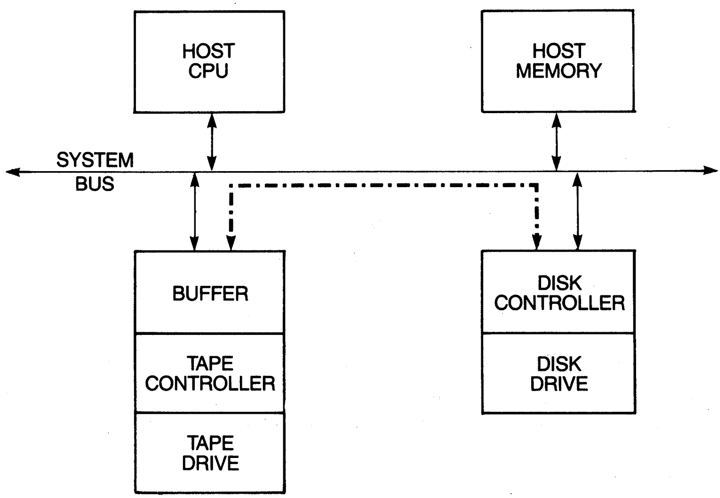


Figure 7-3 Data buffer in tape controller.

Another approach would be to place the data buffers in the tape controller itself. This architecture, shown in Figure 7-3, eliminates the host memory as a restriction and is an effective solution if the system allows device-to-device data tranfers—as in the SCSI (SASI) system architecture.

A third alternative would be a combined disk/tape controler with a shared data buffer (Figure 7-4). This solution has the advantage of off-loading the disk/tape transfer to the controller and frees the host CPU and system bus until the operation is complete. The same data buffer can also be used for disk/host or for tape/host data transfers. The following chapter describes an architecture that meets this criterion.

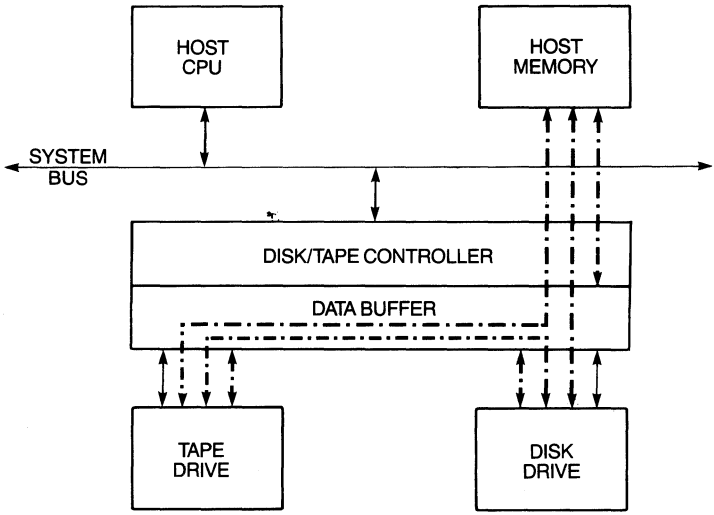


Figure 7-4 Data buffer in combined controller.

CHAPTER 8

Disk-Tape Storage

Software utilities and drivers are required to define and initiate disk-tape backup and restore operations. But in most cases, as noted in Chapter 1, any further participation by the host computer is wasteful in terms of both processing time and memory resources. Backed up and restored data are usually mirror images of each other at the record or file level.

What is needed, then, is enough "off-line" intelligence to control the two storage devices and coordinate disk-tape transfers across a data channel that is completely independent of the host I/O facilities. The two devices become, in effect, a single mass-storage system.

A typical system of this type is illustrated in Figure 8-1. Device-to-device transfers are not only transparent to the host, but can be given a user-selected priority level that allows the host to interrupt the transfers and directly access files on the disk.

The central control element for the entire storage system is a controller board (Figure 8-2) mounted on a Winchester drive. Computer-system integration at the hardware level is reduced, therefore, to the design of two elements: a host adapter board that plugs into the computer I/O bus and a single cable between host adapter and storage-system controller. Software

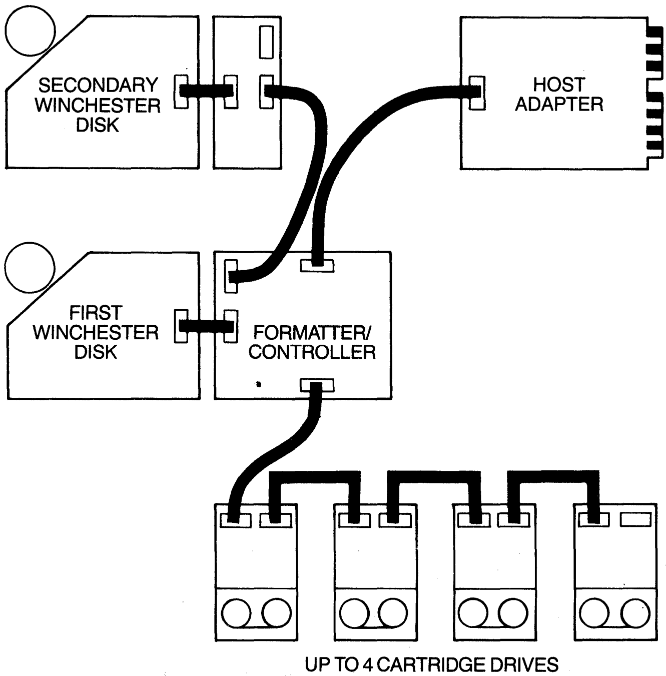


Figure 8-1 Mass-storage system with multiple disks and tape drives.

integration requirements are generally limited to modifications in the existing disk I/O driver and optional changes in the operating system or application programs to take advantage of the combined disk-tape architecture.

Communication between the operating and storage system is through a sequence of commands and status reports relating to both disk and tape. A number of additional daisychained disks, in any combination of capacities, can be typically added to the system. Tape storage can also be expanded by additional streaming-cartridge drives.

CONTROLLER HARDWARE

The on-board microprocessor shown in Figure 8-2 controls the flow of data through the controller circuitry but does not

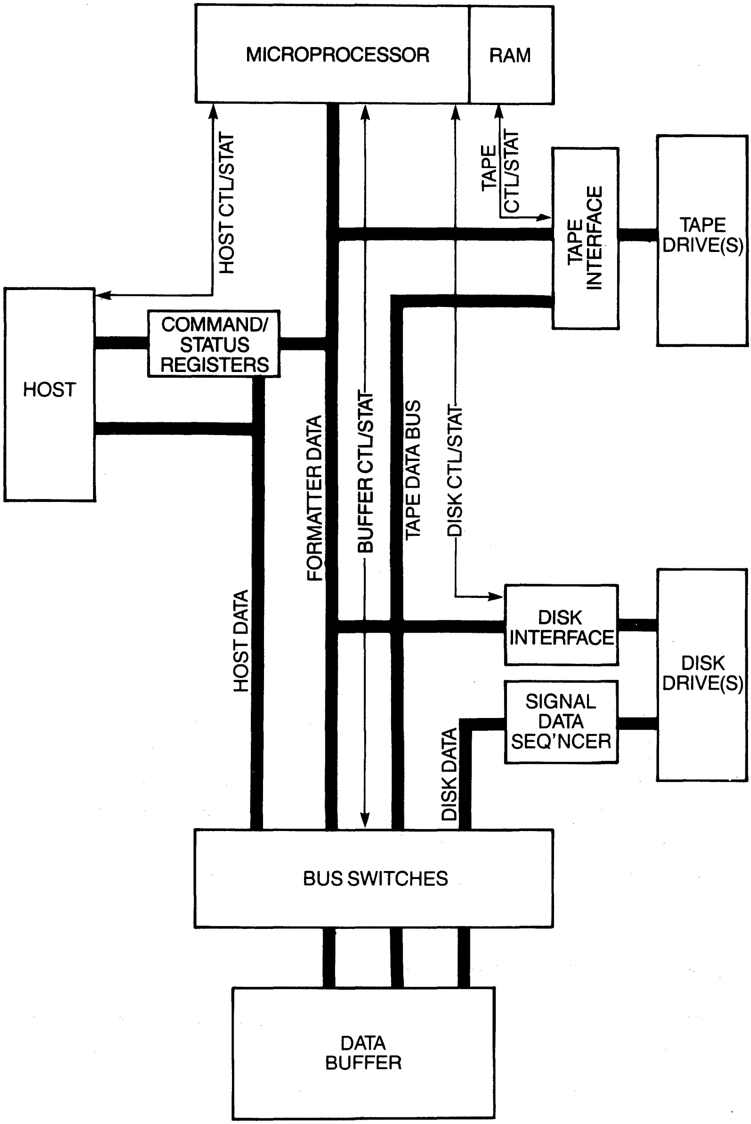


Figure 8-2 Block diagram, typical disk-tape controller.

directly process the transfers. Instead, the connecting link that establishes a host-disk, host-tape, or disk-tape data path is a rotating "ring" buffer. Data is read into one memory module, read out of a second, while the remaining buffers compensate for transfer-rate inequalities and disk latency.

Tape transfers flow to or from the intelligent streaming-cartridge drives on byte-wide paths. RAM stores tape-status information for interpretation by the microprocessor firmware. A serial data sequencer associated with the disk interface performs serial-parallel conversions for disk data and does CRC error checks on the serial data stream.

HOST INTERFACE

Control and handshaking lines interface directly with the microprocessor, but the principal communication links between host and microprocessor are two 16-byte registers. Commands and data-transfer parameters are loaded into one of the registers by the host computer and read by the microprocessor. The second register holds storage-system status bytes that can be accessed at any time by the host software driver.

Data, command, parameter, and status bytes are all carried in bit-parallel format by an 8-line bus. Four additional lines address one of the 16 byte locations in either the "command" or "status" register. Two lines indicate whether the host is writing into the addressed command-register location or reading from the corresponding status-register location. The top location in both registers transfers read or write data bytes between the data bus and the ring buffer.

COMMAND PARAMETERS

Only the first location in the controller/formatter command register contains a disk, tape, system, or test command. The other available locations are used to store parameters that define the operation to be performed. A principal function, therefore, of the software driver for the storage system is to translate operating-system storage instructions into a correct sequence of command/parameter bytes and status-byte responses.

A transparent command priority value defines the number of consecutive cylinders that are to be transferred between disk and tape in an off-line mode before the operation can be interrupted by a direct command from the host requesting immediate disk access. The lower the priority value, the more frequent the potential interruptions.

Tape-command parameters can be used to initiate such specific streaming-cartridge operations as rewinding, retensioning, and a complete erasure of all the data on the tape.

TRANSPARENT COMMANDS

"Transparent" commands relate to off-line tape operations, and can be interrupted by any subsequent direct command that does not involve a tape-drive control, read or write operation. For example, a command to "position" (rewind) the tape is transparent and the host computer is free to issue any other non-tape command before the operation is completed. The same is true of an "advance file mark" command. The term "file mark" refers to a unique marker on the tape which the user can insert into the data stream at any point to separate files and relocate them by counting file marks from the start of the tape or by reading an identification code or directory in the block immediately following a file mark.

Disk-tape transfers are always off-line, but can be either transparent or direct (non-interruptable). "Burst" commands are in the latter category and would be used only when the transfer has priority over any other computer-system operation.

CHAPTER 9

Streaming-Cartridge Drive

Chapter 7 defined the design decisions that must be made before a streaming-cartridge drive can be selected. Among the factors to be considered are the degree of intelligence (including the amount of buffering provided by the drive), storage capacity without a media change, and tape speed (as it affects the streaming transfer rate).

Archive streaming-cartridge drives (Figure 9-1) offer the system designer the following options:

a. Up to four separately addressable drives.

b. Cartridge capacities of 20 or 45 Mbytes.

c. Tape speeds of 30 or 90 inches per second (ips).

Archive drives also share a number of common characteristics. All are designed for 450-foot, industry-standard cartridges, available from several alternate sources. Flux-reversal densities can range up to 10,000 per inch at both tape speeds.

The streaming-cartridge drives are also designed to fit in the space occupied by a standard-size 8-inch flexible disk drive. Power requirements and connectors meet flexible-disk standards. The drives can be mounted in any position, horizontal or vertical, except that in a vertical mount it is advisable to turn the drive counterclockwise. Drive dimensions are given in Figure 9-2, specifications in Figure 9-3.

TAPE SPEED

The two principal electromechanical components of the streaming-cartridge device are the drive capstan and head carriage. Both "engage" the cartridge as it is inserted in place. The head carriage presses against the cartridge tape, maintaining constant tension. The soft-rubber drive capstan presses against the belt capstan inside the cartridge.

Both are also designed for bidirectional, "serpentine" recording. Tape speed, 30 or 90 ips, is precisely controlled in both directions by a servo feedback circuit. (30-ips drives advance or rewind at 90 ips except during a write or read operation.) Revolutions of the capstan motor transmit tachometer pulses to

DATA HANDLING

Formatted Capacity	20 Mbytes	20 Mbytes	45 Mbytes
Tape Speed	30 ips	90 ips	90 ips
No. of Tracks	4	4	9
No. of Heads	2	2	2
Control Lines	8	8	8
Commands	8	8	8
Status Bytes	6	6	6
Transfer Rates (Kbytes/sec)			
Burst	200	200	200
Average	28.9	86.7	86.7
Dump Times			
20 Mbytes	12 min	4 min	4 min
40 Mbytes	25 min*	9 min*	8 min

*Including a media change.

Figure 9-1 Archive intelligent streaming cartridge drives.

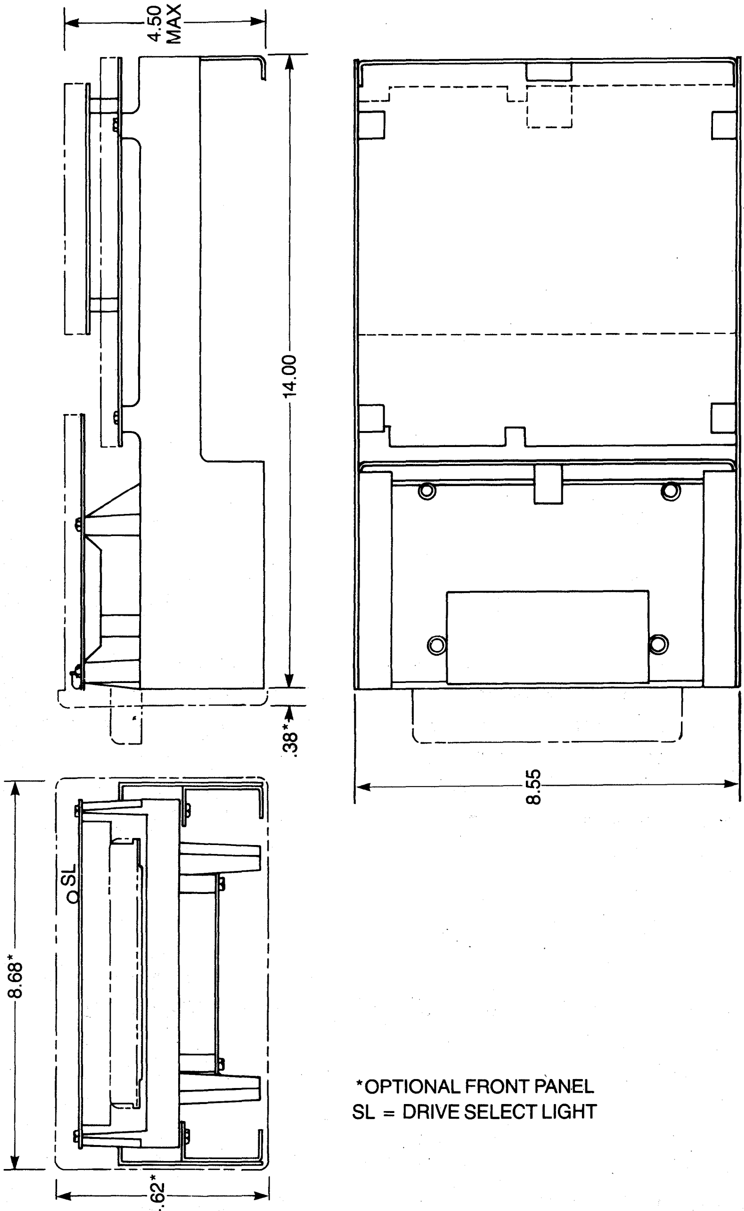


Figure 9-2 Dimensions, intelligent drive.

DATA HANDLING	
Head type	Read-after-write with separate erase bar
Flux Density	10,000 flux reversals per inch
Recording Form	NRZI, bit-serial, serpentine
Recording Code	Run-length limited (0,2)
Data Density	8,000 data bits per inch
Tape Speed	90 ips or 30 ips
Start/Stop Time	
90 ips	300 millisec max
30 ips	100 millisec max
Reliability	
MTBF	Greater than 3500 hours of use
MTTR	Less than 0.5 hours
Error Rates	
Soft Read Errors	Not more than 1 in 10^8
Hard Read Errors	Not more than 1 in 10^{10}
SHOCK	
Equipment Operational	2.5 g max., ½ sine wave, 11 msec duration on any axis
Equipment Non-Operational	25 g max., ½ sine wave, 11 msec duration on any axis
VIBRATION	
Equipment Operational	0.005 inch max peak-to-peak displacement, 0 to 63 Hz; 1 g max acceleration, 63 to 500 Hz
Equipment Non-Operational	0.1 inch max peak-to-peak displacement, 0 to 17 Hz; 1.5 g max acceleration, 17 to 500 Hz

Figure 9-3 Streaming-cartridge drive specifications.

POWER REQUIREMENTS

DC Voltages	+24 VDC	+5 VDC
Tolerance	±10%	±5%
Maximum Peak to Peak Ripple	500 millivolts	100 millivolts
Current		
Standby	0.2 amp nominal	3.5 amps max
Operational	0.8 amp nominal 1.7 amps max	3.5 amps max
Tape Start Surge	2.5 amps max Up to 300 millisec; may be longer for defective cartridge.	—
Power-On Sequence	24 VDC before 5 VDC (or use reset)	
Voltage Rise-Time	100 msec max	50 msec max
Power Dissipation	35 watts typical 60 watts maximum	

ENVIRONMENTAL REQUIREMENTS

	Operational	Non-Operational
Temperature	+5 to +45° C	-30 to +60° C
Relative Humidity (non-condensing)	20 to 80%	0 to 99%
Altitude	-200 to 15K Ft	-200 to 50K Ft

PREVENTIVE MAINTENANCE

Head cleaning should be done after the first 2 hours of tape movement of a new cartridge and every 8 hours of subsequent tape movement, using a lintless cotton swab coated with Isopropyl Alcohol or IBM tape cleaner.

PHYSICAL DIMENSIONS

Depth	14 ±.01 inches	355.6 ±.25 mm
Width	8.55 ±.01 inches	217.2 ±.25 mm
Height	4.5 + 0, -.2 inches	114.3 + 0, -.51 mm
Weight	4 ±.2 pounds	1.81 ±.09 kg

Figure 9-3(con.) Streaming-cartridge drive specifications.

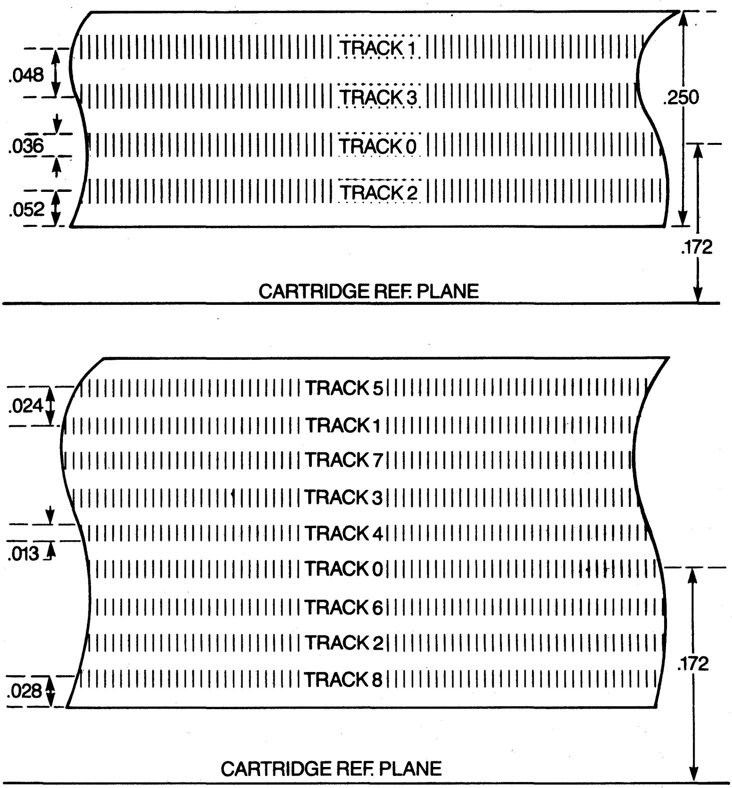


Figure 9-4 Track dimensions, ¼-inch cartridge tape.

a comparator. The comparator circuit outputs an error signal if there is any deviation from a crystal-controlled reference frequency. The error signal is fed to a phase-modulated amplifier which in turn drives the +24-VDC capstan motor.

The entire process is continuously monitored and controlled by the drive microprocessor. If the motor is stalled by a defective cartridge, for example, the motor duty cycle is immediately reduced to 50 milliseconds on, 1 second off.

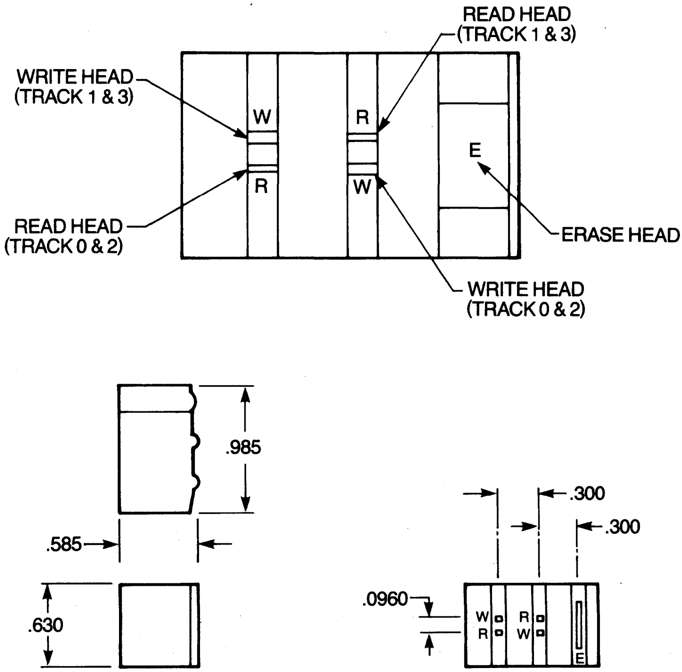


Figure 9-5 Read-after-write head dimensions.

BIDIRECTIONAL RECORDING

Tracks written and read by the bidirectional head assembly are illustrated in Figure 9-4. Dimensions of the head assembly are given in Figure 9-5.

Two sets of read-after-write heads are separated by twice the 0.048-inch distance between tracks when four tracks are recorded. The same head spacing is used for 45-Mbyte, nine-track drives, allowing them to read data written by a four-track drive. The lower set of heads is activated when the tape is moving forward, the upper set when the tape is moving in reverse. A single AC-erase head returns the entire tape to a neutral state during an erase sequence and when data is being written on track 0.

The dual-head design reduces the number of head repositions by half. The system can switch, for example, from tracks 0 to 1 and 2 to 3 electronically. Physical repositioning from one set of tracks to another is accomplished in less than one second.

Head positioning is in 0.001-inch increments. The tape itself is also positioned within close tolerance limits. The ANSI standard for data cartridges allows only a 0.004-inch difference between the narrowest tape and the widest tape guides.

Interchangeability is also enhanced by attenuating the read signal 75% during the read-after-write check for write errors. Blocks containing weakly recorded flux reversals are automatically rewritten. To guard against similar faults in the time domain, the drive controller narrows the read-after-write flux reversal "window" by 25%.

DRIVE INTELLIGENCE

Write and read channels of the streaming-cartridge drive are optimized for a particular flux-reversal density (e.g., 10,000 per inch). Design of the write-read heads and such features as unsaturated recording, AC-erase, and servo-controlled tape motion contribute to this capability. Drive circuits also support a serpentine recording format and include electronic interlocks to prevent the cartridge from running off the beginning or end of tape.

Beyond these capabilities, however, any significance attached to the flux-reversal record must be established by drive intelligence. Included among the functions provided by the drive, therefore, are the encoding of write data, decoding of read data, division of the data into blocks, and the addition of such formatting features as inter-block gaps, sync marks, CRC characters, and block addresses. The drive also corrects "soft" errors, reports "hard" errors and any other conditions affecting the performance of drive or media.

Archive intelligent streaming-cartridge drives perform all these functions.

The streaming-cartridge system can be expanded, moreover, with up to three additional daisychained devices to provide up to 180 Mbytes of attached backup storage.

Device addresses are established by switches on each device. Individual devices are then accessed by a selection command across the host interface. To preserve data integrity, the currently selected cartridge must be rewound to beginning of tape before another drive is selected.

Figure 9-6 is a block diagram of the circuits that contribute to this intelligence. The microprocessor has supervisory control over all activity, but does not directly process data transfers. Instead, write and read data flows through three 512-byte buffers. Parallel-serial conversions are accomplished as data is read in and out of the buffer storage.

PHASE-LOCKED LOOP

The phase-locked loop (PLL) shown in Figure 9-6 plays an important role in both the read and write modes. Servo control of the capstan motor maintains constant tape speed, but high-frequency, longitudinal vibrations in the tape can significantly change the instantaneous tape velocity. The PLL has a bandwidth considerably greater than that of the data rate and rapidly adjusts the flux-reversal "window" to reflect short-term variations—significantly increasing the drive's ability to read data from tape.

INTELLIGENT INTERFACE

The physical interface between the host computer and the streaming-cartridge drive is a 50-line cable, of which 16 lines are active signals.

Eight of the lines form a bidirectional bus that transmits (1) data to and from the drive, (2) single-byte commands from host to drive, and (3) status bytes stored by the intelligent-drive microprocessor and transmitted to the host on demand. The remaining lines represent control and status signals.

A complete description of the interface is given in Figure 9-7. Even pins 02-10 and 44-50 are reserved. All odd pins are signal returns, connected to signal ground at the device and should be connected to signal ground at the host. All of the interface sig-

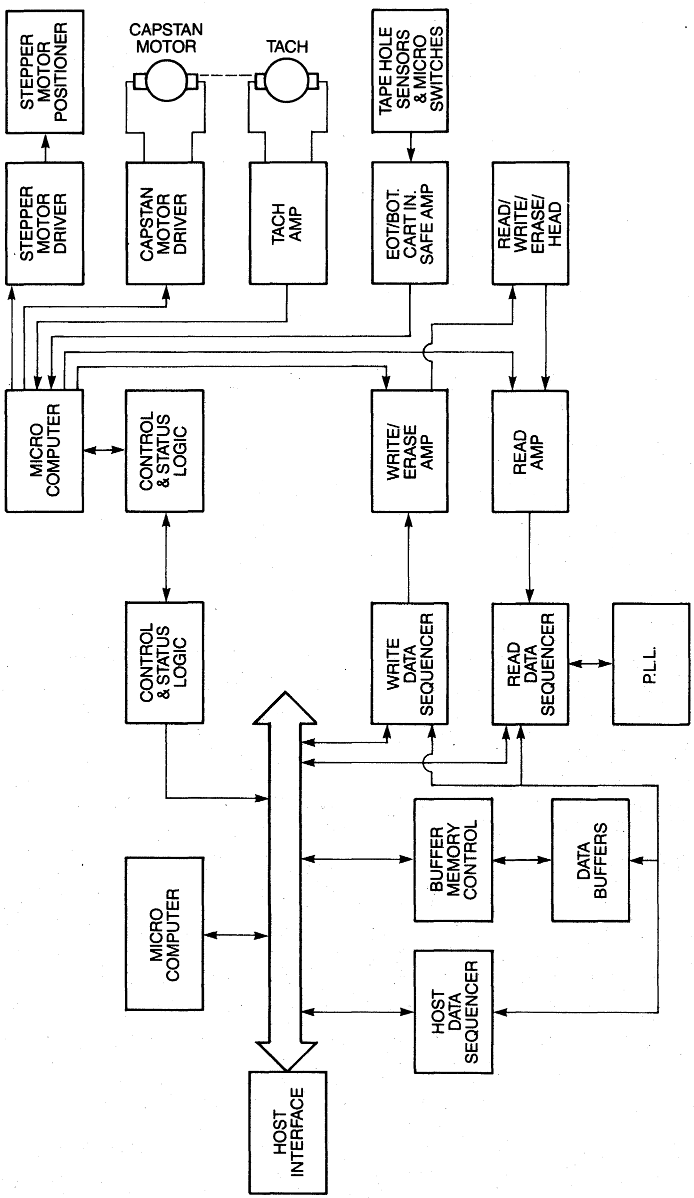


Figure 9-6 Functional block diagram, streaming-cartridge drive.

nals are low-active standard TTL logic levels, as measured at the controller:

True, Logic 1 (low level) = 0 to +0.8 VDC

False, Logic 0 (high level) = +2.4 to 5.0 VDC

The maximum cable length is approximately 30 feet. Signal terminations are shown in Figure 9-8. Signals from the device are capable of driving one standard TTL load (1.6 mA) in addition to the signal terminator. Signals from the host must be capable of driving a similar load.

Signal and power connectors are shown in Figure 9-9. The drive requires two DC voltages: +24 VDC and +5 VDC. Operational current requirements are 3.5 amps for +5 VDC, 0.8 amps for +24 VDC.

CONTROL AND STATUS LINES

Of the eight non-data lines, four provide a control input to the drive, the balance report drive status to the host. Two of the lines, -TRANSFER and -ACKNOWLEDGE, form a handshake for the high speed transmittal of data at burst rates up to 200 Kbytes per second, typically in conjunction with a DMA controller.

Two other lines, -REQUEST and -READY, serve an equivalent handshaking role for command and status bytes, and typically support programmed-I/O transfers. -ONLINE and -RESET are host control signals and -EXCEPTION is a controller status signal. -DIRECTION can be used to enable and disable the host-adapter drivers (Figure 9-8).

COMMAND CODES

The balance of the control-and-status communication link between host and an individual drive takes the form of command and status bytes transmitted across the byte-wide data bus.

The device will accept a command whenever -READY is active. The host indicates that it wants to send a command by setting -REQUEST.

PIN	NAME	TO*	DESCRIPTION
12	-HB7	B	-HOST-BUS Bit 7, MSB
14	-HB6	B	-HOST-BUS Bit 6
16	-HB5	B	-HOST-BUS Bit 5
18	-HB4	B	-HOST-BUS Bit 4
20	-HB3	B	-HOST-BUS Bit 3
22	-HB2	B	-HOST-BUS Bit 2
24	-HB1	B	-HOST-BUS Bit 1
26	-HB0	B	-HOST-BUS Bit 0, LSB
28	-ONL	D	-ONLINE — Host-generated control signal asserted prior to initiating a read or write operation and reset to terminate a read or write operation.
30	-REQ	D	-REQUEST — Host-generated control signal which indicates that command data has been placed on the data bus during a command sequence or that data has been taken from the data bus during a Read-Status Sequence. Can be asserted by host only when -READY or -EXCEPTION is asserted by device.
32	-RST	D	-RESET — Causes device to perform Power-On/Reset Sequence.
34	-XFR	D	-TRANSFER — Host-generated control signal which indicates that data has been placed on the data bus during a Write-Data Sequence or that data has been taken from the data bus during a Read-Data Sequence.
36	-ACK	H	-ACKNOWLEDGE — Device-generated signal which indicates that data has been taken from the data bus during a Write-Data Sequence or that data has been placed on the data bus during a Read-Data Sequence.

Figure 9-7 Interface lines, intelligent drive.

38	-RDY	H	-READY — Device-generated signal which indicates one of the following: (1) A command has been taken from the data bus. (2) Data has been placed on the data bus during a Read-Status Sequence. (3) A Beginning-Of-Tape, Retension or Erase Sequence has been completed. (4) A buffer is ready to be filled by the host, or a WRITE FILE MARK command can be issued or ONLINE deasserted during a Write-Data Sequence. (5) A WRITE FILE MARK command has been completed during a Write-File-Mark Sequence. (6) A buffer is ready to be emptied by the host, or a READ FILE MARK can be issued or ONLINE deasserted during a Read-Data Sequence. (7) Device is ready to receive a new command.
40	-EXC	H	-EXCEPTION — Device-generated signal which indicates that an exception condition exists in the device and host must initiate a Read-Status Sequence.
42	-DIR	H	-DIRECTION — Device-generated signal which, when asserted, causes host data-bus drivers to assume high-impedance states and device drivers to assert their data-bus levels. When reset, causes host drivers to assert their data-bus levels and device drivers to assume high-impedance states.

*B = Bi-Directional; D = Device; H = Host

Figure 9-7(con.) Interface lines, intelligent drive.

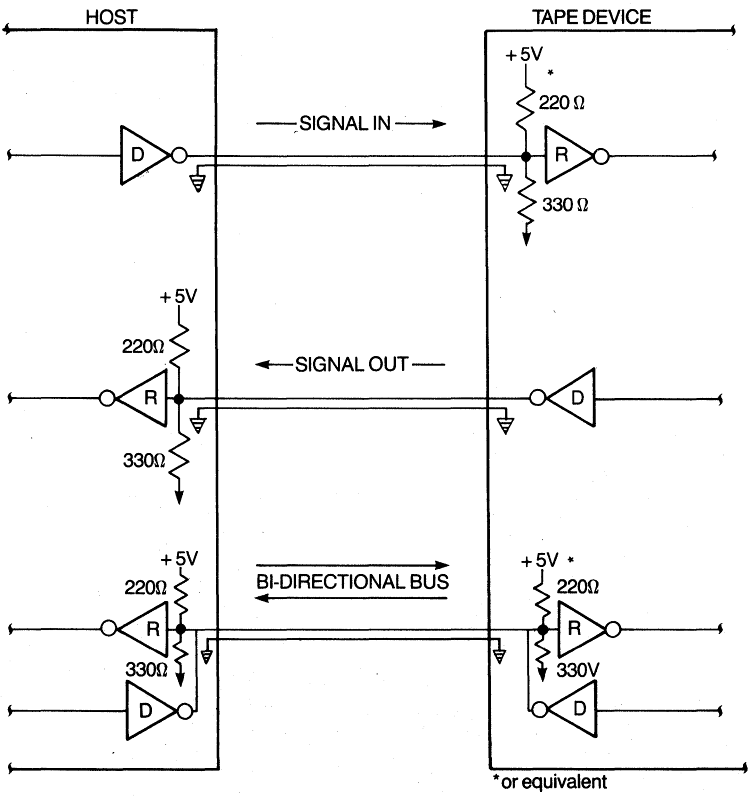


Figure 9-8 Interface drivers and receivers.

Figure 9-10 lists the command codes. The three most significant bits establish the type of command. All of the select commands, for example, have 000 in the command-type field.

In a similar manner, the least significant bits of three "tape-positioning" commands indicate whether the drive is to (1) rewind the cartridge to the beginning of tape, (2) retension the cartridge with a high-speed forward and rewind cycle, or (3) erase the entire tape.

READ STATUS is the only command that will be accepted when -READY is not true. In this case, however, -EXCEPTION must be true. In fact, with -EXCEPTION asserted by the drive, READ STATUS is the only command that will be accepted.

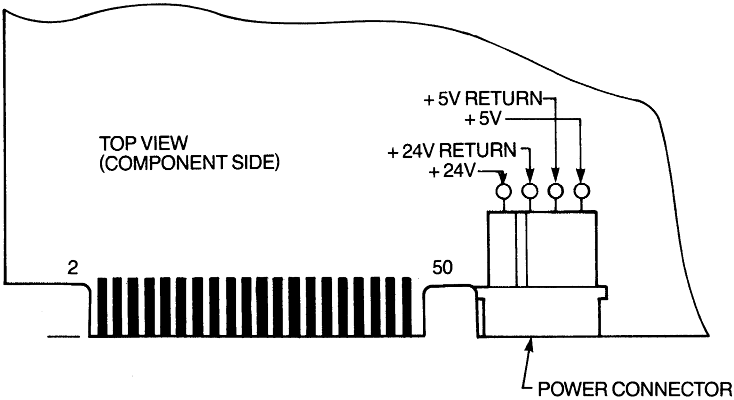


Figure 9-9 Signal and power connectors.

Four commands, WRITE DATA, WRITE FILE MARK, READ DATA, and READ FILE MARK, also require that -ONLINE be true. If one of these is transmitted in the absence of -ONLINE, the device will respond with -EXCEPTION.

SELECT specifies the device to be accessed and controlled in a multi-drive system. The currently selected drive must be at beginning of tape before another drive can be selected. A Select Light on the drive serves as a warning to the operator that a cartridge extraction with the Select Light on will create an -EXCEPTION condition. If no drive is selected before another command is issued, the controller automatically defaults to Drive 0.

BEGINNING OF TAPE rewinds the cartridge and positions the head to the start of Track 0. The drive is then deselected if bit 4 of the SELECT command is ZERO.

RETENSION performs a retensioning pass to "condition" the cartridge tape. The cartridge is fully rewound, advanced to end of tape, and again rewound. The procedure reduces the number of read errors after the tape has been in storage for an extended length of time or has been subject to physical or thermal shock. As in the case of the previous command, the drive is deselected if bit 4 of the SELECT command is ZERO.

ERASE completely erases the tape by rewinding the cartridge to beginning of tape, then advancing the cartridge to the

BIT 7654 3210	COMMAND
0000 0001	SELECT Drive 0*
0000 0010	SELECT Drive 1*
0000 0100	SELECT Drive 2*
0000 1000	SELECT Drive 3*
0001 0001	SELECT Drive 0**
0001 0010	SELECT Drive 1**
0001 0100	SELECT Drive 2**
0001 1000	SELECT Drive 3**
0010 0001	BEGINNING OF TAPE
0010 0010	ERASE
0010 0100	RETENSION
0100 0000	WRITE
0110 0000	WRITE FILE MARK
1000 0000	READ
1010 0000	READ FILE MARK
1100 0000	READ STATUS
111X XXXX	(Reserved)

*Drive Select light OFF (drive deselected) when cartridge is returned to beginning of tape, otherwise always ON.

**Drive Select light always ON.

Figure 9-10 Commands recognized by streaming-cartridge drive.

physical end of tape with the erase head activated. All tracks are erased simultaneously. The cartridge is rewound to beginning of tape, and again the drive is deselected if bit 4 of the SELECT command is ZERO.

WRITE records user data blocks on the tape. The device automatically divides the data into 512-byte blocks and generates block address, CRC, gap and sync codes.

WRITE FILE MARK records a non-data File Mark block to mark the division between files and facilitate the search for previously recorded data. File Marks are also generated when, for example, -ONLINE is dropped during a write operation.

READ transfers data from tape to host (minus CRC, gap and sync codes and block addresses). Each block is completely read and verified before it is transmitted.

READ FILE MARK initiates a forward search for a block containing the File Mark code. The tape is stopped when the next File Mark has been located. No data is transmitted to the host during this operation.

READ STATUS is used by the host to request a status report from the device. The host must read status any time that the device sets -EXCEPTION. It is also advisable to read status at the completion of any read or write operation. This allows the host to receive an error report for the operation and clear the error count stored by the drive microprocessor.

STATUS BYTES

The drive maintains six status bytes to inform the host of (1) operational and data-handling errors, (2) the number of write/-read errors, and (3) the number of buffer underruns (Figure 9-11).

The host may request the status bytes at any time by issuing a READ STATUS command. Except for bit 3 of Status Byte 1 and bits 3 through 6 of Status Byte 0, all of the status-byte information is reset, including the error and underrun logs, each time status is read.

Byte 1 is the more significant of the two status-bit bytes. In both cases, bit 7 is set only if another bit in the byte is set and provides a convenient way to check whether the other bits should be tested.

STATUS BYTE 1

Bit 0: **Reset/Power-Up Occurred** bit is set after the host asserts -RESET or when the device is powered up. The bit is reset by a Read-Status Sequence.

Bit 1: Reserved
Bit 2: Reserved
Bit 3: **Beginning of Media** bit is set whenever the car-

STATUS BYTE 1

(Bit)	
0	Reset/Power-Up Occurred
1	Reserved
2	Reserved
3	Beginning of Media
4	Eight or More Retries
5	No Data Detected
6	Illegal Command
7	Exception Byte 1

STATUS BYTE 0

(Bit)	
0	File Mark Detected
1	Block-in-Error (BIE) Not Located
2	Unrecoverable Data Error
3	End of Media
4	Write Protect
5	Drive Not Online
6	Cartridge Not In Place
7	Exception Byte 0

STATUS BYTE 2
Most significant byte, soft-error count

STATUS BYTE 3
Least significant byte, soft-error count

STATUS BYTE 4
Most significant byte, buffer-underrun count

STATUS BYTE 5
Least significant byte, buffer-underrun count

Figure 9-11 Status bytes returned to host.

tridge is logically at beginning of tape, track 0. The bit is reset when the tape moves away from beginning of tape. This is the only bit in this byte that does not set -EXCEPTION when it goes true, nor is it reset by a Read-Status Sequence.

Bit 4: **Eight or More Retries** bit is set when eight or more attempts are made to recover the same data block (an indication that the tape cartridge is nearing end of life). The bit is reset by a Read-Status Sequence.

Bit 5: **No Data Detected** bit is set when an unrecoverable data error occurs due to lack of recorded data. Absence of recorded data is the failure to detect a data block within a controller time-out equal to the read-data time for 32 blocks. The bit is reset by a Read-Status Sequence.

Bit 6: **Illegal Command** bit is set if any of the following occurs. The bit is reset by a Read-Status Sequence.

a. SELECT command is issued with no drives or more than one drive indicated.

b. Position-type command (BEGINNING OF TAPE, ERASE, RETENSION) is issued with incorrect qualifier bits.

c. -ONLINE is not asserted and a WRITE, WRITE FILE MARK, READ or READ FILE MARK command is issued.

d. A command other than WRITE or WRITE FILE MARK is issued during the execution of a Write-Data Sequence.

e. A command other than READ or READ FILE MARK is issued during the execution of a Read-Data Sequence.

f. A drive is deselected by another SELECT command when the cartridge in the currently selected drive is not at beginning of tape, track 0.

Bit 7: **Exception Byte 1** bit is set if any other bit in Status Byte 1 is set.

STATUS BYTE 0

Bit 0: **File Mark Detected** bit is set when a File Mark is detected during a Read-Data or Read-File-Mark Sequence. The bit is reset by a Read-Status Sequence.

Bit 1: **Block-in-Error (BIE) Not Located** bit is set when an unrecoverable data error occurs and the device can not confirm that the last block transmitted was the BIE. The bit is reset by a Read-Status Sequence.

Bit 2: **Unrecoverable Data Error** bit is set when the device experiences a hard error during read or write operations. The bit is set after 16 attempts to write or read a block of data. The bit is reset by a Read-Status Sequence.

Bit 3: **End of Media** bit is set when the logical early-warning hole of the last track is detected during a write operation. This bit will remain set as long as the drive is at logical end of media. The End of Media bit will not be set during a normal read operation, nor will it be reset by a Read-Status Sequence.

Bit 4: **Write Protect** bit is set if the cartridge write-protect plug is set in the file-protect "safe" position. Operator must change the write-protect plug position before the status bit will reset.

Bit 5: **Drive Not Online** bit is set if the selected drive is not physically connected to the host or is not receiving power. Operator must correct the condition before the status bit will reset.

Bit 6: **Cartridge Not In Place** bit is set if a cartridge is not fully inserted into the drive. Operator must correct the condition before the status bit will reset.

Bit 7: **Exception Byte 0** bit is set if any other bit in Status Byte 0 is set.

STATUS BYTES 2 AND 3

Status Bytes 2 and 3 contain a 16-bit binary count of soft write and read errors. Byte 2 is the most significant, Byte 3 is the least significant.

For write operations, the count increments for each data block that is rewritten due to a read-after-write error. Since the rewrite sequence rewrites two data blocks for each rewrite attempt, the counter increments twice for each rewrite.

During read operations, the count is incremented for each data block requiring one or more read retries. A read retry occurs whenever an unexpected CRC error occurs (see Read-Error Sequence).

STATUS BYTES 4 AND 5

Status Bytes 4 and 5 contain a 16-bit binary count of buffer underruns. Byte 4 is the most significant, Byte 5 is the least significant.

For write operations the count is incremented each time the host is unable to keep data flowing to the device. If the device is ready to write the next block, but a buffer is not full and ready to be written, the device will stop tape motion and wait for the host.

During a read operation, the count increments when the host is unable to empty the drive buffers fast enough. If an empty buffer is not available for the next block of data to be read from the tape, tape motion will stop and the device will again wait for the host.

OPERATIONAL SEQUENCES

The intelligence built into the streaming-cartridge drive responds to commands and control-line signals from the host and initiates a series of operational sequences that can be grouped into four categories:

Control and Status Operations

Tape Positioning Operations

Write Operations

Read Operations

The drive microprocessor firmware is also programmed to respond to a variety of conditions which the drive may encounter as these operations are attempted or performed, such as the lack of a cartridge, inability of the host to accept data at a streaming rate, or a write or read error.

Appropriate responses, which again take the form of operational sequences, are performed automatically by the drive, independent of the host. In nearly every case, however, the sequence ends with a prompt to the host (-EXCEPTION set) which initiates a status request.

AT POSITION FLAG

The cartridge tape can be at either one of two locations at the start of any operational sequence: at the logical beginning of tape or at a midway "at position" point where the tape was stopped by a previous operation.

In the latter case, the drive microprocessor sets an "At Position flag" to ONE to indicate that any subsequent write or read operation should start with a repositioning sequence to locate the end of the data that has already been written or read. The At Position flag is set to ZERO if the cartridge is at beginning of tape.

Current status of the At Position flag is not included in the information returned to the host in response to a READ STATUS command.

SELECT LIGHT BIT

The SELECT command (Figure 9-10) identifies the drive that is to perform subsequent operations requested by the host. The drive microprocessor stores bit 4 of the SELECT command. If no SELECT command has been issued, the microprocessor program defaults to Drive 0 and sets bit 4 to ZERO.

The visible function of bit 4 is to control the state of the "select" light on the face of the specified drive, warning the operator that when the light is on, the cartridge should not be removed. Removal of the cartridge will generate an -EXCEPTION condition. Hence, the term "Select Light bit" for the stored value of bit 4. Equally important, however, is the fact that with the Select Light bit set to ONE, the drive remains selected until the host issues another SELECT command (the cartridge must be at beginning of tape) or asserts -RESET. By comparison, with the Select Light bit set to ZERO, the drive is also deselected whenever the cartridge is rewound, for any reason, to beginning of tape.

CONTROL AND STATUS OPERATIONS

Four operational sequences are included in the control-and-status category:

Power-On/Reset Sequence

Command-Reject Sequence

Select-Drive Sequence

Cartridge-Extraction Sequence
Read-Status Sequence

POWER-ON/RESET SEQUENCE

The Power-On/Reset Sequence (Figure 9-12) provides the host with information on power-on occurrences in the device.

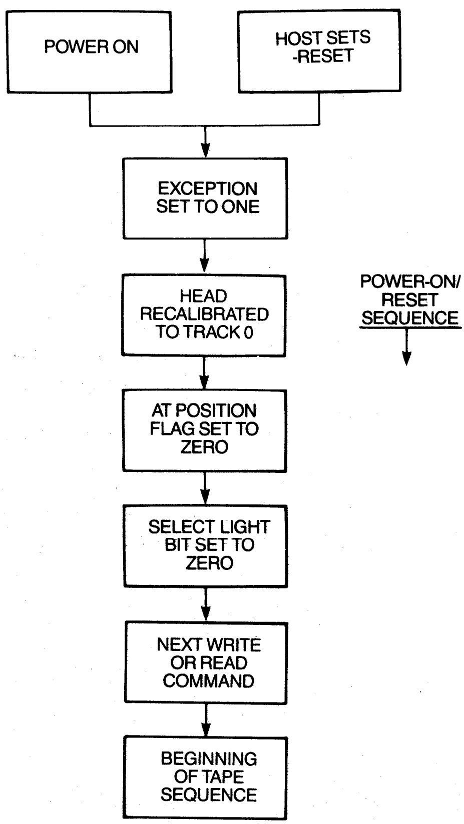


Figure 9-12 Power On/Reset Sequence.

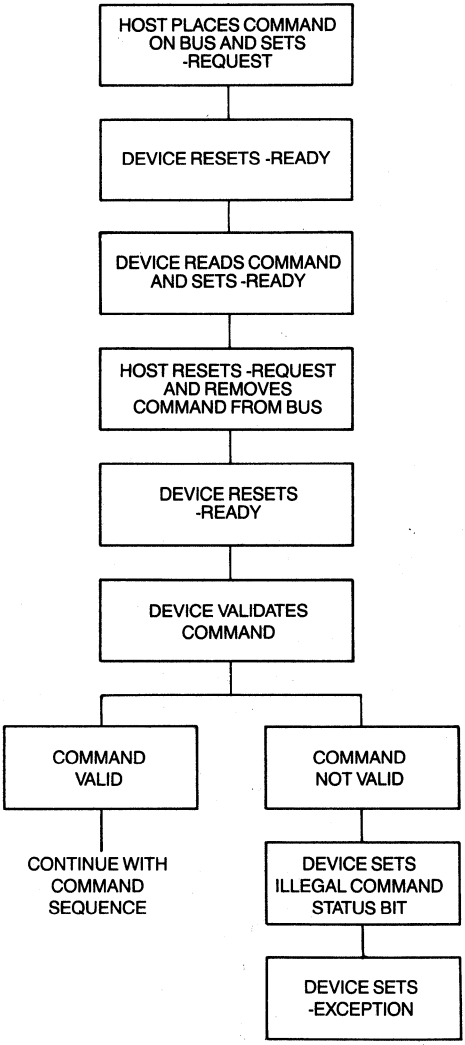


Figure 9-13 Command-Reject Sequence

The sequence also provides the user with a convenient method for initializing the device.

1. Host applies power to the device or asserts the -RESET line for at least 13 microseconds. Drive circuitry is reset.
2. When power has been applied for approximately 3 seconds or when the -RESET pulse has terminated, the device initializes operating parameters, clears the At Position flag and Select Light bit, defaults to Drive 0 for subsequent commands, and recalibrates the drive heads to Track 0.
3. Device informs host that Power-On/Reset Sequence has been completed by asserting -EXCEPTION. The host responds with a Read-Status Sequence.

COMMAND-REJECT SEQUENCE

1. Host places command on bus and then sets -REQUEST.

2. Device resets -READY and after reading the command sets -READY to its true state to indicate that the command has been read.

3. Host resets -REQUEST and removes command from the bus. Device completes the handshake by resetting -READY.

4. Device then validates the command (e.g. valid command code, write operation to a write protected cartridge, etc.)

5. If the command is valid the device continues with the command execution sequence.

6. If the command is not valid the controller aborts the command, sets the Illegal Command status bit, and sets EXCEPTION to initiate a Read Status Sequence.

SELECT-DRIVE SEQUENCE

The Select-Drive Sequence allows the host to select one of up to four drives (Figure 9-10).

1. Host verifies that -READY is true, places SELECT command on bus, then sets -REQUEST.
2. In response to -REQUEST, the device resets -READY. Host must maintain the command on the bus and keep the REQUEST line in a true state until the device returns -READY to its true state (normally less than 500 microseconds).

3. Host resets -REQUEST and removes command from the bus. Device resets -READY, validates the command, stores the drive address (bits 0 through 3) and the Select Light bit (bit 4), and indicates command completion to the host by again asserting -READY.

4. If Select Light bit is a ONE, the drive remains selected until deselected by another SELECT command (provided the At Position flag is off) or a -RESET pulse.

5. If Select Light bit is ZERO, drive remains selected until deselected by a -RESET pulse or by any sequence that ends with cartridge at beginning of tape.

CARTRIDGE-EXTRACTION SEQUENCE

The host/device interface requires that a drive be continuously selected and a cartridge in place during the performance of any function involving tape motion. Moreover, integrity of recorded data requires that the cartridge be completely rewound before it is removed from the controlled environment of the drive.

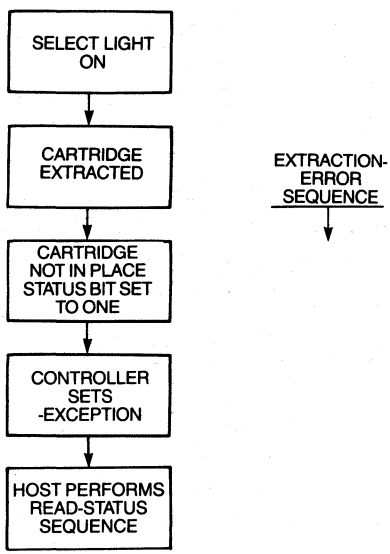


Figure 9-14 Cartridge-Extraction Sequence.

If bit 4 of the SELECT command is a ZERO, the Select Light on the selected drive is continuously lit until the drive is automatically deselected by rewinding the cartridge to beginning of tape. If bit 4 is a ONE, the Select Light is always lit—independent of the tape position.

Should the cartridge be removed while the Select Light is on, the device will perform the Cartridge-Extraction Sequence (Figure 9-14).

1. Device sets the Cartridge Not In Place status bit to ONE.
2. Device informs the host that the cartridge was removed by asserting -EXCEPTION to initiate a Read-Status Sequence.

A drive is selected momentarily during a Read-Status Sequence. Cartridge removal is not monitored during this non-motion sequence.

READ-STATUS SEQUENCE

The Read-Status Sequence provides host with information about the selected drive in response to -EXCEPTION going true or, at the user's option, to receive an error report and clear the error count at the completion of a read or write operation. In the latter case, host must verify that -READY is true and a drive has been selected. Figure 9-15 is a sampling of typical status responses.

1. Host places READ STATUS command on the bus.

2. Host sets -REQUEST. In response, device resets -EXCEPTION (or -READY), reads command, sets -READY to its true state to indicate that the command has been read but not necessarily accepted.

3. Host resets -REQUEST and removes command from the bus. Device resets -READY, validates the command, and changes -DIRECTION to true (device to host).

4. Device places first status byte on bus and begins handshaking of status bytes by setting -READY. Host reads the first byte, then sets -REQUEST to indicate byte has been read. Host must ensure that -REQUEST remains set for a minimum of 20

	BYTE 1 7654 3210	BYTE 0 7654 3210	DESCRIPTION
a	0000 0000	110x 0000	No Cartridge*
b	0000 0000	1111 0000	No Drive*
c	x000 x000	1001 0000	Write Protected*
d	0000 0000	1000 1000	End of Media
e	1000 1000	100x 0100	Read or Write Abort*
f	0000 0000	100x 0100	Read Error, Bad Block Transfer
g	0000 0000	100x 0110	Read Error, Filler Block Transfer
h	1010 0000	100x 0110	Read Error, No Data
i	1010 0000	100x 1110	Read Error, No Data and End of Media
j	0000 0000	100x 0001	Read a File Mark
k	1100 x001	xxxx 0000	Illegal Command*
l	1000 x000	xxxx 0000	Power On/Reset*
m	1xxx 1xxx	xxxx xxxx	8 or More Read Retries

Note: x denotes "could be either 0 or 1."
*Inhibits or terminates operation.

a. **No Cartridge** — Selected drive did not contain a cartridge when BEGINNING OF TAPE, RETENSION, ERASE, WRITE, WRITE FILE MARK, READ, or READ FILE MARK command was issued or cartridge was removed while the Drive Select light was on.
b. **No Drive** — Selected drive was not present when BEGINNING OF TAPE, RETENSION, ERASE, WRITE, WRITE FILE MARK, READ or READ FILE MARK was issued.
c. **Write Protected** — Selected drive contained write-protected (safe) cartridge when ERASE, WRITE, or WRITE FILE MARK was issued.
d. **End of Media** — Tape has passed the early-warning hole of the last track during a write operation.

Figure 9-15 Typical error and status responses.

e. **Read or Write Abort** — Sixteen same-block rewrites occurred during a write operation or unrecoverable reposition error occurred during a write or read operation. Tape has returned to beginning of tape.

f. **Read Error, Bad Block Transfer** — Sixteen same-block retries failed to recover block without CRC error, last block transferred contained data from the erroneous data block for off-line reconstruction.

g. **Read Error, Filler Block Transfer** — Sixteen same-block retries failed to recover block without CRC error, last block transferred contained filler data to keep total block count correct.

h. **Read Error, No Data** — Two same-block retries failed to recover the next or subsequent blocks or a File Mark. No filler data is transferred.

i. **Rear Error, No Data and EOM** — Sixteen same-block retries failed to recover the next or subsequent blocks and the logical end-of-tape holes on the last track were encountered.

j. **File Mark Read** — A File Mark block was read during a READ or READ FILE MARK command.

k. **Illegal Command** — One of the following events occurred:
Attempt to select 0, 2, 3 or 4 drives.
Attempt to change drive selection while At Position flag is set.
Attempt to command BEGINNING OF TAPE, RETENSION or ERASE simultaneously.
Attempt to WRITE, WRITE FILE MARK, READ or READ FILE MARK with -ONLINE reset.
Attempt to issue a command other than WRITE or WRITE FILE MARK during a WRITE command.
Attempt to issue any command other than READ or READ FILE MARK command during a READ command.

l. **Power On/Reset** — A power-on or reset by the host has occurred.

m. **8 or More Read Retries** — At least eight read retries were attempted on the same block.

Figure 9-15(con.) Typical error and status responses.

microseconds. Device resets -READY and host resets -REQUEST to complete the transmission of the first status byte.

5. Step 4 is repeated for the remaining five status bytes. All six status bytes must be transferred in order to exit the sequence.

6. After host resets -REQUEST on the last status byte, device changes -DIRECTION to false (host to controller). Device sets -READY and waits for next command.

TAPE POSITIONING OPERATIONS

Three operational sequences, each initiated by a separate host command, move the tape without writing or reading data:

Beginning-of-Tape Sequence

Retension Sequence

Erase Sequence

All three operations end with the tape positioned at beginning of tape, ready for a write or read operation.

BEGINNING-OF-TAPE SEQUENCE

The Beginning-of-Tape Sequence allows the host to position the tape in the selected drive at the start of Track 0 (Figure 9-16).

1. Host verifies that -READY is true, places BEGINNING OF TAPE command on the bus, then sets -REQUEST.

2. Device resets -READY and after reading the command, again sets -READY to its true state to indicate that the command has been read.

3. Host resets -REQUEST and removes command from the bus. Device completes the handshake by resetting -READY.

4. When the command has been validated the device selects high speed (if 30-ips model), sets reverse direction, clears the At Position flag to ZERO, enables the capstan motor, and rewinds cartridge to beginning of tape.

5. Device indicates command completion to host by asserting -READY.

6. Drive is deselected if Select Light bit is ZERO.

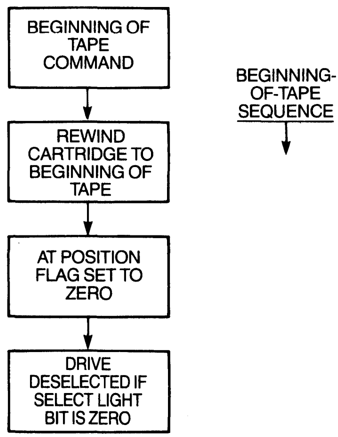


Figure 9-16 Beginning-of-Tape Sequence.

RETENSION SEQUENCE

Tape retension is recommended by cartridge-tape suppliers before writing or reading data when a cartridge has been subjected to a change in environment or a physical shock, has been stored for a prolonged period of time or at extreme temperature, or has been previously used in a start-stop mode. It could also assist in recovering data when reading difficulties are encountered.

The Retension Sequence of full-tape rewind, advance, and rewind is shown in Figure 9-17.

1. Host verifies that -READY is true, places RETENSION command on the bus, then sets -REQUEST.
2. Device resets -READY and after reading the command, again sets -READY to its true state to indicate that the command has been read.
3. Host resets -REQUEST and removes command from the bus. Device completes the handshake by resetting -READY.
4. When the command has been validated, the device selects high speed (if 30-ips model), clears the At Position flag to ZERO, and rewinds cartridge to beginning of tape.

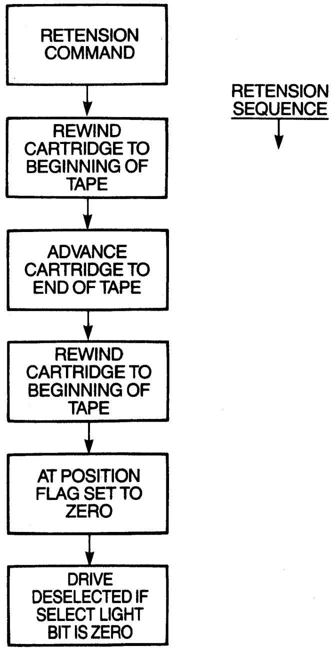


Figure 9-17 Retension Sequence.

5. Device selects high speed, reverses direction, and advances cartridge to end of tape.
6. Device selects high speed, again reverses direction, and rewinds cartridge to beginning of tape.
7. Device indicates command completion to the host by asserting -READY.
8. Drive is deselected if Select Light bit is ZERO.

ERASE SEQUENCE

The Erase Sequence (Figure 9-18) is used to completely clear the tape of previous data. It also retensions tape.

1. Host verifies that -READY is true, places ERASE command on the bus, then sets -REQUEST.
2. Device resets -READY and after reading the command, again sets -READY to its true state to indicate that the command has been read.

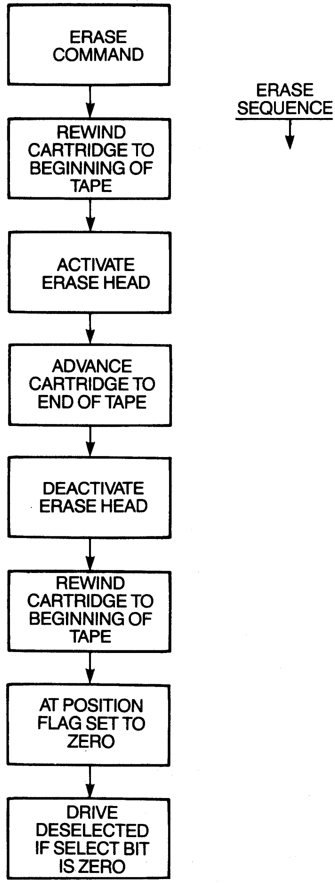


Figure 9-18 Erase Sequence.

3. Host resets -REQUEST and removes command from the bus. Device completes the handshake by resetting -READY.

4. When the command has been validated, the device selects high speed (if 30-ips model), clears the At Position flag and rewinds cartridge to beginning of tape.

5. Device enables the erase bar, selects high speed and advances cartridge to end of tape (erasing the entire tape).

6. Device disables the erase bar, selects high speed, reverses direction, and rewinds cartridge to beginning of tape.

7. Device indicates command completion to the host by asserting -READY.
8. Drive is deselected if Select Light bit is ZERO.

WRITE OPERATIONS

Write operations are initiated by one of two commands: WRITE DATA or WRITE FILE MARK. A total of nine different sequences can result as these operations are initiated, performed, or terminated:

Write-Data Sequence

Last-Block Sequence

Write-Underrun Sequence

End-of-Media Sequence

Write-File-Mark Sequence

Write Off-Line Sequence

Write-Reposition Sequence

Write-Error Sequence

Write-Abort Sequence

WRITE-DATA SEQUENCE

The Write-Data Sequence transfers data from host to device and stores it as a verified, formatted tape record. As shown in Figure 9-19, a Write-Reposition Sequence replaces Steps 6 through 8 if the At Position flag is ONE.

1. Host verifies that -READY is true, places WRITE command on the bus, sets -ONLINE, then sets -REQUEST.
2. In response to -REQUEST, device resets -READY, reads the command, and returns -READY to its true state to indicate that the command has been read.
3. Host resets -REQUEST and removes command from the bus. Device resets -READY.
4. When the command has been validated, the device sets -READY to inform host that it is ready to accept the first block

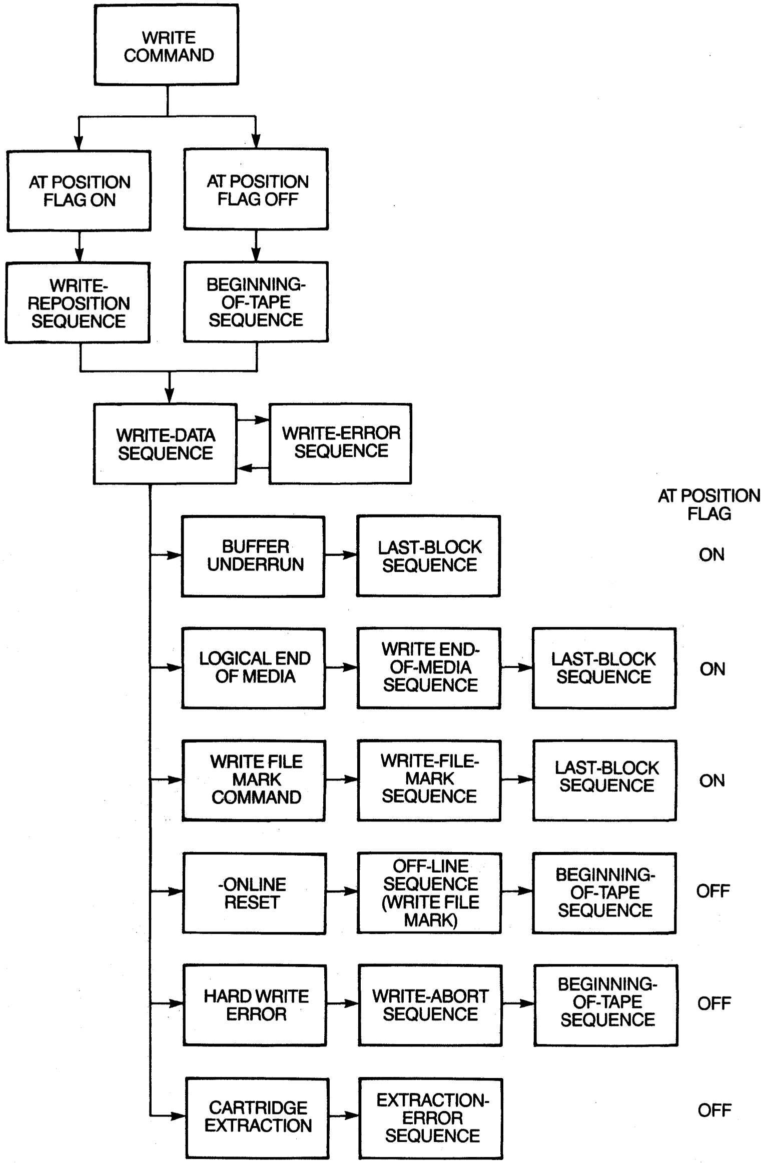


Figure 9-19 Write-Data Sequence.

of data. Host places first byte of first block on bus and sets -TRANSFER.

5. Device reads byte from bus, sets -ACKNOWLEDGE and resets -READY, which remains false until all 512 bytes of the first block have been transferred. Host removes byte from bus and resets -TRANSFER. Device resets -ACKNOWLEDGE to complete byte-transfer handshake. When 512 bytes have been transferred, device sets -READY to indicate a full block of data has been transferred and stored.

6. Device rewinds the cartridge to beginning of tape and selects Track 0.

7. Device enables the erase bar to erase the entire tape just prior to recording it. The device also enables the capstan motor to initiate tape motion and writes gap.

8. Device continues to write gap until the write head is over the certified media area (past the tape load-point hole).

9. Device begins recording blocks of data on the tape, adding gap, sync, block address and CRC. Device attempts to keep all buffers filled by initiating another block transfer as soon as a buffer becomes available.

10. As each recorded block passes the read head, the block is checked for errors.

11. After a block has been verified as correctly written with a read-after-write check, the corresponding buffer is released for further data transfers from the host.

12. Steps 9 through 11 are repeated for each data block (approximately 10,000 per track) until the device detects the tape early-warning hole. The device stops the transfer of blocks from the host and writes all buffers to tape, verifying each with a read-after-write check.

13. Device continues to erase and write a gap until the end-of-tape holes have reached the tape-hole sensors. The device disables the erase bar and stops the capstan motor.

14. Device requests and transfers the first data block for Track 1, switches to Track 1, reverses the capstan motor direction, enables the capstan motor, writes gap until the write head is over certified media (past the tape early-warning hole) and then resumes writing data blocks.

15. Unless the host discontinues data transfers, Steps 12, 13, and 14 are repeated for all additional tracks, repositioning the write/read heads when necessary.

16. Host can discontinue writing by initiating a Write-File-Mark Sequence at a block boundary when -READY is true. Host should, in this case, count 512-byte blocks, stop transferring data after the 512th byte of the final block before the File Mark, and wait momentarily for -READY to go true.

17. Host can also discontinue writing at a block boundary by resetting -ONLINE to initiate a Write-Off-Line Sequence, again counting bytes and waiting for -READY to go true.

18. If the early-warning hole of the last track reaches the tape-hole sensors before the host discontinues data transfers, the device performs a Write End-of-Media Sequence.

LAST-BLOCK SEQUENCE

A Last-Block Sequence (Figures 9-20 and 9-21) is called when there is a write underrun (no data available from the host) and

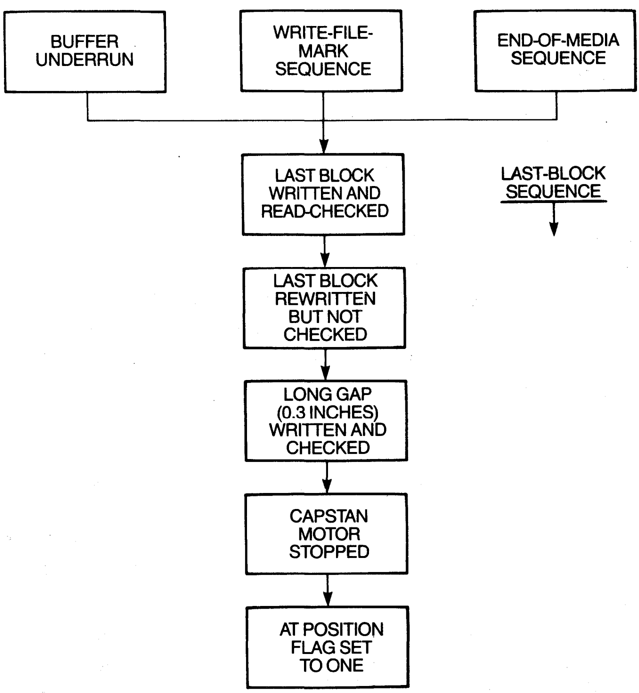


Figure 9-20 Last-Block Sequence.

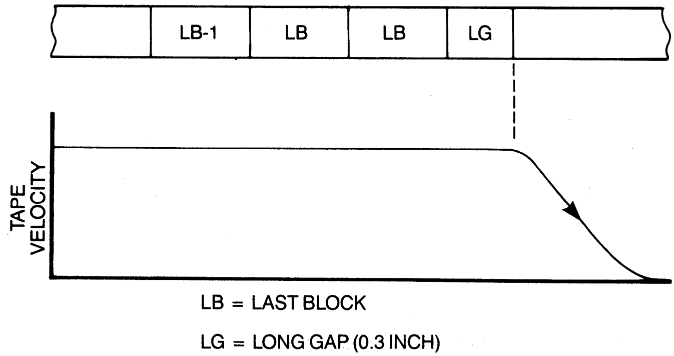


Figure 9-21 Tape Motion, Last-Block Sequence.

at the conclusion of the Write-File-Mark and End-of-Media Sequences.

1. Device continues to perform a read-after-write check as the last block is written.
2. Device finishes writing the last block and starts to write the last block again.
3. Device finishes checking the last block and starts to read but not read-check the rewritten last block.
4. Device finishes writing the rewritten last block and commences to write a long gap (0.3 inches).
5. Device finishes reading (but not read-checking) the rewritten last block and tests for 2 milliseconds of long gap.
6. Erase bar is disabled (if on Track 0), write head is disabled, the At Position flag is set to ONE, and the capstan motor is stopped.

WRITE-UNDERRUN SEQUENCE

Streaming is based on constant tape motion combined with small gaps between data blocks. The host must maintain, therefore, an uninterrupted stream of data to the device during a write operation. When writing has been initiated and a full buffer (block of data) is not available, a write-underrun condition has occurred. The response is summarized in Figures 9-22 and 9-23.

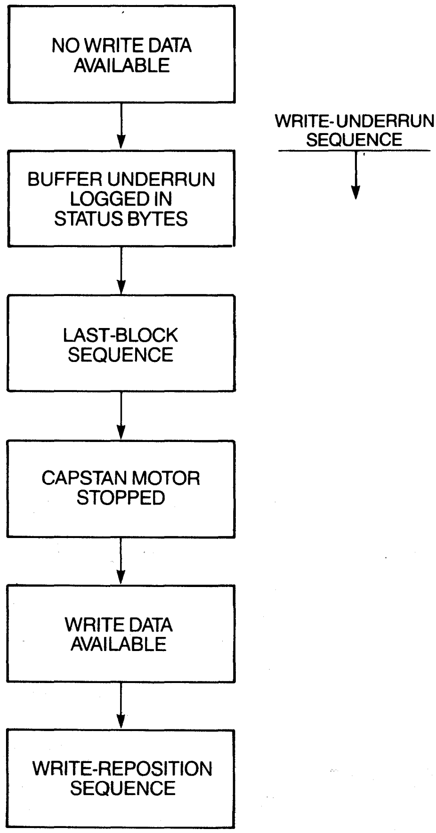


Figure 9-22 Write-Underrun Sequence.

1. A Buffer-Underrun event is logged into Status Bytes 4 and 5.
2. Device initiates a Last-Block Sequence, rewriting the last block.
3. If a full buffer of data is available before the device finishes read-checking the last block, writing continues without decreasing the system throughput by terminating tape motion.
4. If a full buffer is not available, the Last-Block Sequence is completed and the tape is stopped.
5. When a full buffer of data becomes available, writing resumes with a Write-Reposition Sequence.

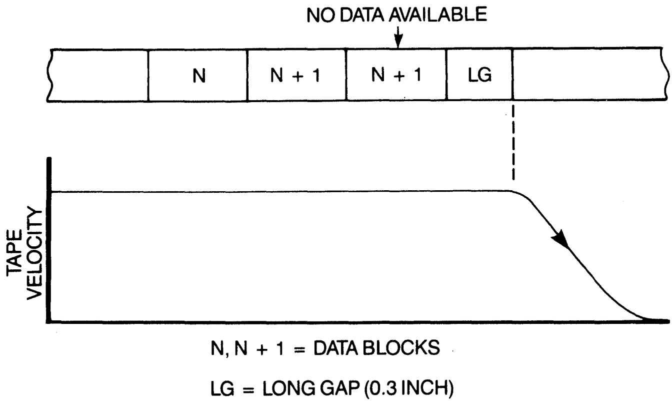


Figure 9-23 Tape motion, Write-Underrun Sequence.

A complete last block sequence requires 0.528 inch for the rewritten last block plus 0.300 inch for the long gap. A subsequent 8.0-millisecond delay imposed by the Write-Reposition Sequence adds 0.240 inch for 30-ips drives, 0.720 inch for 90-ips drives.

Each write underrun therefore "wastes" tape and reduces the tape capacity by approximately two blocks for 30-ips drives, nearly three blocks for 90-ips drives. Moreover, by terminating tape motion, data throughput is reduced severely. For two important reasons, then, write buffer underruns are highly undesireable and should be considered when designing system hardware and software.

WRITE END-OF MEDIA SEQUENCE

The Write End-of-Media Sequence (Figure 9-24) is invoked by the Write-Data Sequence when the host continues to transfer data after the last track is full.

1. Early-warning hole of the last track is detected by the controller.
2. Device stops the transfer of data from the host on the next block boundary and writes all buffers to tape, verifying each

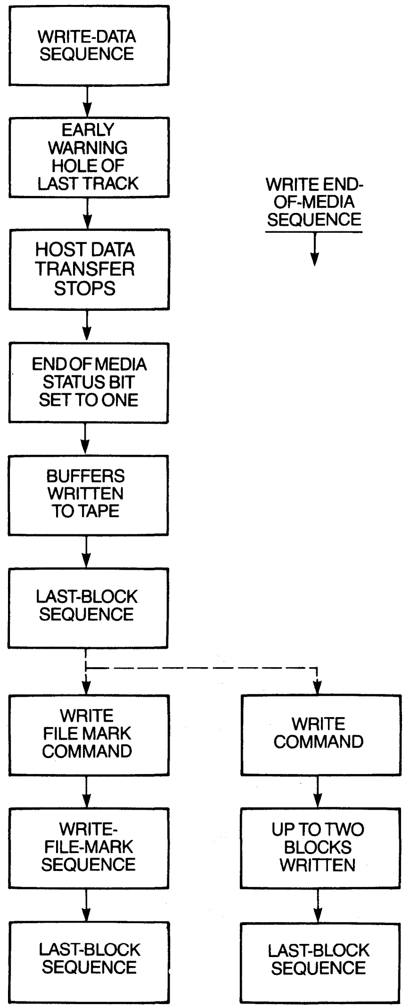


Figure 9-24 Write End-of-Media Sequence.

with a read-after-write check. A Last-Block Sequence is performed on the last block.

3. Device terminates the Write-Data Sequence, sets the End

of Media status bit, and asserts -EXCEPTION to initiate a Read-Status Sequence.

4. Device waits for the next command from the host. If the host issues a WRITE command, up to two blocks can be transferred. One or both blocks could be used to record the location of any remaining records or files. The device then repeats the End-of-Media Sequence. The cartridge is now full and further writing of data should not be attempted.

5. A File Mark can be added to the end of the record, indicating that all of the data has been recovered during a read operation, by issuing a WRITE FILE MARK command.

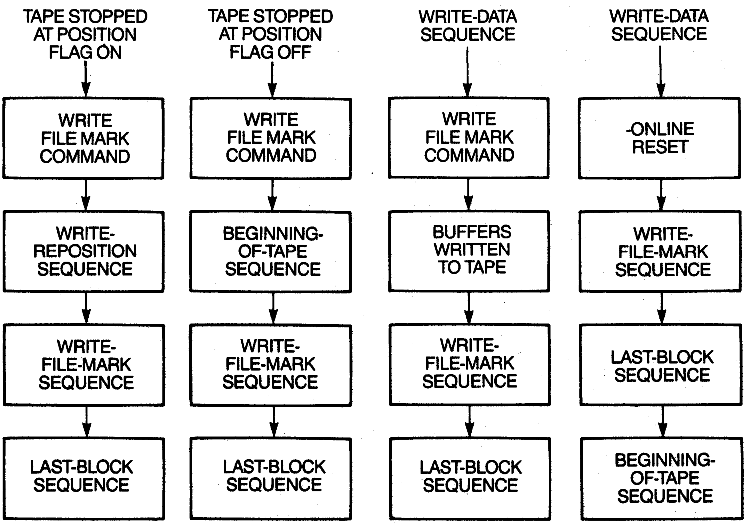


Figure 9-25 Write-File-Mark Sequence.

6. If -ONLINE is reset without a previous WRITE or WRITE FILE MARK command, the cartridge will be rewound to beginning of tape without a File Mark to terminate the record (see Read End-of-Media Sequence).

WRITE-FILE-MARK SEQUENCE

The Write-File-Mark Sequence (Figures 9-25 and 9-26) generates a standard-length data block with unique codes in the user data field, a block address, and a CRC. The host does not transfer any data for the file-mark block.

Steps 1 through 4 apply if the device is executing a Write-Data Sequence. If the device is not executing a Write-Data Sequence, the host must assert -ONLINE prior to placing the WRITE FILE MARK command on the bus and setting -REQUEST. If the At Position flag is ZERO, the device substitutes Steps 6 through 8 of the Write-Data Sequence in place of Step 4. If the At Position flag is ONE, the device substitutes steps 2 through 9 of a Read-Reposition Sequence in place of Step 4.

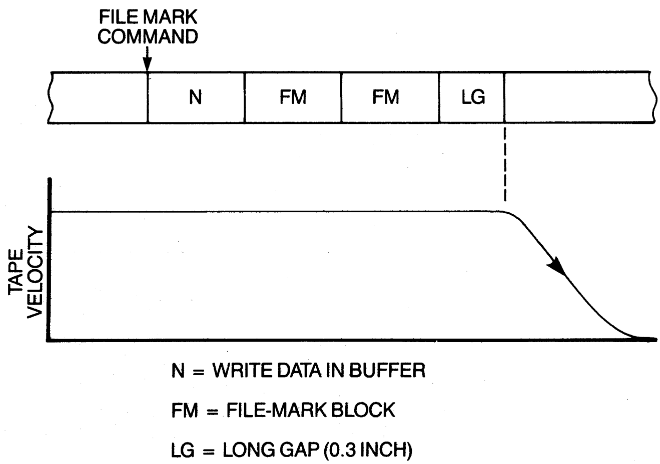


Figure 9-26 Tape motion, Write-File-Mark Sequence.

There is no restriction concerning the number of file marks that can be placed on the tape, or the number of user data blocks placed between the file marks. After issuing a WRITE FILE MARK the host can immediately issue either another WRITE FILE MARK or a WRITE command and continue transferring files to the tape.

1. Host verifies that -READY is true (typically at a block boundary, see step 16, Write-Data Sequence), places WRITE

FILE MARK command on the bus, then sets -REQUEST.

2. In response to -REQUEST, device resets -READY, reads command, and return -READY to its true state to indicate that the command has been read.

3. Host resets -REQUEST and removes command from the bus. Device resets -READY and validates command.

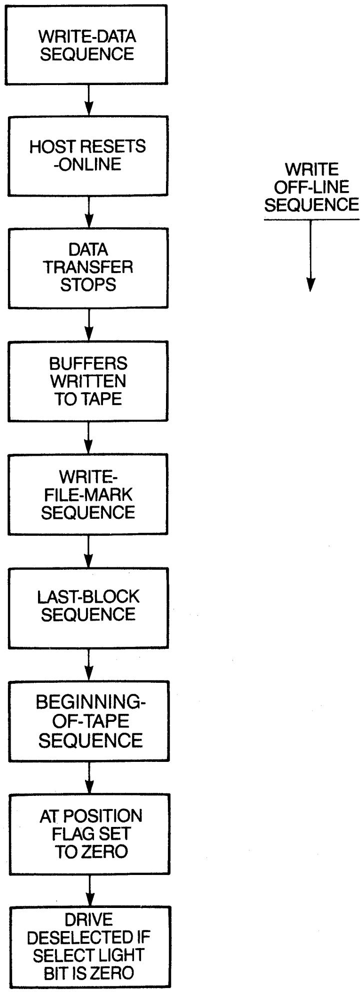


Figure 9-27 Write-Off-Line Sequence.

4. Device stops the transfer of further data from the host.

5. Device writes and verifies the remaining blocks of data in the buffers.

6. Device generates a File Mark block and writes this to the tape.

7. Device performs a Last-Block Sequence (rewriting the File Mark) and stops tape motion.

8. If the host terminates a Write-Data Sequence without issuing a WRITE FILE MARK command (e.g., by resetting -ONLINE), the device automatically generates a File Mark followed by a Beginning-of-Tape Sequence.

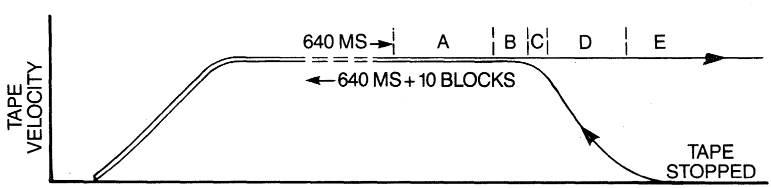
WRITE-OFF-LINE SEQUENCE

The Write-Off-Line Sequence is called when the host discontinues a Write Sequence by resetting -ONLINE at a block boundary (see step 17, Write-Data Sequence, and Figure 9-27).

1. Host resets -ONLINE.
2. Device stops the transfer of data blocks and writes remaining buffers to tape.
3. Device performs a Write-File-Mark Sequence.
4. Device sets the At Position flag to ZERO and rewinds cartridge to beginning of tape.
5. Device deselects drive if the Select Light bit is ZERO.

WRITE-REPOSITION SEQUENCE

The Write-Reposition Sequence (Figure 9-28) is called during a write operation when the At Position flag is ONE and the tape



A = SEARCH FOR LAST BLOCK WRITTEN
B = SEARCH FOR LONG GAP (0.3 INCH)
C = 1 MS DELAY
D = WRITE GAP RECORD (ALL 1'S)
E = WRITE DATA

Figure 9-28 Tape motion, Write-Reposition Sequence.

has been stopped, typically by a buffer underrun or a Write-File-Mark Sequence. It also positions the tape whenever additional data is appended to a previously written record (Figure 9-29).

1. Device requests and transfers the next data block (See Write-Data Sequence).
2. With -READY still false, device reverses tape direction and enables the capstan motor.
3. Device delays for 640 milliseconds.
4. Device delays for 10 blocks (128 blocks if more than two consecutive Write-Reposition Sequences have occurred).
5. Device stops the capstan motor, selects original direction and enables the capstan motor.
6. Device delays for 640 milliseconds.
7. Device searches for a block containing the address of the last written block.

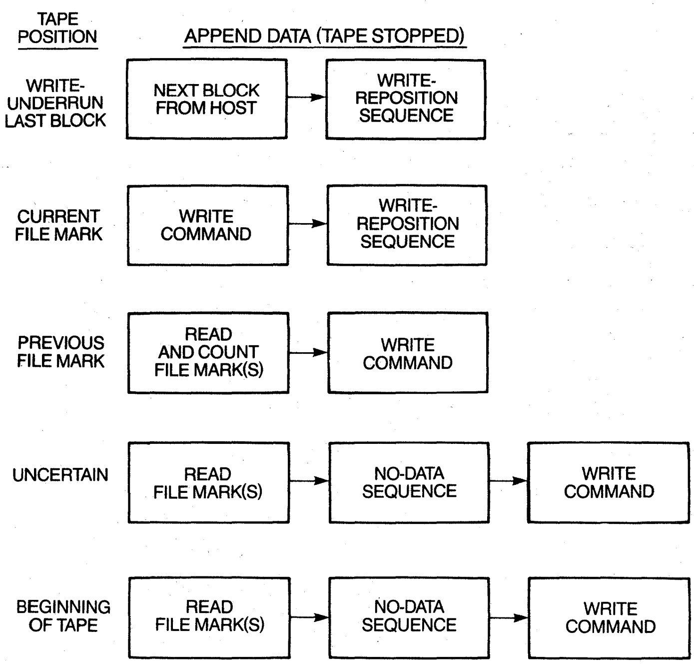


Figure 9-29 Appending data to existing record.

8. Device searches for a long-gap record and delays 1 millisecond.

9. Device starts writing a gap record and enables erase if on Track 0.

10. Device resumes Write-Data Sequence (at Step 9) by setting -READY true.

WRITE-ERROR SEQUENCE

Write and read heads are separated by a distance of 0.3 inch. The inter-block gap length is only 0.013 inch. The device must begin writing the next record before the previous record has been completely verified by a read-after-write check. Taking into account these parameters, a read-after-write error results in the following sequence (see Figures 5-18 and 9-30).

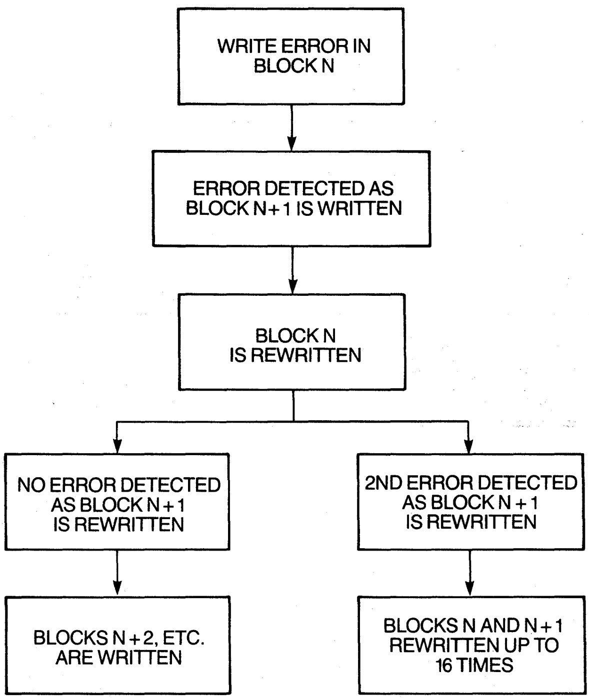


Figure 9-30 Write-Error Sequence.

1. Device writes block N.
2. When block N reaches the read head, read-checking begins.
3. Device finishes writing block N.
4. An inter-block gap is generated and written.
5. Device begins writing block N+1.
6. Device finishes read-checking block N.
7. Block N has an error.
8. Device finishes writing block N+1.
9. Device begins rewriting block N.
10. When block N (rewritten) reaches the read head, read-checking begins.
11. Device finishes rewriting block N.
12. Device begins writing block N+1 (again).
13. Device finishes reading block N (rewritten).
14. The error-correction sequence is repeated until no error occurs or until a limit of 16 consecutive same-block rewrites have been attempted.
15. Device increments the soft-error log stored in Status Bytes 2 and 3 each time a block is rewritten. Because two blocks (N and N+1) are rewritten during each error-correction attempt, the count will be generally twice the number of write errors that actually occurred.

If only one block remains to be written, a slightly altered Write-Error Sequence is performed (see Last-Block Sequence).

1. Device writes block N (last block).
2. When block N reaches the read head, read-checking begins.
3. Device finishes writing block N.
4. An inter-block gap is generated and written.
5. Device begins writing block N again.
6. Device finishes read-checking block N. If an error occured in block N, the sequence resumes at step 2 and continues until no error occurs or until 16 same-block rewrite attempts have occurred.
7. If the read-check of block N is valid, device commences to read but not read-check the rewritten last block (step 3 of Last-Block Sequence).
8. Steps 4, 5 and 6 of Last-Block Sequence are performed.

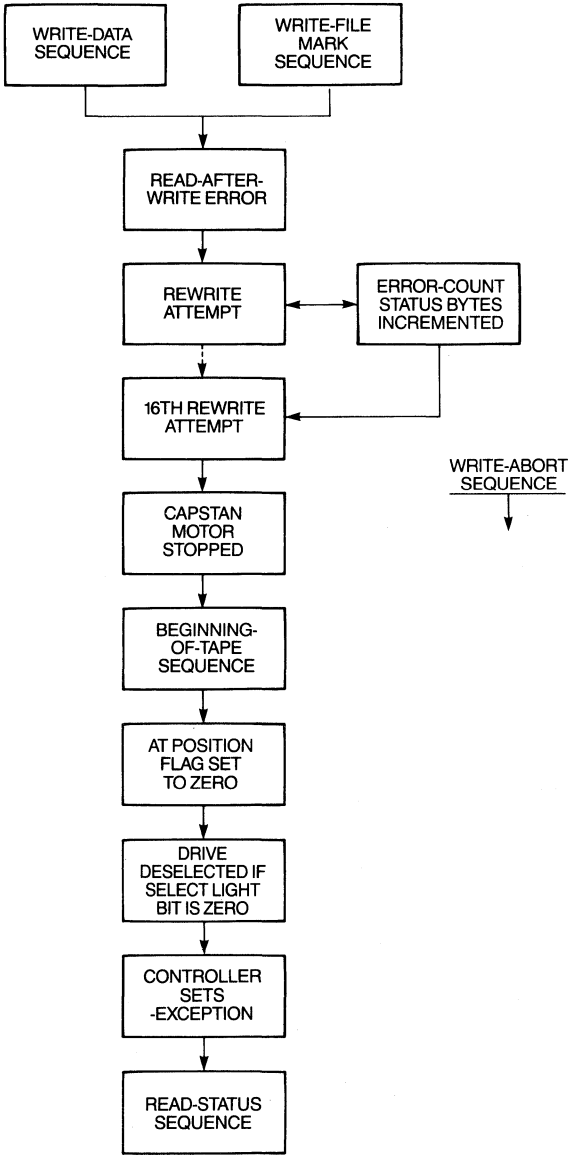


Figure 9-31 Write-Abort Sequence.

WRITE-ABORT SEQUENCE

"Soft" error rewrites are repeated 16 times before a "hard" (uncorrectable) write error is recognized. Hard errors will normally occur only in the case of equipment failure or a tape cartridge that has reached the end of its useful life. Because further attempts to write correct data would be wasteful of both tape and system throughput, device initiates a Write-Abort Sequence (Figure 9-31).

1. Device stops the capstan motor, rewinds cartridge to beginning of tape, and sets the At Position flag to ZERO.
2. Device sets the Unrecoverable Data Error status bit.
3. Device sets -EXCEPTION to initiate a Read-Status Sequence.
4. The drive is deselected if the Select Light bit is ZERO.

READ OPERATIONS

A similar but predictably different set of sequences apply to read operations:

Read-Data Sequence

Read-Underrun Sequence

Read-File-Mark Sequence

Read End-of-Media Sequence

No-Data Sequence

Read-Reposition Sequence

Read Off-Line Sequence

Read-Error Sequence

Block-in-Error Sequence

Read-Abort Sequence

Again, either one of two commands can initiate the read operation: READ or READ FILE MARK. And again the initial response, a reposition sequence or a rewind to beginning of tape, depends on the state of the At Position flag.

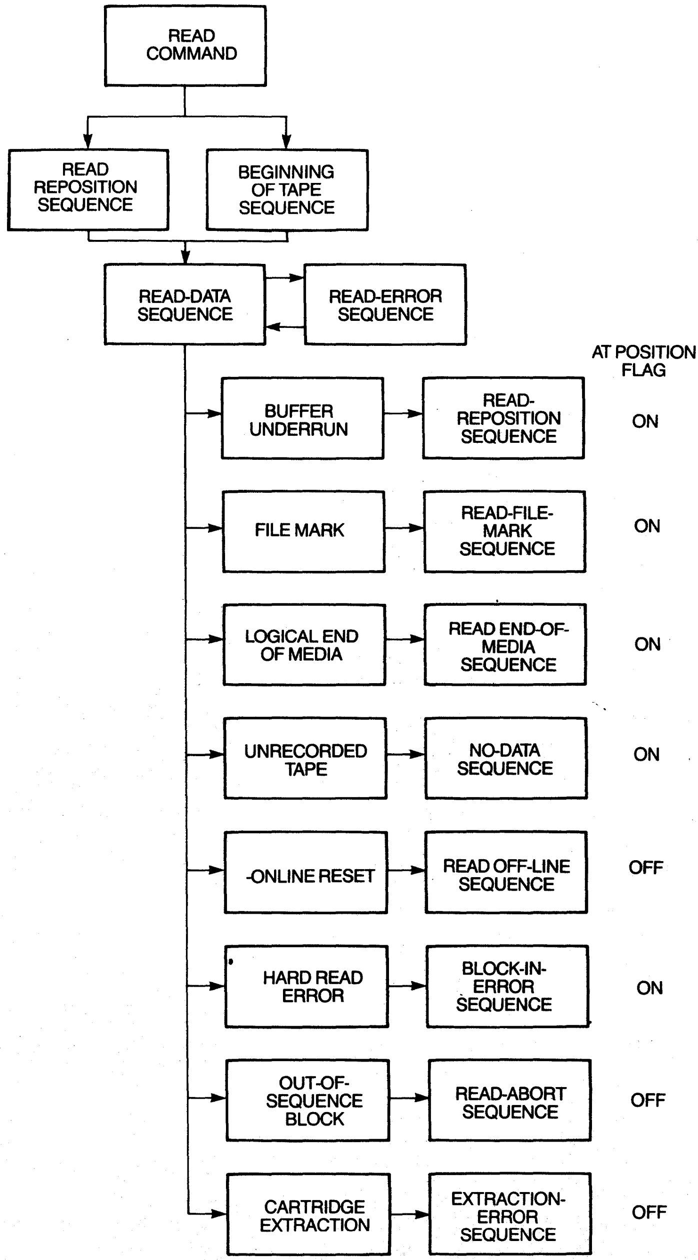


Figure 9-32 Read-Data Sequence.

READ-DATA SEQUENCE

The Read-Data Sequence (Figures 9-32) reads a formatted tape record and transfers the data to the host. A Read-Reposition Sequence replaces Steps 4 and 5 if the At Position flag is ONE.

1. Host verifies that -READY is true, places READ command on the bus, sets -ONLINE, then sets -REQUEST.
2. In response to -REQUEST, device resets -READY, reads the comand, and returns -READY to indicate it has read the command.
3. Host resets -REQUEST and removes command from the bus. Device resets -READY and changes -DIRECTION to true (device to host).
4. Device rewinds the cartridge to beginning of tape and selects Track 0.
5. Device enables the capstan motor, delays until the tape is up to speed, and searches for the first data block.
6. Device reads the entire data block and checks the CRC and block address.
7. If the CRC and block address are correct, device sets -READY to inform the host that the first block of data is ready, places the first byte on the bus, and sets -ACKNOWLEDGE.
8. Host reads the data from the bus and sets -TRANSFER. Device resets -READY (which remains false until approximately the start of the next block) and also resets -ACKNOWLEDGE. Host resets -TRANSFER.
9. Device places the next byte on the bus and signals the host that the next byte is available by setting -ACKNOWLEDGE to its true state. Handshaking continues until a complete 512-byte block has been transferred from the device to the host.
10. Steps 6 through 9 are repeated for all of the recorded blocks on Track 0. To terminate the read operation at a block boundary by issuing a READ FILE MARK command or dropping -ONLINE, the host should count 512-byte block increments, stop the data transfer, and wait for -READY to go true.
11. When the end-of-tape holes are detected for Track 0, device stops the capstan motor, selects Track 1, reverses the direction of the drive's capstan motor, enables the capstan motor and delays until the tape is up to speed.

12. All of the blocks on Track 1 are read, error checked, and transferred to the host.

13. Remaining tracks are handled in the same manner as Tracks 0 and 1, with any required track-to-track head repositioning.

14. The Read-Data Sequence is terminated by the device if a File Mark is detected. Device resets -DIRECTION and asserts -EXCEPTION, initiating a Read-Status Sequence. Tape motion is stopped with At Position flag set to ONE.

15. Host can terminate the Read-Data Sequence whenever -READY is true (see Step 10 above) by issuing a READ FILE MARK command. Device performs a Read-File-Mark Sequence, beginning with Step 6. Tape proceeds to the next File Mark without transferring any data. For example, the host may have written file-identification (ID) data in the first block or blocks of a file. If the first block or blocks indicate that the file does not contain the desired data, a READ FILE MARK command allows the host to avoid handling the remaining blocks of the file.

16. Host can also terminate the Read-Data Sequence at any time by resetting -ONLINE (see Step 10 above). Device performs a Read-Off-Line Sequence, which rewinds cartridge to beginning of tape, the At Position flag to ZERO, and deselects the drive if the Select Light bit is ZERO.

READ-UNDERRUN SEQUENCE

In a normal read operation, the device reads a block of data from the tape into one of three buffers, then transfers the data to the host. The buffers are allocated so that one buffer (block of data) is used for data being read, one for data being transferred, and the third as a reserve.

A read underrun occurs when the device has located the next block of data but none of the three buffers is available. To prevent the loss of this next block of data, the device initiates a Read-Underrun Sequence (Figure 9-33).

1. A Buffer-Underrun event is logged into Status Bytes 4 and 5.

2. Device stops the capstan motor.

3. When an empty buffer becomes available, reading resumes with a Read-Reposition Sequence.

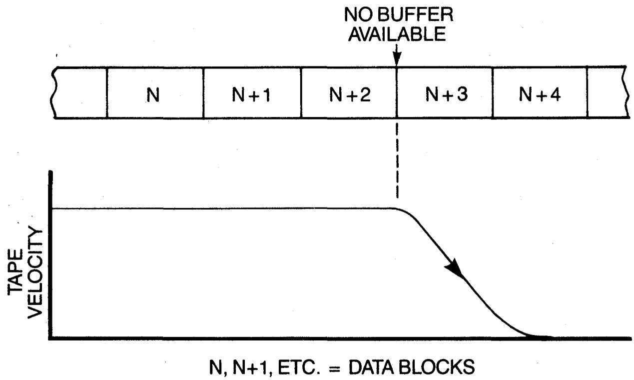


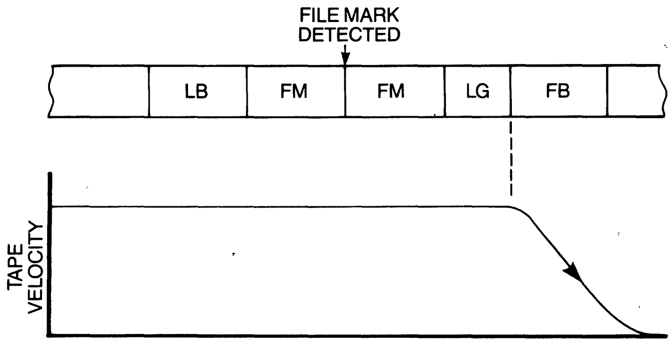
Figure 9-33 Tape motion, Read-Underrun Sequence.

Read underruns do not "waste" tape, as in the case of write underruns, but can severly reduce system throughput.

READ-FILE-MARK SEQUENCE

The Read-File-Mark Sequence (Figure 9-34) reads data but none is transferred to the host. A Read-Reposition Sequence replaces Steps 4 and 5 if the At Position flag is ONE.

1. Host verifies that -READY is true, places READ FILE MARK command on the bus, sets -ONLINE (if not already asserted), then sets -REQUEST.
2. In response to -REQUEST, device resets -READY, reads command, and returns -READY to its true state to indicate that the command has been read.
3. Host resets -REQUEST and removes command from the bus. Device completes handshake by resetting -READY.
4. Device rewinds the cartridge to beginning of tape, and selects Track 0.
5. Device enables the capstan motor, delays until the tape is up to speed, and searches for the first data block.
6. Device reads data record, searching for the first data block containing File Mark data.
7. Device terminates the Read-File-Mark Sequence if a File Mark or end-of-tape holes for the last track are detected. The



LB = LAST BLOCK OF PREVIOUS FILE

FM = FILE MARK

LG = LONG GAP (0.3 INCH)

FB = FIRST BLOCK OF NEXT FILE

Figure 9-34 Tape motion, Read-File-Mark Sequence.

device stops the capstan motor, sets the At Position flag to ONE, and informs the host by asserting -EXCEPTION to initiate a Read-Status Sequence.

8. Host can terminate the Read-File-Mark Sequence at any time by resetting -ONLINE at a block boundary (see Step 10, Read-Data Sequence). Device performs a Read-Off-Line Sequence, which in this case advances tape to the next File Mark, rewinds cartridge to beginning of tape, sets the At Position flag to ZERO, and deselects the drive if the Select Light bit is ZERO.

READ END-OF-MEDIA SEQUENCE

The Read End-of-Media Sequence (Figure 9-35) is invoked by the Read-Data or Read-File-Mark Sequence when the physical end of tape is encountered. Physical end of tape is detected when the device encounters the EOT or BOT hole pattern on the last track. The area between physical end of tape and the early-warning hole will normally have only two or three blocks of data written in this area, typically ending with a File Mark which will terminate the read operation.

1. Physical end-of-tape holes of last track are detected by the device.

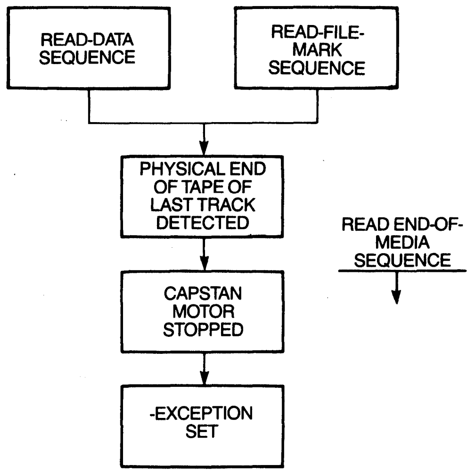


Figure 9-35 Read End-of-Media Sequence.

2. Device performs 16 attempts to read the BIE (see Read-Error Sequence).
3. Device stops the capstan motor and sets -EXCEPTION to initiate a Read-Status Sequence. Status will indicate unrecoverable Data Error, Block in Error Not Located (BIE), No Data Detected and End of Media.
4. Host can rewind cartridge by issuing a BEGINNING OF TAPE command.

NO-DATA SEQUENCE

The No-Data Sequence (Figure 9-36) serves two purposes. It automatically stops a Read-Data or Read-File Mark Sequence when there are no written records to read and transfer and there is no File Mark to terminate the record. It also facilitates a search for the end of the written record when additional files are to be added to the tape.

1. During a read operation, no recoverable blocks are found during a time period of approximately 0.6 seconds (equivalent to 32 blocks at 30 ips).
2. Device performs a long reposition sequence (see Read-

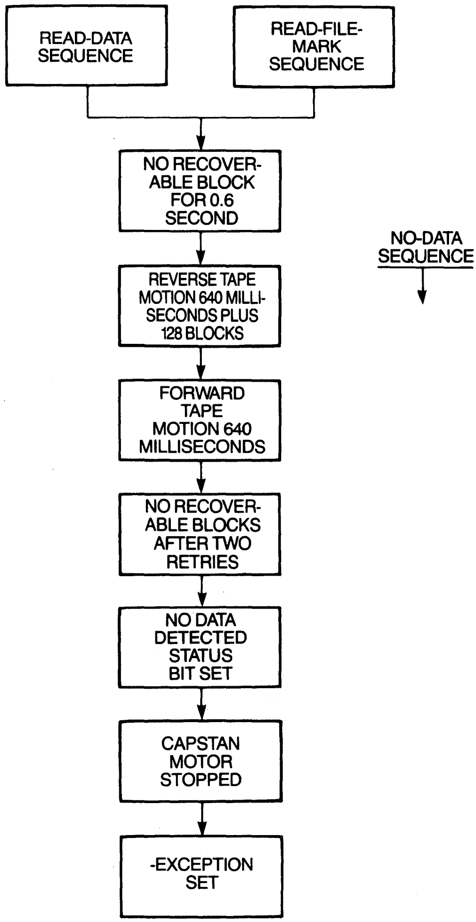


Figure 9-36 No-Data Sequence.

Reposition Sequence) and repeats for two attempts.

3. If no data blocks are found, device stops capstan motor, sets No Data Detected status bit, and sets -EXCEPTION to initiate a Read-Status Sequence.

READ-REPOSITION SEQUENCE

The Read-Reposition Sequence (Figures 9-37 and 9-38) is invoked during a read operation when the At Position flag is ONE, and the tape has been stopped.

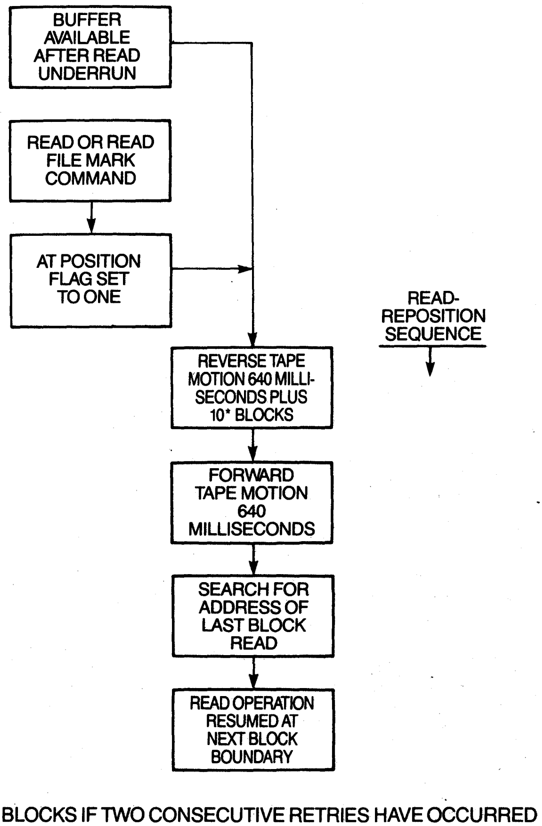


Figure 9-37 Read-Reposition Sequence.

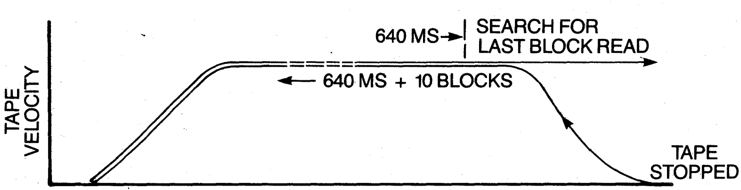


Figure 9-38 Tape motion, Read-Reposition Sequence.

1. If buffers contain data, device transfers all full blocks to the host.
2. With -READY still false, device reverses tape direction and enables the capstan motor.
3. Device delays for 640 milliseconds.
4. Device delays for 10 blocks (128 blocks if more than two consecutive Read-Reposition Sequences have occurred).
5. Device stops the capstan motor, selects the original direction, and enables the capstan motor.
6. Device delays for 640 milliseconds.
7. Device searches for the data block containing the address of the last block read.
8. Device resumes Read-Data Sequence (at step 6).

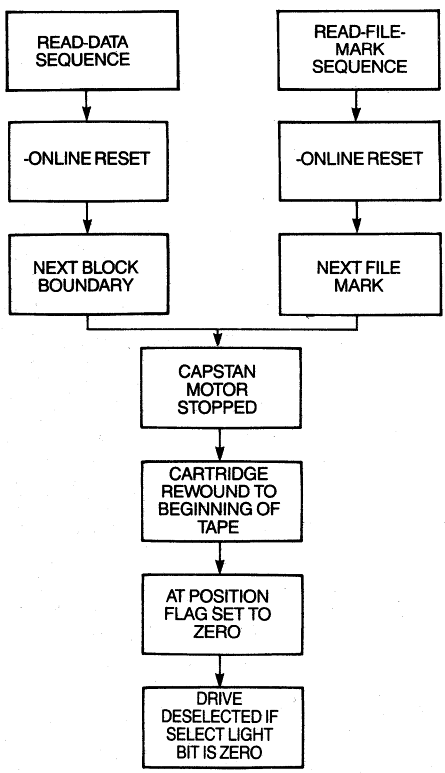


Figure 9-39 Read-Off-Line Sequence.

READ-OFF-LINE SEQUENCE

The Read-Off-Line Sequence (Figure 9-39) is called when the host discontinues a Read-Data or Read-File-Mark Sequence by resetting -ONLINE.

1. Host verifies that -READY is true and resets -ONLINE.
2. Device terminates a Read-Data Sequence at the next block boundary. A Read-File-Mark Sequence is terminated at the next File Mark. Data in buffers is not transferred to host.
3. Device sets the At Position flag to ZERO, and rewinds cartridge to beginning of tape. Drive is deselected if the Select-Light bit is ZERO.

READ-ERROR SEQUENCE

Read errors must be expected and an error recovery scheme is mandatory if a satisfactory level of data integrity is to be maintained. The Read-Error Sequence (Figure 9-40) achieves this goal by rereading a block-in-error (BIE) sixteen times before informing the host of an unrecoverable read error.

1. Device reads the next data block following the BIE.
2. If the next data block contains no CRC error and the same block address as the BIE, then the BIE was rewritten during the write operation. This is an expected condition (see Last-Block

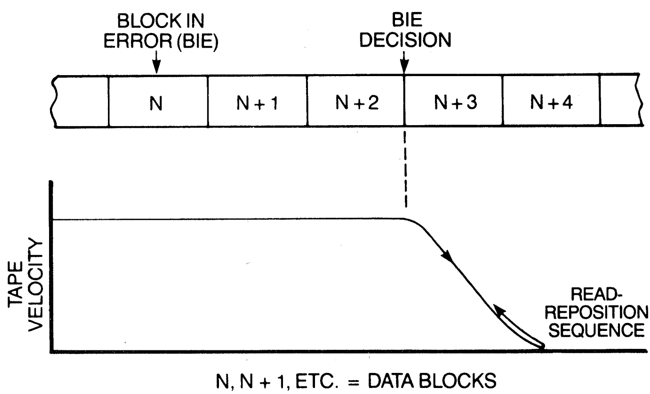


Figure 9-40 Tape motion, Read-Error Sequence.

Sequence) and is invisible to the host. Device continues the read operation without informing the host.

3. If the next data block contains no CRC error and a block address 1 greater than the BIE, the BIE may or may not have been rewritten. This is also an expected condition (see Write-Error Sequence) and is invisible to the host. The device continues the read operation without informing the host.

4. If the next data block contains no CRC error and a block address 2 or more greater than the BIE, then the BIE was not rewritten. The device performs a read-error retry by stopping the capstan motor and initiating a Read-Reposition Sequence.

5. The device increments the read error log stored in Status Bytes 2 and 3 the first time a read-retry is performed for a given block.

6. If the data block has not been successfully recovered after eight read-error attempts, the device sets the Eight or More Read Retries bit in Status Byte 1.

7. If the data block has not been successfully recovered after 16 retries, the device transfers the BIE (see Block-in-Error Sequence) if it can be located (see Read-Abort Sequence), terminates the read operation, and informs the host of an unrecoverable read error by setting -EXCEPTION to initiate a Read-Status Sequence.

BLOCK-IN-ERROR SEQUENCE

The Read-Error Sequence is repeated 16 times before a "hard" (unrecoverable) read error is recognized. Hard read errors are generally indicative of equipment failures or tape wear. To avoid a further reduction in system throughput, the device initiates a Block-in-Error Sequence (Figure 9-41).

1. Device transfers the data resulting from the 16th attempt to read the BIE to the host.

2. Device stops the capstan motor, sets the Unrecoverable Data Error status bit, and sets -EXCEPTION to initiate a Read-Status Sequence and inform the host that the last block transmitted contained an error. (Host may then perform a more sophisticated error detection and correction procedure.)

3. If the device is uncertain whether the transmitted data represents the BIE, it also sets the Block-in-Error Not Located

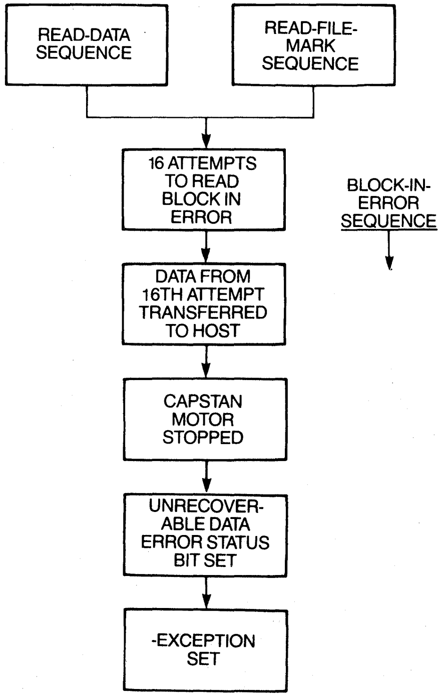


Figure 9-41 Block-In-Error Sequence.

bit to indicate that the block is "filler" to maintain the address sequence.

4. If the host wants to continue the read operation, it must continue to assert -ONLINE and reissue a READ or READ FILE MARK command. The device will start the read operation with the next block following the BIE.

5. If -ONLINE is reset following the BIE status report, the cartridge is rewound to beginning of tape (see Read Off-Line Sequence).

READ-ABORT SEQUENCE

Any recorded or reading error in the one-byte block address is treated as a conventional write or read error. However, such errors will generally disrupt the orderly sequence of block

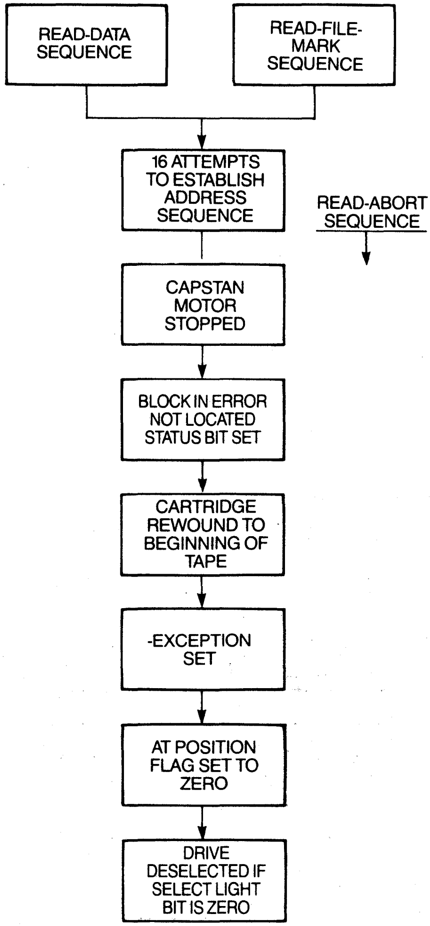


Figure 9-42 Read-Abort Sequence.

addresses required by the controller when it is performing a Read-Reposition Sequence to reread a BIE. If for this or any other reason, the device is unable to relocate the BIE or reestablish the address sequence, it initiates a Read-Abort Sequence (Figure 9-42).

1. Instead of transferring the data resulting from the 16th attempt to read the BIE, the device sets the Block-in-Error Not

Located, Unrecoverable Data Error, and Beginning of Media status bits and sets -EXCEPTION to initiate a Read-Status Sequence.

2. Device rewinds the cartridge to beginning of tape, and sets the At Position flag to ZERO. The drive is deselected if the Light Select bit is ZERO.

ARCHIVE

ARCHIVE CORPORATION
3540 Cadillac Avenue
Costa Mesa, California 92626
(714) 641-0279

ISBN 0-9608810-0-X