Intelligent disk drive for the 1980's

MAARTEN PRANGER, Storage Technology Corp.

This micro-based device could be the harbinger of a whole new class of intelligent peripherals.

Ever since the microprocessor first became available, disk designers have paid close attention to its improvements. The first microprocessors were too slow—at a few hundred thousand operations a second—to handle disk applications. Today, however, micro operating speeds have soared above two megahertz, and this opens the door to a new type of micro-based product—the "intelligent disk."

Such a device has now appeared on the market for the first time with the introduction by Storage Technology Corp. of a model 2700 disk drive that incorporates a micro in the drive itself (Fig. 1). Of course, micros have been used in disk drives before, but the model 2700 is the first to exploit the micro's power to turn the disk drive into an intelligent peripheral. Indeed, I believe that the STC 2700 is the forerunner of a new generation of intelligent disk drives that will dominate the market in the 1980s. This article examines the new STC drive to show how such devices will improve the performance of computer systems and reduce their costs.

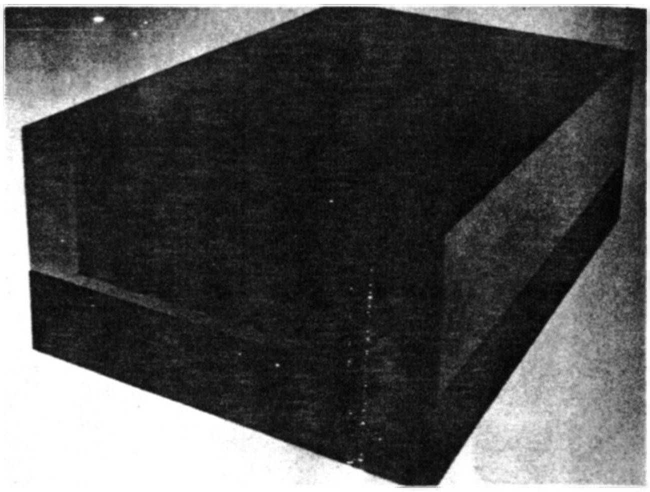

Fig. 1. The model 2700 *disk drive by Storage Technology Corp.*

A basic 2700 series system includes a microprocessor, 64K bytes of dynamic RAM, an RS-232C interface, a floppy disk interface, two I/O ports, and a fixed disk drive (Fig. 2).

The heart of the new 2700 disk drive is a Motorola 6801 microprocessor. An enhanced version of the 6800, the 6801 has ten additional instructions, including a multiply instruction that enables the 2700 to use relatively sophisticated algorithms to control the disk's servo mechanisms. Other hardware enhancements on the 6801 include:

- an onboard 2048 byte ROM that is used to initialize the 2700,
- an onboard timer used by the servo control programs and as the basis of a real-time clock in the device,
- a serial line unit that has enabled us to implement an RS-232-C interface as a standard feature. The RS-232-C interface gives the 2700 a very powerful diagnostic capability and makes the unit easy to service either locally or from a remote service facility.

Incorporating a micro in the 2700 has enabled us to eliminate a key element in today's disk subsystems—the controller. Most disk drives on the market are dumb devices, and hence require a separate controller to supply the operating intelligence. Such controllers have some big drawbacks. For one, they are expensive, ranging in price from about $2000 to more than $10,000. Moreover, the typical controller can operate only one disk drive at a time. This means that while one drive is transferring data to the computer, all the other drives connected to the controller must stand idle. (Some controllers do allow functions, such as head seeks, to run at the same time on different drives, but this is a very inefficient solution compared to the completely independent operation possible with the STC 2700.)

With the 2700, we have moved about 80 percent of the functions found in today's controller inside the drive itself. This is tantamount to supplying a controller with each drive at no increase in cost.

The STC design eliminates most of the functions of conventional controllers.

Since the STC design eliminates most of the functions of conventional controllers, we now call them adapters. Such adapters are much simpler than the conventional controller. For example, an adapter for a relatively difficult bus structure, such as the DEC PDP-11's, requires less than 100 IC packages to implement. This compares to about 200-300 ICs for a typical PDP-11-compatible disk controller. The adapters for many other computers can be configured with as few as 40 ICs if DMA is not required.

The 2700 design derives other benefits from the use of a microprocessor. For one, we were able to eliminate all potentiometers in the 2700 by using the micro to adjust circuit parameters. This helped to boost performance. For instance, we were able to increase data integrity by having the microprocessor adjust circuit parameters in the read channel.

Savings in CPU time

Besides handling control functions, the 2700's micro can perform many data processing tasks now done on a system CPU. This can result in a significant savings in CPU time.

Consider, for example, the editing function. Done with a conventional disk, the editing process is very inefficient. The system must transfer data from the disk to the host CPU, modify the data, and then transmit the edited data back to the disk subsystem. With the 2700, on the other hand, a user can invoke an application package that accepts and executes editing instructions in the disk itself. The system bus is tied up only as long as it takes to transmit the instructions; there is no need to transmit the data base to the computer and back. Approximately 90 percent of the resources typically allocated to the editing task are left free. As a result, both the bus and CPU can turn to other jobs.

Merging data files is another application where operations can be carried out completely at the disk level, with no CPU resources required after the initial command sequence has been

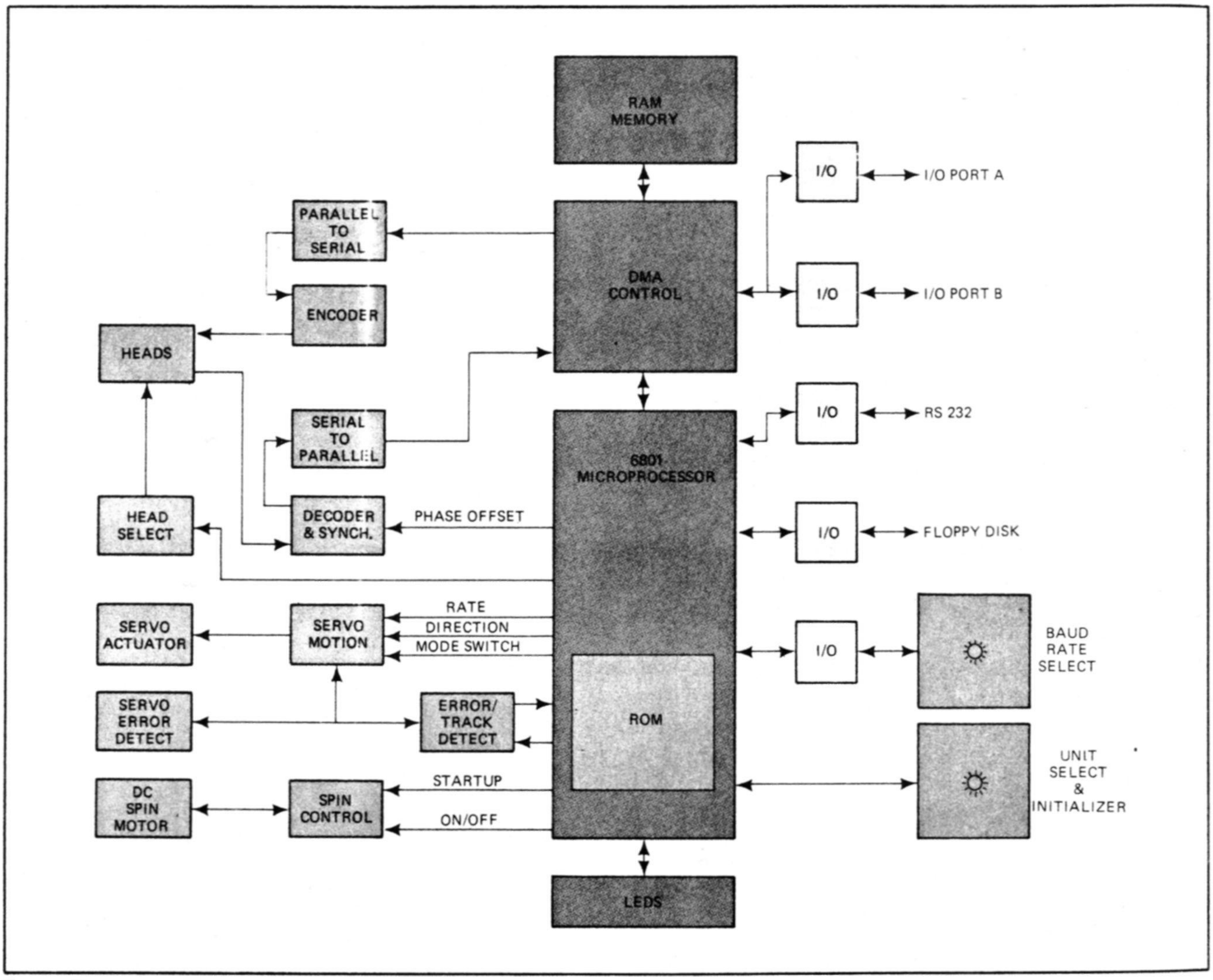

Fig. 2. The model *2700 subsystem.*

The 2700's micro can perform many data processing tasks now done on a system CPU.

transmitted to the drives. In conventional systems, each block of data would have to be transmitted to and from the CPU. Approximately 99 percent of the CPU and bus resources are saved in this operation.

To enable users to customize the 2700 in an orderly fashion, we've programmed the 2700's micro in a high level language—a subset of PL/1. This enables users to add capabilities such as file editing or file merging to the disk drive by reprogramming the drive's existing algorithms or adding new algorithms. In brief, the high level language makes it easy for a user to tailor the 2700's run-time software to his application.

Another innovation

The use of a micro is not the only innovation on our new drive. The 2700 employs a high-speed, byte parallel interface instead of a serial interface as in current drives. Obviously, this byte handling capability is intended to enable the 2700 to move more data in a given time than it could with a serial interface. But it also has a more subtle purpose. Let me explain.

Since their inception, the storage capacity of disk drives has doubled every year or two. This increased capacity has been obtained by increasing linear and/or radial density. Increasing linear density causes an increase in data rate. Thus, a new design based on a serial interface would limit data rates and hence data capacities for follow-on products. Since we expect the trend to larger data capacities to continue, we decided to use a byte parallel interface to assure that the 2700 design will remain competitive.

Bus architecture

The 2700's basic interface bus comprises 15 lines: eight data lines, an odd-parity line, and six

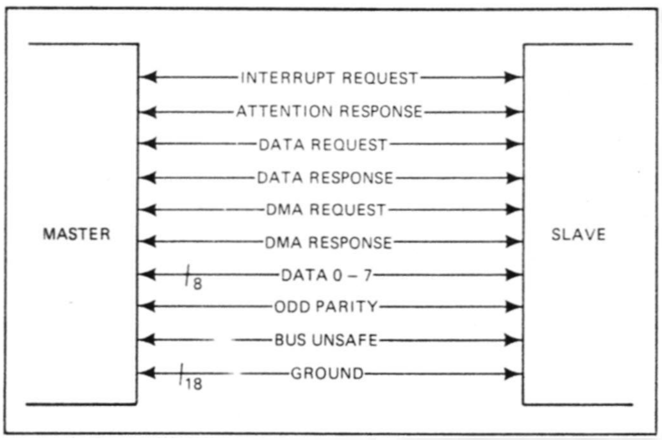

Fig. 3. Interface signals *used to link the model 2700 to other devices.*

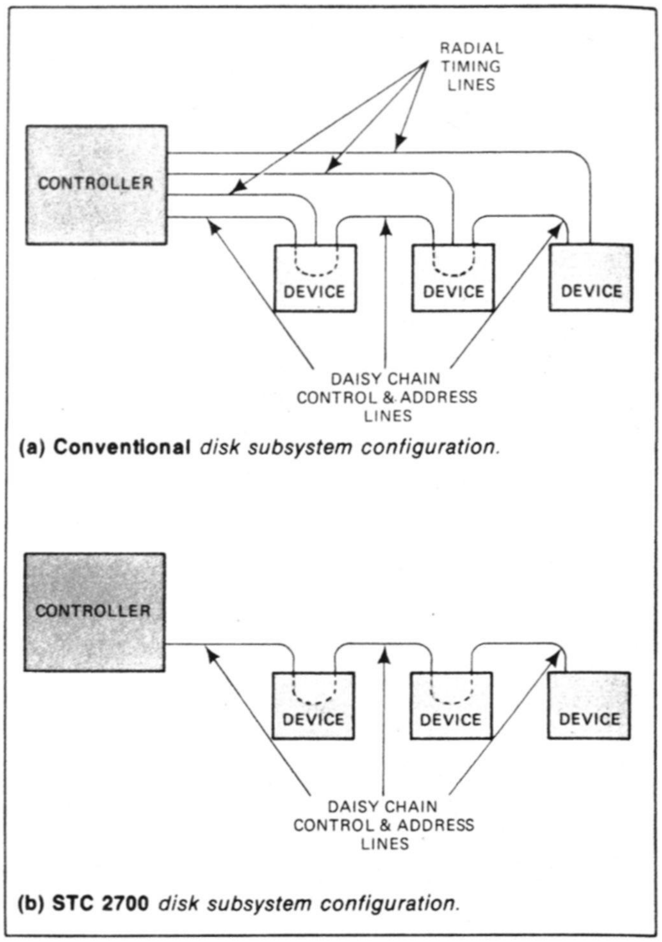

Fig. 4. A conventional disk *subsystem (a) requires both radial and daisy-chain cables to interconnect drives. A 2700-based system (b) requires only the daisy-chain cables.*

control lines (Fig. 3). A sixteenth line is used to notify all units on the bus that the bus is potentially unsafe. For example, the line issues a warning signal if a unit has lost power, or if a cable has been plugged in upside down. In addition, the 2700 interface includes 18 ground lines that enable a user to hang three or more units on the same bus in a daisy-chain configuration.

Conventional disk subsystems use both radial and daisy-chain cables to interconnect drives (Fig. 4a). This requires two cables per device. The 2700's interface bus supports both conventional hardware hookups—radial and daisy chain. However, the 2700 design eliminates the need to transmit data signals directly with a radial cable to the controller. As a result, only one cable per device is required to interconnect 2700 drives (Fig. 4b).

Master/slave

The 2700 uses a handshaking protocol to control the operation of devices on the interface bus. This permits a master/slave type of operation. For example, to transfer data to a slave, the master device (normally the disk adapter) sends out a request pulse simultaneously with data

(Fig. 5a). The selected slave device then samples the data and ackowledges its receipt by issuing a response signal. When the master detects this pulse, it drops its own request pulse and removes data from the bus. The slave device then terminates its response signal.

When a slave transfers data to a master, the master device first issues a request pulse (Fig. 5b). When the selected slave device replies with a response signal, the master device samples the data lines and then turns off its request signal. The slave device then terminates the data and response signals.

Any device on the 2700 interface bus can be a master or slave device. Normally, the host CPU adapter is the master device and the drives are slaves. However, any 2700 can become the bus master and transmit data to another 2700 device. For example, drives can transfer data on the non-adapter bus shown in Fig. 4b. This greatly enhances system throughput for or y the cost of the additional cables.

Other advantages

The 2700's handshaking protocol has other advantages. For one, it enables a master to interrupt processing at any time and perform another function. A master device, for example, can suspend a DMA operation currently in progress simply by failing to re-issue a DMA request pulse. As soon as the slave unit completes the DMA cycle, the master can change units, sample unit status, or perform other operations elsewhere in the system. The master can then reselect the interrupted unit, issue a DMA restart instruction, and continue from the point of interrupt.

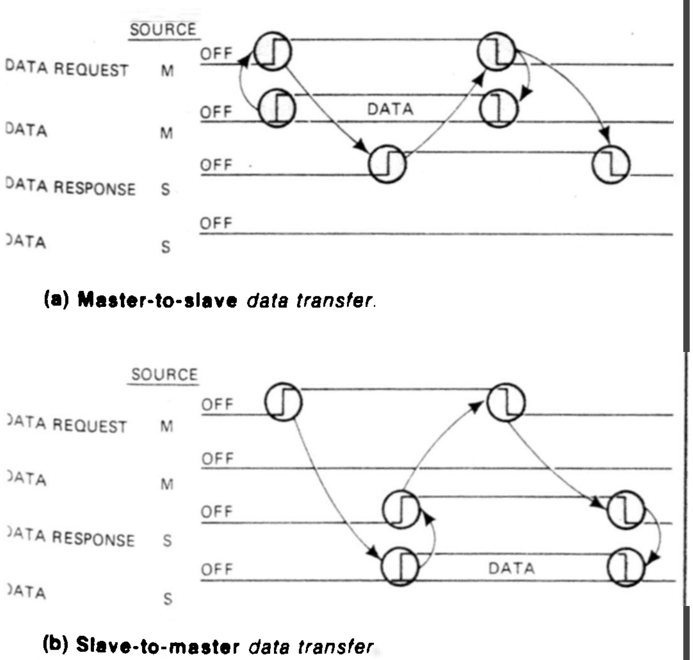

Fig. 5. The model 2700 *uses a handshaking interface protocol to control devices on the bus*

All this makes handling multiple devices much more efficient than on conventional drives where the system talks directly to the disk's read/write head. With such devices, there is no way to interrupt a data transfer once it has started other than stopping the disk, which would be impractical.

The handshaking protocol also has a speed advantage. The model 2700 supports data rates as high as 2M bytes/sec. on an instantaneous basis. But because the 2700 design uses a handshaking type of protocol, it is by no means limited to that data rate. In fact, the handshaking interface is physically capable of handling data rates as high as 10M bytes/sec. To achieve this speed, however, follow-on products will require faster logic. As a result, we expect future STC products to use new semiconductor technologies, such as silicon on saphire (SOS), that promise to yield the requisite speeds.

The question naturally arises: Will such speed be needed? We believe the answer is yes. Today, head/disk mechanisms typically transfer data at a rate slightly under 10M bits a sec., the rate being a function of disk speed and linear data density. However, we expect future disk products to have higher bit densities and hence higher transfer rates.

Dual ports

Dual I/O ports are an option on most conventional drives. The model 2700, on the other hand, incorporates I/O ports as a standard feature. The reason? We feel that data transfer rates on disk drives have become so high that dual ports are now almost a necessity.

The advantage of dual ports (Fig. 6) is that they enable different CPUs to address the same disk drive. In addition, they can be used to enable different tasks in a CPU to address the same drive via separate I/O paths. This can reduce CPU overhead, since the CPU no longer has to arbitrate demands for disk resources by the two tasks.

Conventional disks that have dual ports typically allow only one of the ports to be used at any given moment. In contrast, the ports on the model 2700 do not exclude each other and hence both can operate simultaneously. The drive never makes one port busy in favor of another unless a customer has invoked a port priority option. In short, the 2700 can simultaneously transfer data through both ports and at the same time read or write data to the disk media itself.

Software interface

The 2700 has a very powerful set of I/O commands. The set includes a basic R/W com-

The 2700 has a very powerful set of I/O commands.

mand sequence that enables a user to access the device as he would a conventional disk drive. In addition, the set includes generalized commands that enable the user to transfer data from any storage or I/O area in one device to any storage or I/O area in the same device or any other device on the bus. For example, a user can copy a sector from one area of a disk to another, or to a floppy as a backup operation. Moreover, the data transfer can vary in length from one to 64K bytes.

The 2700 includes a command stack that enables all sorts of operations to be initiated by a single command. Additional commands can be entered in the stack prior to execution, and even during execution. Also, prior to execution, the current command stack can be scanned and the commands executed in an optimal order.

The model 2700 uses a 64K-byte dynamic RAM to buffer data being transferred to or from the disk drive. This has several advantages. For one, it enables data rates to-and-from disk media to be different from data rates to-and-from their CPU, thus optimizing the efficiency of the host CPU bus.

The RAM also helps to boost throughput via a process that we call "intelligent anticipation of data." Whenever the system is idle, it will obtain and load into memory the data following the last data read. Normally, this is done during the same revolution as the one during which data was acquired. This results in placing an entire track of data into RAM. Since many disk files—for example, those used in ledger-type work—tend to be sequential in nature, this feature should increase the throughput of read operations significantly. If a user has a different anticipation algorithm for his particular system, he can modify the 2700 software to implement the algorithm.

Technology independent

The integral RAM has also enabled us to build a technology-independent disk. That is, it enabled us to configure the software and hardware interfaces in such a way that the CPU does not know it is talking to a disk drive. As far as the CPU is concerned, it could just as easily be talking to a tape drive, bubble memory, or any other type of peripheral. Now, users no longer need be concerned about generating signals to satisfy write gates, read gates, and all the other control signals typically associated with disk drives.

Error handling

The integral RAM has yet another advantage. Since the advent of IBM 3350 disk technology, track and bit densities have reached such a level that media manufacturers can no longer produce defect-free media. As a result, drive manufacturers have had to address the problem of detecting and correcting media-induced errors. The conventional approach has been to handle media defects in disk controllers or in the CPU. The 2700's integral RAM enabled us to do the error handling in the drive itself. The 2700 loads data into RAM, makes corrections, then passes the corrected data on to the user.

To handle media defects, we reserve a 256-byte block, designated a system sector, on each track of the drive. When the 2700's software encounters a defect on a data sector, it moves the data in the defective sector to a system sector. The system sector space is used for other functions besides handling media defects. Those functions include diagnostics, failure logs, and configuration control. The entire overhead consumes 10 percent of available raw disk space. Thus, regardless of the number of defects, 90 percent of usable disk is always available to the user.

The 2700 contains a large amount of software. To manage this software, we have placed an operating system inside the disk itself. In normal mode, the OS handles power up, loads the run time program into RAM, and then transfers execution to the loaded program. The OS supports task queuing and task optimization as a standard feature. In addition, it assigns data buffers in real time to assure maximum utilization of RAM. ■

For More Information, Circle No. 451 on Inquiry Card

MAARTEN PRANGER is manager of OEM disk subsystems at Storage Technology Corp. in Louisville, CO. Mr. Pranger was responsible for the design of STC's new 2700 series disk drives.