SPECIAL REPORT ON DISK MEMORY

INTERFACE PAVES THE WAY TO HIGH PERFORMANCE

Removing the restrictions placed by existing interfaces on 5¼-in. hard disk transfer rates, the Enhanced Small Device Interface clears the way for higher recording densities.

by Frank Seuberling

During the past year, an interface has emerged that will not only handle today's mass storage needs, but will also expand to meet foreseeable future needs. The Enhanced Small Device Interface is designed to handle a variety of 5¼-in. Winchester disk and tape drives. It opens the door to higher performance system designs by incorporating more intelligence into the drives, and by allowing higher data transfer rates—10 Mbits/s and above. Thus, it allows higher recording densities and, ultimately, higher storage capacities in the 5¼-in. form factor.

By facilitating the design of multiple device controllers, the Enhanced Small Device Interface (ESDI) will allow system designers to mix and match disk and tape devices to meet designer needs on a single controller, independent of the system bus (Fig 1). Microprocessors are used in both the control units and the disk/tape drives. Thus, firmware can accommodate different command structures, and connector pin designations will be the same for both disk and tape devices.

The single controller provides several advantages for the system designer. First, the host need not provide buffer space in backup operations. (Other configurations require buffer space because disk and tape devices rarely operate at the same data rate.) Second, the system bus is not tied up during data transfer. Using ESDI, a minimum number of commands may be sufficient to back up an entire disk,

Frank Seuberling is a development engineer at Maxtor Corp, 61 Daggett Dr, San Jose, CA 95134. He was responsible for the digital design ESDI, and for developing the microprocessor code for interface and servo control. Mr Seuberling holds a BS in physics and a BA in mathematics from Humboldt State University, Arcata, Calif.

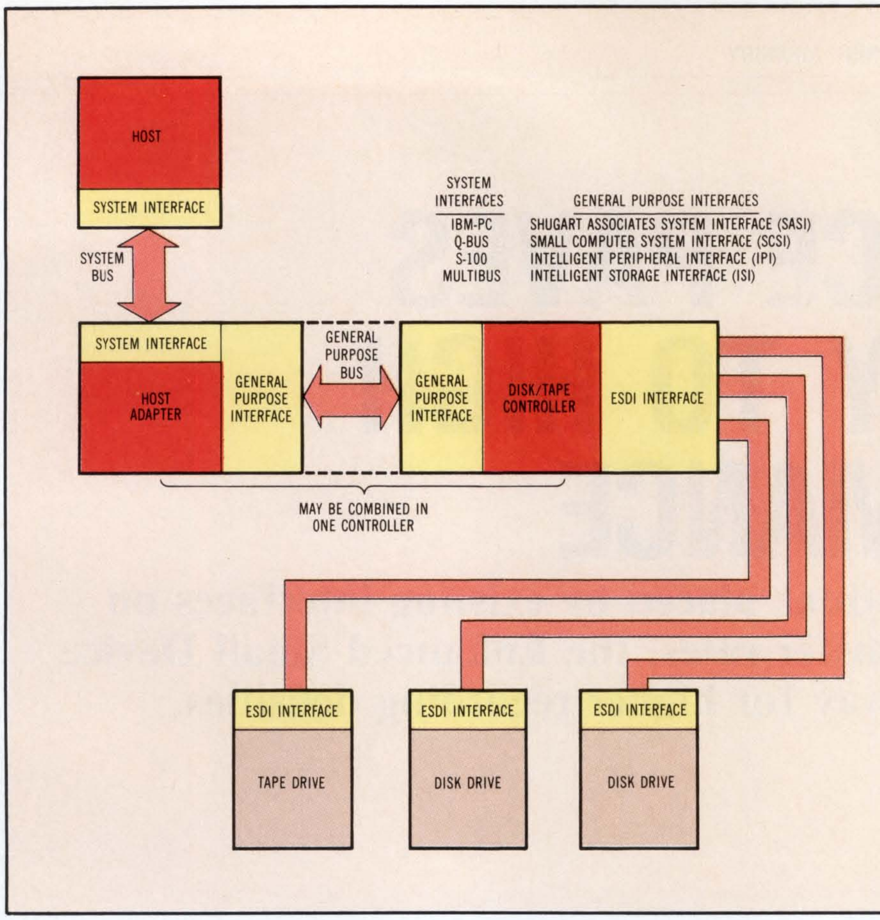

Fig 1 The Enhanced Small Device Interface (ESDI) defines the connection of disk and tape devices to the controller; it does not define the controller-host interface. Thus, the ESDI can be configured into systems using a variety of general purpose and system interfaces with a minimal impact.

thus freeing the system to handle other tasks. Overall, putting both disk and tape drives on a single, low cost controller solves the problem of designing in cost-effective backup for high capacity, high performance systems. It also provides for program distribution via the tape drive.

Limitations of the existing standard

The *de facto* industry standard interface for 5¼-in. Winchester disk drives has been Seagate Technology's ST506/412 interface. This interface was originally designed for low performance 5- and 10-Mbyte disk drives that used stepper motor head positioners and transferred data at a 5-Mbit/s rate. These relatively low performance, low cost drives provided an excellent upgrade for floppy-based systems. For this reason, this interface in some ways resembles a floppy disk drive interface.

The advent of 16- and 32-bit microprocessor-based systems, however, has caused an increase in the demand for higher storage capacity in smaller packages. Since the ST506/412 interface served as the 5¼-in. *de facto* standard, it was the obvious first choice for a higher transfer rate interface design. However, considering the capacity and performance desired, it has two serious limitations. It assumes a stepper motor positioning system, and it limits the data transfer rate to 5 Mbits/s.

There are three ways to increase capacity on a disk storage device: increase the track density, measured in tracks/in.; increase the flux reversals/in. (FRPI); or increase the recording area (number of platters) in the package. All three methods pose problems with the ST506/412 interface.

Increasing the FRPI increases the transfer rate, thus putting the device in immediate violation of the ST506/412 standard. Increasing the number of disks also violates the standard since it allows only eight heads to be selected. Furthermore, since open-loop positioning systems (usually stepper motors) typically have an upper limit of less than 500 tracks/in., increasing track density requires that a more accurate closed-loop positioning system be used. Closed-loop servo systems generally employ a voice coil actuator. Since the ST506/412 is designed to operate as a stepper motor interface, the full performance benefit of the drive will not be realized. Increasingly, as drive designers turn toward thin-film heads, thin-film media, and closed-loop, track following positioners to provide high capacity, this interface imposes a limitation on performance.

These limitations induced a group of controller and disk drive manufacturers to meet, beginning in October 1982. Addressing the current and future interface needs of the industry, in March 1983, the meetings resulted in the Enhanced Small Disk Interface. In August 1983, another series of meetings between controller and tape drive manufacturers brought forth the Enhanced Small Tape Interface. In November 1983, these interfaces were combined

ESDI Controllers under Development

Manufacturer	Model	Interface supported
Adaptec	—	
Concept Development	—	SCSI
DSD	—	Multibus
DTC	500C series	Multibus
Distributed Logic	—	SASI
Emulex	SC05	Q-bus
Interphase	unknown	Q-bus
Konan	DGC-100	IBM PC
	DJ	S-100
	Taisho	SASI
OMTI	6100/6300	Multibus
Xylogics	421	SASI
		Multibus

into one specification, ESDI, to provide a system solution to disk storage and tape backup for both current and future needs. Controllers implementing the interface are currently under development by a number of vendors (see Table).

The cable used to connect the disk drives and tape drives to the ESDI controller is the same as that used with the ST506/412 interface. A control cable is daisy chained to all devices connected to the controller, and carries signals to the drives and multiplexed signals from the selected drive. In addition, each device is connected to the controller through its own radial data cable.

Defining the ESDI

The ESDI allows two modes of implementation—step mode or serial mode. Either step mode or serial mode can be used for disk drives, while only serial mode can be used for tape drives. Very similar in definition to the ST506/412 interface, step mode is designed for stepper motor positioning drives. However, "index," as well as "seek complete," and "sector/byte clock/address mark found" are available on either the daisy chained or radial cable, making overlapped seeks possible.

On the other hand, serial mode takes advantage of more intelligent, higher performance devices such as disk drives that use closed-loop servo systems and voice coil actuators. When serial mode is implemented, the full benefit of ESDI becomes apparent. Voice coil actuator positioning systems provide access times of around 30 ms and allow track densities in excess of 800 tracks/in. This allows systems that run multi-user/multitasking operating systems to obtain the capacity and performance previously attainable only with large drives (8 and 14 in.), with 5¼-in. form factor drives. In addition, attaching serial interface disk and tape drives to ESDI controllers allows systems to be self-configuring. This simplifies the task of writing software for a wide variety of devices.

Since the purpose of a disk or tape drive is to store or retrieve data, a key feature of any interface is how that data is transferred between drive and controller. In the past, most 5¼-in. drives have transferred data in the same manner in which it was encoded on the drive, ie, in a modified frequency modulation (MFM) format. The ESDI, however, specifies that data separation is done on the drive, and that data transferred between the drive and controller are nonreturn to zero (NRZ). Since NRZ data is not self-clocking, separate read and write clocks must be provided (Fig 2). Although the requirement that each drive have its own data separator adds to the drive cost, the advantages that it provides over interfaces transferring MFM data seem to justify the added cost. These advantages include more reliable data transfer between the controller and drive, the ability to interface disk drives that transfer data at different rates, and the flexibility to incorporate advanced encoding schemes on the drive.

The data interface is more reliable because NRZ encoded data is transmitted. Instead of transmitting a single signal composed of both clock and data pulses down a cable, the drive or controller provides a separate 50-percent duty cycle clock. This allows data bits to be a full bit cell in length, and makes NRZ data much less susceptible to noise and jitter than MFM.

The variable data transfer rate stems from the fact that all ESDI drives provide a reference clock signal that determines the drive's transfer rate. On the other hand, with the ST412 HP interface, the controller sets the data rate. Thus, if the manufacturer wishes to take advantage of the standard interface, drive performance may be impaired. Furthermore, the encoding scheme used on the drive becomes almost transparent to the interface because data is transferred in NRZ format.

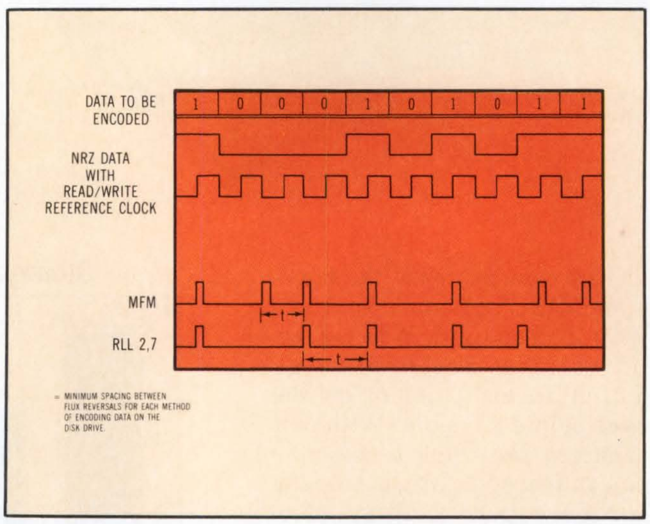

Fig 2 The ESDI calls for data separation to be done on the drive, with nonreturn to zero (NRZ) data transferred between drive and controller. As NRZ data transfer requires read/write reference clocks, it provides a reliable controller to drive interface. Modified frequency modulation (MFM) data encoding or an advanced encoding scheme such as a 2,7 run-length-limited (RLL) code may still be used on the drive. The diagram shows that advanced encoding schemes offer significantly increased bit density.

MSB LSB

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>P</td></tr>
<tr><td colspan="4">COMMAND FUNCTION</td><td colspan="4">COMMAND MODIFIER</td><td colspan="8">ALL ZEROS</td><td>P</td></tr>
<tr><td colspan="4">COMMAND FUNCTION</td><td colspan="12">COMMAND PARAMETER</td><td>P</td></tr>
</table>

BIT P: PARITY (ODD)

COMMAND FUNCTION BIT 15	14	13	12	DISK DRIVE COMMAND FUNCTION DEFINITION	TAPE DRIVE COMMAND FUNCTION DEFINITION
0	0	0	0	SEEK	LOCATE BLOCK (OPTION)
0	0	0	1	RECALIBRATE	LOAD TAPE (OPTION)
0	0	1	0	REQUEST STATUS	REQUEST STATUS
0	0	1	1	REQUEST CONFIGURATION	REQUEST CONFIGURATION
0	1	0	0	SELECT HEAD GROUP (OPTIONAL)	SELECT TRACK
0	1	0	1	CONTROL	DECK CONTROL
0	1	1	0	DATA STROBE OFFSET	READ
0	1	1	1	TRACK OFFSET	UNLOAD TAPE (OPTION)
1	0	0	0	INITIATE DIAGNOSTICS (OPTIONAL)	INITIATE DIAGNOSTICS
1	0	0	1	SET BYTES PER SECTOR (OPTIONAL)	SPACE FORWARD/REVERSE
1	0	1	0	RESERVED	FAST POSITION EOT/BOT
1	0	1	1	RESERVED	WRITE
1	1	0	0	RESERVED	ERASE FIXED LENGTH
1	1	0	1	RESERVED	RESERVED
1	1	1	0	RESERVED	RESERVED
1	1	1	1	RESERVED	RESERVED

Fig 3 The serial mode of implementation for disk and tape devices takes advantage of added drive intelligence. Communications with the drive are carried out through 16-bit commands and responses, followed by a parity bit. Thus, the system can be self-configuring, simplifying the writing of software to handle a wide variety of devices.

This transparency allows data to be encoded using high density methods such as MFM or a variety of run-length limited (RLL) codes. The trade-off in selecting a code is one of circuit complexity versus bit density. For example, 2,7 RLL code increases bit density by 50 percent over MFM, while maintaining the same number of FRPI. However, the implementation of 2,7 RLL code requires much more circuitry than does MFM.

Perhaps of greatest significance to system designers is the fact that the driver provides the data separation function internally. Thus, the drive manufacturer can control the drive error rate more precisely, independent of the controller used. This is not the case with MFM data transfer, as different controller manufacturers use different data separators. Thus, disk drives with NRZ data transfer can provide higher data integrity with a wide variety of controller designs.

The performance benefits attained by putting the data separator on the drive more than offset the cost in terms of system efficiency and reliability. Furthermore, NRZ data transfer is not new to the world of disk drives. For years, 8- and 14-in. drives have used NRZ interfaces. MFM data transfers were used for 5¼-in. Winchesters because they evolved in a floppy-disk drive environment. With ESDI and NRZ data transfers, 5¼-in. disk drives can incorporate the performance and data reliability traditionally provided by larger drives.

Surveying step and serial modes

The ESDI step mode is intended for drives with stepper motor positioning. This mode is very similar to the ST506/412 interface; however, the data transferred is NRZ. A maximum of three drives is supported in this mode. Each drive can have up to 16 heads (if fixed media), or eight heads (if removable media). In a removable cartridge drive, the most significant head select line is used as an optional "remove cartridge" line, allowing the spindle motor to be stopped for cartridge removal.

In step mode, head positioning is achieved with the controller setting the direction line, and issuing a step pulse for every track to be moved. Step pulses can be sent at up to a 4-μs rate. As with the ST506/412 interface, the system is responsible for keeping track of the drives' parameters (ie, number of heads and cylinders). Usually, the controller must be initialized with this information before the drive can be used for normal operations.

Serial mode, which supports up to seven devices, assumes a more intelligent drive, and is intended to take advantage of the high performance capabilities of disk drives with voice coil actuators and closed-loop servo systems. Communications between the controller and the drive are carried out using a series of 16-bit commands and responses with a parity bit (Fig 3). With a simple request/acknowledge handshake protocol for each bit, commands and responses are sent in a serial fashion. The controller initiates all transfers. A drive can request that the controller read the drive's status by asserting the attention line on the interface.

Performance benefits attained by putting the data separator on the drive offset cost through efficiency and reliability.

Commands consist of a 4-bit command function code, followed by either 12 parameter bits, or 4 bits of command modifier and 8 bits set to 0. The command function code determines the type of command, such as seek or request configuration. The parameter bits are used to pass information to the drive, such as a cylinder address in a seek command. The modifier bits are used to obtain up to 16 variations of each command. Through various modifier bits, the request configuration command can determine all of a drive's attributes. With the modifier bits set to 0, the command returns the general configuration, while 1 in the modifier bits, for example, would return the number of fixed cylinders in a drive.

BIT POSITION	FUNCTION
15	RESERVED
14	REMOVABLE MEDIA NOT PRESENT
13	WRITE PROTECTED, REMOVABLE MEDIA
12	WRITE PROTECTED, FIXED MEDIA
11	RESERVED
10	RESERVED
9	RESERVED
8	POWER ON RESET CONDITIONS EXIST
7	COMMAND DATA PARITY FAULT
6	INTERFACE FAULT
5	INVALID OR UNIMPLEMENTED COMMAND FAULT
4	SEEK FAULT
3	WRITE GATE WITH TRACK OFFSET FAULT
2	VENDOR UNIQUE STATUS AVAILABLE
1	*WRITE FAULT
0	REMOVABLE MEDIA CHANGED (REMOVABLE MEDIA HAS BEEN CHANGED SINCE LAST STATUS REQ)

(a)

COMMAND MODIFIER BITS				FUNCTION
11	10	9	8	
0	0	0	0	GENERAL CONFIGURATION OF DRIVE AND FORMAT
0	0	0	1	NUMBER OF CYLINDERS, FIXED
0	0	1	0	NUMBER OF CYLINDERS, REMOVABLE
0	0	1	1	NUMBER OF HEADS
0	1	0	0	MINIMUM UNFORMATTED BYTES PER TRACK
0	1	0	1	UNFORMATTED BYTES PER SECTOR (HARD SECTOR ONLY)
0	1	1	0	SECTORS PER TRACK (HARD SECTOR ONLY)
0	1	1	1	MINIMUM BYTES IN ISG FIELD*
1	0	0	0	MINIMUM BYTES PER PLO SYNC FIELD
1	0	0	1	NUMBER OF WORDS OF VENDOR UNIQUE STATUS AVAILABLE

(b)

Fig 4 Standard status responses to the controller's status request command include error reports and state of the drive information (a). Responses to the request configuration command (b) return information as to drive type, sectoring (hard or soft), encoding method, and capacity.

In serial mode, the controller passes the cylinder address to the drive as part of the seek command (direct cylinder addressing). The drive calculates the direction and magnitude of the seek to be performed. Other commands allow the controller to request the drive's status, start or stop the spindle motor, and perform other functions. Optional data strobe and track offset commands can be used to attempt data recovery in case of error.

Responses to the request status command allow the controller to gain information as to the current drive state [Fig 4(a)]. This includes error reporting and some state information such as drive write protected or removable media not present in a cartridge drive. Values other than 0 for the modifier bits in the request status command allow the controller to read status words unique to the drive vendor, such as diagnostic results.

Responses to the request configuration command return static attributes of the drive including 16 bits of general configuration. The general configuration gives the controller such information as the drive's type, whether it is hard- or soft-sectored, and whether or not it uses MFM encoding [Fig 4(b)]. This last point is very important in the design of controllers that use error correction circuitry to detect and correct errors. A media defect that will cause a single-bit error in MFM can cause an error up to 5 bits long in 2,7 code system. (This propagation in error occurs since 2,7 code encodes data up to 4 bits at a time.) Therefore, a controller that attempts correction in a system using 2,7 code should be able to handle a burst error of at least 5 bits.

The other responses to request configuration return physical and format information necessary for drive use. The result is that the system need not keep track of the device's parameters in order to use

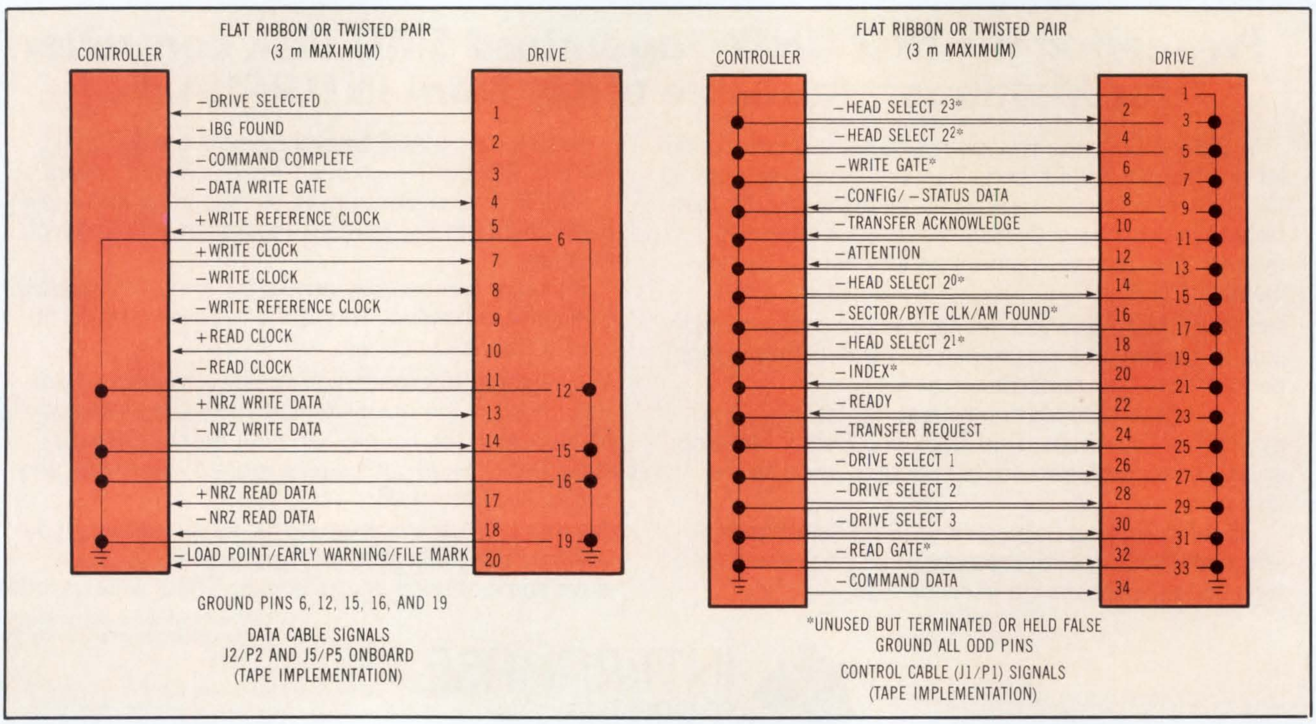

Fig 5 In ESDI tape implementation, pin assignments differ slightly from those used for disk implementations. Only serial mode is supported, and a write reference clock has been added to allow read after write operation.

it. This greatly simplifies the task of using different storage devices in a system.

Tape drive implementation

Tape drive ESDI implementation differs slightly from that for disks. Only serial mode is supported for tape drives, and an added write reference clock signal allows read after write operation (Fig 5). In this case, the write reference clock from the drive controls the NRZ data transfer rate during write operations.

Implementating disk drives and tape drives on the same controller reduces system overhead and controller count.

The ESDI supports two types of tape devices: streaming and start/stop tape drives. As with disk drives, the controller can obtain the tape drives' operating characteristics from the request configuration command. An optional locate block command adds considerable flexibility. When this command is issued, the controller may deselect the drive and monitor the drive's command complete signal on the data cable while the drive searches for the specified block. This allows the controller to tend to other tasks while the tape drive locates the correct block. The space forward/reverse command allows single block movement; it can also be used to search for file marks if supported by the drive. Although the streaming tape capabilities provided by the ESDI are similar to those of the QIC-02 interface, the ESDI goes further by allowing devices to be streaming, start/stop, or to have the capability of both modes.

Thus, the EDSI offers system designers a number of benefits. Implementation of both disk drives and tape drives on the same controller reduces system overhead and controller count. Because controllers have standard host interfaces, they can be easily incorporated into systems. Maximum reliability and efficiency from all devices are attained by virtue of NRZ data transfer. Self-descriptive drives provided by the interface's serial implementation allow systems to be self-configuring. In these ways, the interface provides the tools that designers will need to meet the demands of oncoming high performance storage devices.

Please rate the value of this article to you by circling the appropriate number in the "Editorial Score Box" on the Inquiry Card.

High 719 *Average 720* *Low 721*