Hard disc interfacing

2/6/89
Ingenious
Elec Dec. 88

Falling prices and easy-to-use controllers bring hard-disc drives into the realms of add-on periperals.

J.H. ADAMS

Hard or fixed disc drives, usually Winchester types, are appearing in more and more high-performance microcomputers. As a result, the cost of drive units has fallen until now, byte for byte, they are at least ten times cheaper than floppy-disc drives.

Despite this cost fall, hard disc drives are not significantly affecting the general add-on peripheral field as they are perceived as being difficult and expensive to interface to general-purpose microcomputers in hardware and software terms.

Certainly the i.cs involved are almost as expensive as the drive unit and the drive sends out data so fast that it would overwhelm most microcomputers. But the same factor that has brought down the cost of drives – the enormous IBM PC and clone market – has led the manufacturer of the highly priced i.cs to supply ready-built drive controllers capable of working with high or low-performance systems at a reasonable price.

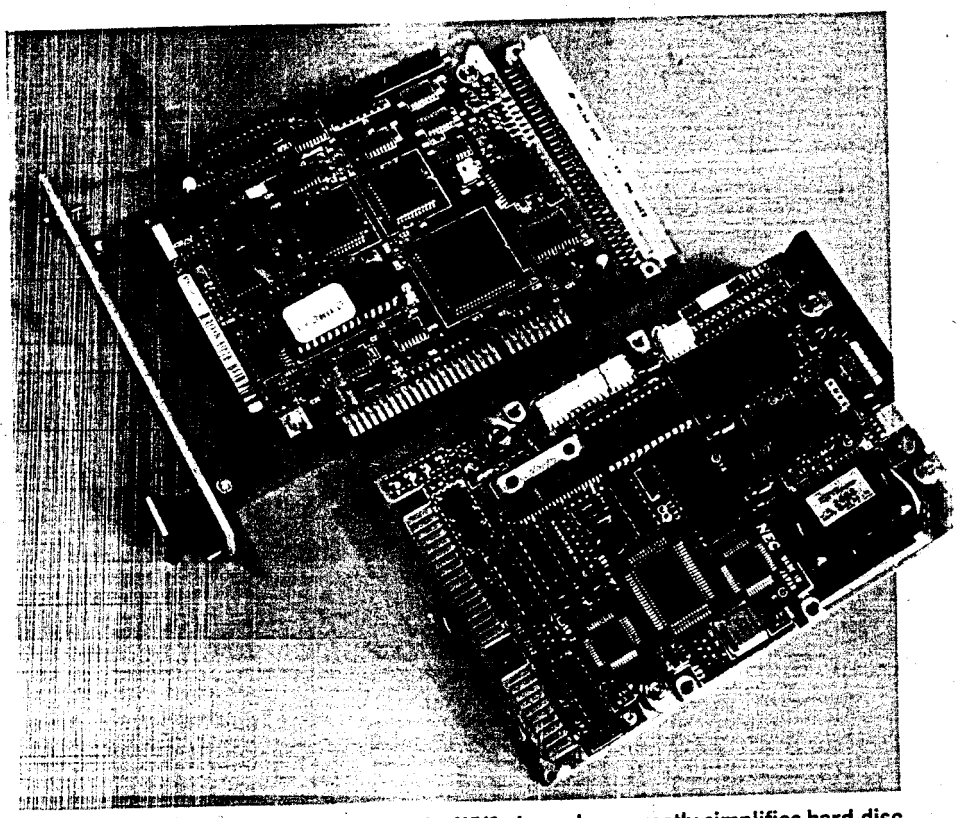
With the aid of a working example in the form of a hard-disc interface to the SC84 microcomputer, this article describes how such a ready built controller is applied.

When fitting a hard-disc drive to a computer, there are two distinct interfaces. The interface between the controller and drive unit is usually well defined. In the design example it is an ST506 type, which is similar to a floppy-disc drive interface. Open-collector active-low drive control signals are used and each alternate line on the cable is grounded. The difference is a separate 20-way signal cable carrying data to and from the drive using RS422 balanced lines. Lines conforming to RS422 can carry data at up to 10 Mbaud over 1200m and so are well suited to the 5Mbaud requirement of standard hard-disc drives.

The second interface, between the controller and host computer, should be made to suit the host's internal bus. In the design example, the controller used is intended for the IBM bus but the structure of this interface and options on the controller interface make connection to a different bus quite easy.

For the design example, the controller chosen is a Western Digital WD1002S-WX1. Through extensive use of surface-mounting technology, this half IBM-size board measures only 4.9 by 3.8in and is small enough to sit on a Eurocard. A slightly cheaper alternative using conventional mounting technology, the WD1002S-WX2, measures 8 by 3.8in.

The controller board is a complete microcomputer with its own instruction set. These instructions cover drive-parameter initialization, resetting, seeking specific cylinder positions, reading/writing single or multiple sectors, formatting single or multiple sectors and self-testing.



Using a proprietary controller card like the WX1 shown here greatly simplifies hard-disc interfacing and reduces costs. Note that the interface boards are not wired together.

Besides features such as programmable write precompensation and reduced write current to improve reliability, the controller includes an error-correction code (e.c.c.) function and options for d.m.a. and interrupt-on-completion functions.

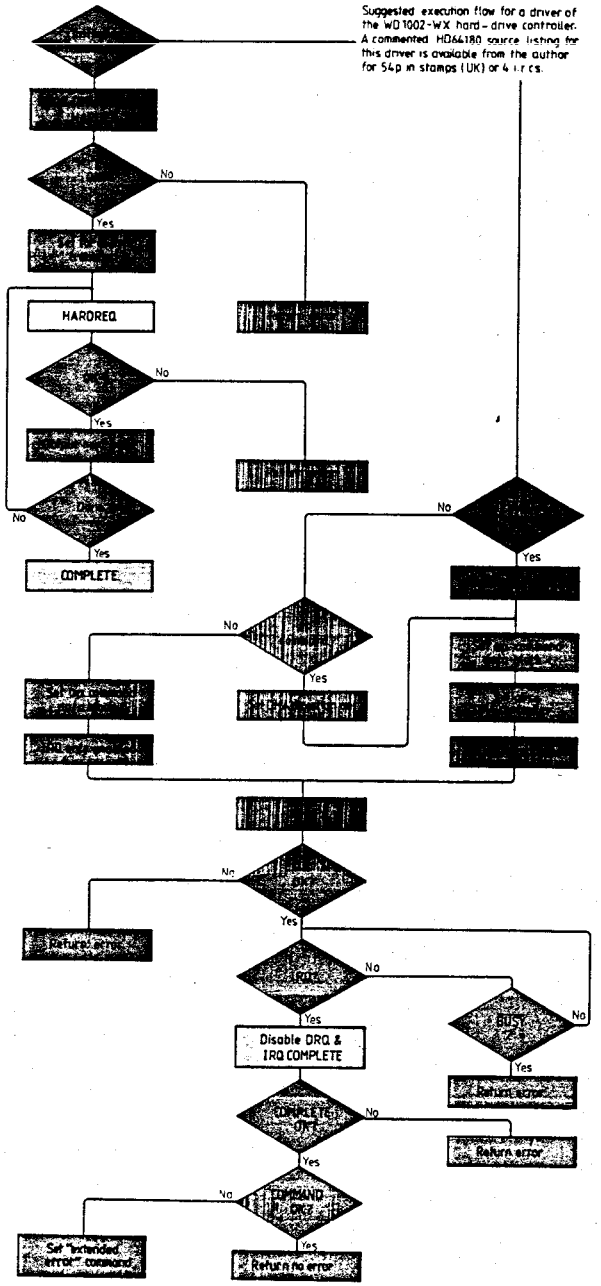
There is a fully-decoded 8K-byte eprom on the board and an eight-bit input port. These features can form part of the hard-disc system, as they do in the IBM PC, or alternatively become general system facilities.

The host interface is designed for a non-multiplexed 8-bit data and 20-bit address bus with 8088 control signals. How such signals are derived from HD64180 Z80-like signals using simple gates and inverters is shown in the circuit diagram. Most of the circuit is devoted to matching two different types of d.m.a. system.

In direct memory access, d.m.a., data is transferred from one part of a system to another as a background task to the main processing. This task is performed by a dedicated i.c. or, as with the HD64180, by a controller integrated into the processor. Transfers to be discussed here are between memory and an i/o device.

The d.m.a. controller is initialized with data including source memory address or i/o port number, destination memory address or i/o port number, transfer count and a flag for increment or decrement memory address. Once initialized, on receipt of a hardware request signal from a device, the d.m.a. controller gains control of the system bus, makes the transfer and then hands control back to the main processor. This process may repeat until access is complete or may be controlled by the d.m.a. rate requested by the device. In practice there are two distinct methods used for the transfer.

The 8237 d.m.a. controller, as fitted to the IBM-PC, has a series of request (DRQ) inputs and matching acknowledge (DACK) outputs. On receipt of a DRQ signal the controller sets the corresponding DACK line active. This line goes directly to the requesting device and is used as a cue in conjuction with IOR or IOW to gate data from or to the device, independent of chip select or addressing logic. This means that the d.m.a. controller can be sending the memory address for the other



end of the transfer and by activating the MEMWR or MEMRD signal data is transferred directly between the two.

During this transfer another signal, AEN, is produced which disables all devices it drives so that they (i/o devices) don't interpret the memory address and i/o control strobe as an invitation to perform a conventional i/o operation during the d.m.a. transfer. Increased speed is the advantage of this type of d.m.a. Disadvantages are the extra bus lines required and the dedication of DACK lines to individual devices, making expansion difficult.

The HD64180 d.m.a. controller responds to a d.m.a. request by performing two processor-like cycles to transfer the data. There is no intrinsic DACK signal; 'servicing' of the requesting device acts as an acknowledge. This interrupt-like service allows the system to be expanded almost indefinitely but also means that the d.m.a. process is slower. Having two – and more – d.m.a. systems complicates using peripheral devices in d.m.a. applications.

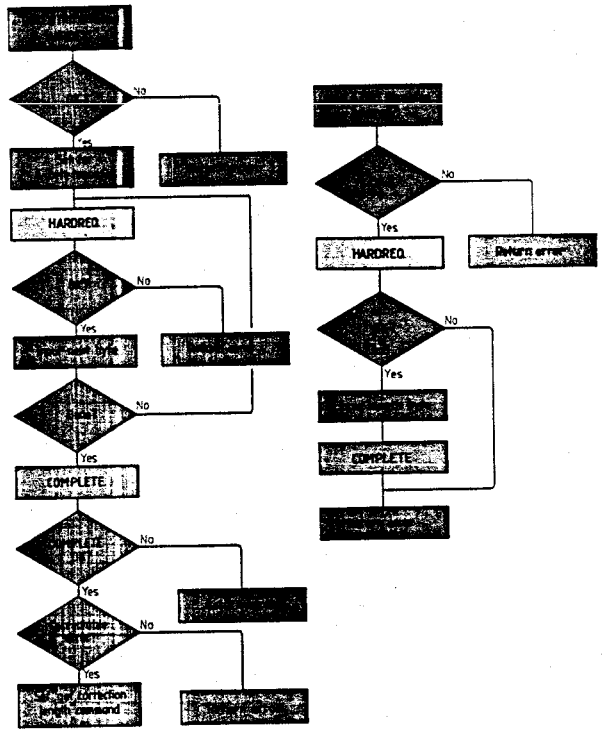
Probably the most dissimilar examples of d.m.a. processing are the 8237 and the HD64180 so the approach taken in the circuitry should cover other processors' requirements. By gating HD64180 $\overline{IOE}$, $\overline{WR}$ and $\overline{RD}$ signals together, an active-high signal is produced during any i/o operation. This combined signal is inverted using part of a transparent latch wired to be permanently transparent and then gated with the ST signal to produce an active-low signal signifying a d.m.a. i/o operation. When active, this signal freezes the states of the DRQ lines coming from the hard-disc controller.

After further gating with the "DMA I/O" signal, this latched signal forms an active-low signal signifying "d.m.a. i/o taking place in response to my d.m.a. request" which conforms to the 8237 $\overline{DACK}$ signal. Signal DRQ has to be latched because one of the responses of the peripheral board to a $\overline{DACK}$ signal is to turn off the DRQ. If DRQ is not latched, the DACK signal is removed immediately.

One awkward feature of the controller board is that its DRQ output floats when the controller d.m.a. is disabled. An open-collector gate drives the bus $\overline{DREQ}$ line and polarity of the DRQ signal combines with the naturally high input of t.t.l. gates to produce a permanently active $\overline{DREQ}$ signal under these conditions.

To counter this effect a pull-down resistor is used. The resistor's value must be low enough to pull the low-power Schottky t.t.l. input to a logical zero when DRQ is disabled yet high enough to allow the controller to pull the DRQ line high enough to produce a logical one without overloading its output. For this reason, only low-power Schottky t.t.l. must be used for this device.

The circuit is a general purpose IBM interface with extra DRQ and INT lines which are not used with the SC84 implementation of the hard disc interface but which may be of use in other applications. Signal AEN is provided but should not be used in SC84, the AEN input to the controller being wired low. No buffering of the address or data lines is provided as it is expected that the peripheral



board will provide standard buffering.

When using the unit in the SC84 the controller board should be mounted on an IBM inter-face card using nylon p.c.b. spacers and hard wired to it using short sections of ribbon and single cable.

To the host computer, the hard disc controller appears as an 8-Kbyte eprom at address C8000-C9FFF (48000-49FFF for SC84) in the memory map and as four consecutive i/o points at addresses 0320-3 or, by a link option, 0324-7. The first port is a bi-directional data port. Status of the controller is provided by the second port, which is read only. While read-only, this port may be written to in order to reset the microprocessor on the controller board. The third port is read-only and reads the external eight-bit port described earlier. This port can also be written to in order to prime the controller board for a command sequence. Interrupt and d.m.a. hardware functions are enabled or disabled through the fourth port, which is write only. When using a Z80 or HD64180-based host, load the BC register with the i/o port address and then use instructions IN g,(C) and OUT (C),g to effect the transfer between the port and general-purpose register g.

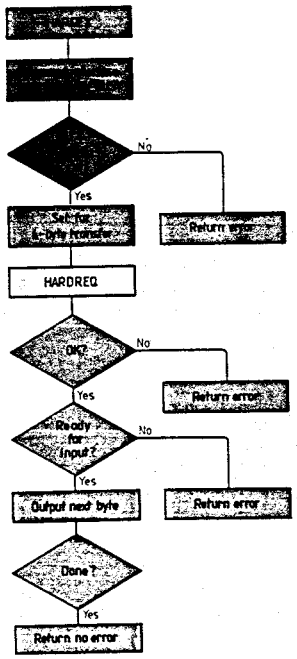
There are four phases in the controller execution sequence – selection, command, data (some commands only) and completion. The BUSY flag is set by the selection phase and remains active through to the completion phase. During the command phase six bytes of data are passed to the controller. Most commands have no data phase but if they do, data is transferred by either d.m.a. or programmed means.

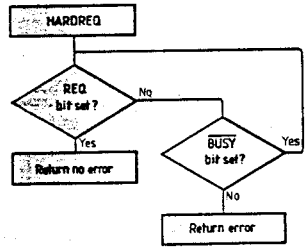
If the IRQ function is selected during the command phase, the IRQ bit in the status register or the external IRQ signal can be used to indicate that the controller is ready to pass to the completion phase. When the host reads the completion byte from the controller data port the command is completed. Any error in the command execution is signalled in this byte.

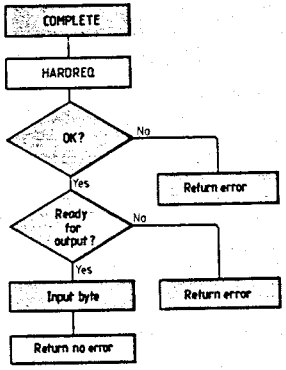
In the event of an error, the host may execute another command for reading the status of the last operation to obtain information about the error. Note that this command is completely separate from the one it is reporting on and is subject to its own errors. Software for the controller must allow for this.

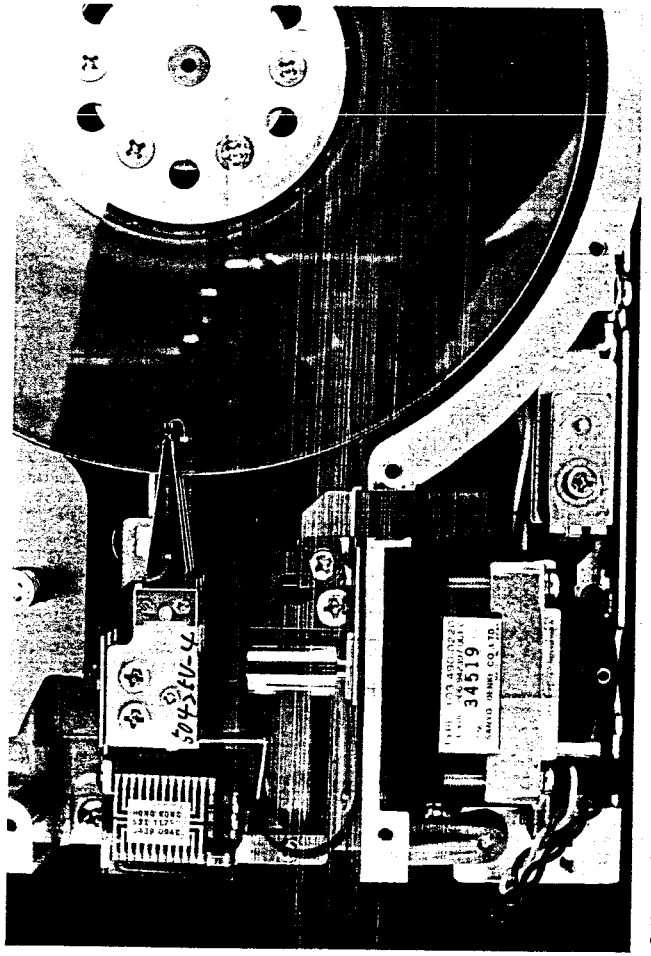
All non-d.m.a. data transfers between the host and port 0320 of the controller must be qualified by the bits BUSY, C/D, I/O and REQ in the status register. Driving software must contain a routine to check the status of the controller (REQ and BUSY set) before contemplating a transfer and then must further check C/D and I/O to make sure that the direction and type of transfer expected by the host is that expected by the controller.

Error data available when the command completion byte indicates an error has the same structure as the first four bytes of the









command block except that the first byte is the error code with bit seven set to indicate validity of the current and subsequent three bytes. By reading this code back into the buffer used for the command block, the block is automatically set to retry or continue since the details returned are those of the failed sector and/or track when an error occurs or in the event of no error those of the next track.

Sector count is the number of sectors to be processed during read, write and verify operations. Sector interleave is the skew allowed in between sequentially numbered sectors and applies during formatting operations.

Interleaving (or skewing) is necessary as the hard-disc controller reads data from a sector into an internal buffer memory and then offers the data to the host computer (or vice versa for writes). Hence if truly sequential sectors were used, unless the host was extremely fast, the start of the next sector to be processed would already have passed under the drive heads before the system was ready to process it. A delay of one disc revolution would be needed before that sector could actually be accessed. The optimum interleave is a function of the host processor's ability to service the internal buffer.

Unfortunately, Western Digital (and other controller manufacturers) do not provide direct d.m.a. bypassing the buffer, for systems such as the SC84 which are easily fast enough to process the disc data directly. More annoyingly the controller card cannot handle the interleave of two (i.e. read/write every other sector) even when the external d.m.a. process only takes half a sector period, and so the minimum practical interleave appears to three.

Thus it takes three disc revolutions rather than one to read a complete track. This is the price to be paid for using an off-the-shelf product. In practice this may not greatly affect overall system performance. While it adds 100ms to the time taken to read a complete cylinder (34-Kbytes), the average seek time of 85ms and the fact that files are rarely written as one contiguous block means that operations to access different parts of the file may become dominant.

All errors other than e.c.c. ones are affected by the general-retry bit. When the bit is set any error is immediately reported. When the bit is clear ten attempts are made to recover the error. If the requested sector is not found, recalibration to cylinder zero, a re-seek and then ten more attempts to execute the command are carried out.

When the e.c.c. retry bit is set the e.c.c. code is used to attempt to correct an error as soon as one is detected. When the bit is clear no attempt is made to correct the error until two consecutive error patterns are the same, providing extra protection against error.

Stepping rates vary between 3ms (bit pattern 000) and 10.5µs (bit pattern 111). Given a drive capable of "buffered seeks" as high a rate as the drive can accept should be used. Buffered seeks are ones where the number of stepping pulses are logged rather than executed by the drive, which then produces the optimum sequence of pulses to move and settle the head as quickly as possible. Increased speed produced by this "intelligent" stepping is quite impressive and so drives with buffered seek capability should be chosen.

A feature of this controller is that it doesn't check the seek-complete signal at the end of the SEEK command's execution but at the latest point possible in the following command. Thus the host computer can issue a seek, do something else such as set

Hard drive controller-board connections and options

A	Edge connector	B
n.c.	1	0V
D_7	2	RESET
D_6	2	+5
D_5	4	NC (IRQ_3)
D_4	5	NC
D_3	6	NC
D_2	7	NC
D_1	8	NC
D_0	9	+12
NC	10	0V
0V (AEN)	11	NC
A_{19}	12	MEMWR
A_{18}	13	IOW
A_{17}	14	IOR
A_{16}	15	DACK
A_{15}	16	DRQ
A_{14}	17	NC
A_{13}	18	NC
A_{12}	19	NC
A_{11}	20	NC
A_{10}	21	NC
A_9	22	NC
A_8	23	NC (IRQ_2)
A_7	24	NC
A_6	25	NC
A_5	26	NC
A_4	27	NC
A_3	28	NC
A_2	29	+5
A_1	30	NC
A_0	31	0V

Hard-drive option switches: SW_1 (D3126), TERMRES SW (D5126) – all switches on. SW_2 (D3126), DS (D5126) – switch 2 on, all others off. SW_3 (D3126) – all switches off.
J_1 is 34 way ribbon cable to drive J_2 is unused and J_3 is 20 way ribbon cable to drive.
Option links in place are W_3 link 1-2, W_4 link 2-3 nd W_5 link 2-3.

Controller board data structures

Port 320 bit significance (returned by routine COMPLETE)

Bit	Label	Function
1	E	Set if an error occured
5	D	Drive number

Port 321

Bit	Label	Function
0	REQ	Controller is ready for a data transfer when set.
1	I/O	Controller is to be read when bit set, written to when bit clear.
2	C/D	Data is expected when bit set, a command or status byte when bit clear.
3	BUSY	Controller is executing a command if bit set.
4	DRQ	D.m.a. request bit.
5	IRQ	Interrupt request bit.

Port 323 bit significance

Bit	Label	Function
0	DRQEN	Enables DRQ external signal and status bit.
1	IRQEN	Enables IRQ external signal and status bit.

Command block

Byte	Function
	Op-code for command to be executed
1	Bits 0 to 4, head number. Bit 5, drive number
2	Bits 0 to 4, sector number. Bits 5 to 7; m.s.b. of cylinder number
3	L.s.r. of cylinder number
4	Sector count or interleave
5	Bits 0 to 2, step rate. Bit 6, e.c.c. retry bit. Bit 7, general-retry bit.

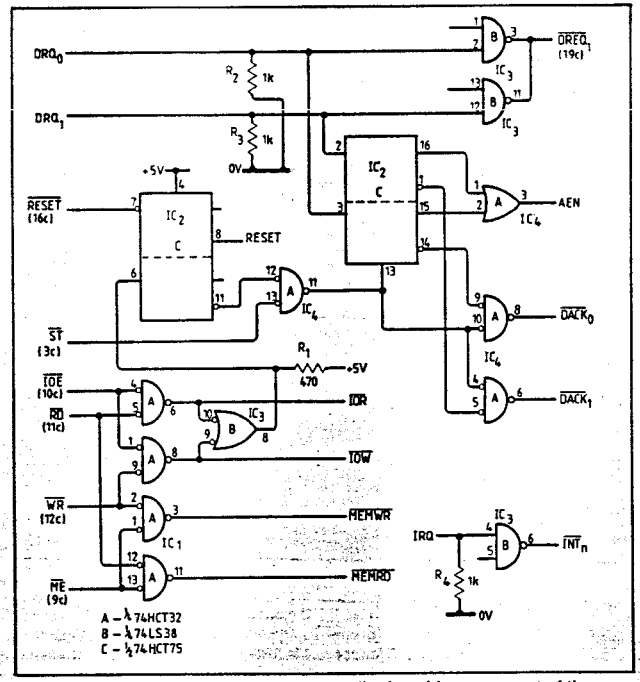
Initialisation data

Length	Function
Word	Number of cylinders (615)
Byte	Number of heads (4)
Word	Reduced write-current start cylinder (doesn't matter)
Word	Write precompensation start cylinder (128)
Byte	Maximum e.c.c. Burst Length (11)

Values in parenthesis are for the NEC D5126 drive

Error codes in hexadecimal

Code	Error
00	No error
02	SEEK COMPLETE has not become active (occurs 3.5s after executing SEEK)
03	WRITE FAULT active
04	DRDY not active when drive selected
06	TRACK ZERO has not become active during a RECALIBRATE instruction
08	Drive still seeking. This is no so much an error as a request to wait
11	Uncorrectable read error
12	Data address mark not found in sector
15	Sector identification requested not found
18	Corrected read error. This warns that the read was marginal. The e.c.c. burst length may be read to assess how bade the potential failure was, and on the strength of that, the sector re-written or the complete track reformatted.
19	An attempt to access a track marked as being bad
20	Invalid op-code
21	Invalid sector number
30	Error during sector-buffer test
31	Internal rom sum-check error diagnostic test of controller c.p.u.
32	Error during test of e.c.c. generator.



Interfacing Z80-like signals to a low-cost controller board is easy – most of the circuit is devoted to matching two different d.m.a. systems. The connector numbers are for SC84 using the 64180 processor.

These commands ignore command block parameters.

Op-code	Operation
0E	Read sector buffer
0F	Write sector buffer
E0	Execute sector buffer diagnostic test
E4	Execute controller diagnostic test

Only a drive needs to be specified in these commands.

Op-code	Operation
00	Test drive ready
01	Recalibrate to track zero (also needs general-retry bit)
03	Read status or last operation
0C	Initialise drive parameters
0D	Read e.c.c. burst length (only valid after error type 18)
E3	Execute drive diagnostic test (also needs general-retry bit and step rate)

These commands need all parameters. Byte four is the sector count.

Op-code	Operation
05	Verify sectors
08	Read sectors
0A	Write sectors, e.c.c. bit not used
E5	Read long
E6	Write long

Following commands need all parameters to be valid, although the sector and e.c.c. parameters are not used. Byte four is the interleaved factor.

Op-code	Operation
04	Format complete drive
06	Format track
07	Format bad track
0B	Seek specified track (byte 4 not used)

Format commands generate 17 sectors of 512 bytes each numbered 0 to 16 and interleaved as specified.

All diagnostic commands start with E_{16}. No error codes are generated by the executive controller-diagnostics test. The series of tests are repeatedly executed until an error occurs or until the controller is primed by writing to port 0322. When an error occurs and error code is put on the ST506 head-select lines. Under these conditions, the head number implies 1, error in WD1010A, 2, error in WD11C00-17 e.c.c. mechanism, 3, error in sector buffer, 4, error in WD1015 ram and 5, error in WD1015 rom.

Read and write-long operations allow the e.c.c. system to be tested. If an e.c.c. error is occurring it may be due to a disc fault, an e.c.c. checker fault or an e.c.c. generator fault. Read-long reads both the 512 data bytes and the four appended e.c.c. bytes. Used after a conventional write, this command can be used to see it is the e.c.c. pattern on the disc or the e.c.c. checker that it at fault.

Having performed the read-long test, the write-long operation writes both the 512 data bytes and four e.c.c. bytes back onto the disc, bypassing the e.c.c. generator. Now, reading the sector normally it can be establishes whether the disc or e.c.c. generator is at fault.

DRIVE AND CONTROLLER PRICES

Two drives suitable for this development are similar in performance and storage capacity, the only differences being in size and power consumption. The D5126 has the outline of a half-height 5.25in floppy-disc drive and requires 5V at 1A and 12V at 1.8A (2.5A peak start up current).

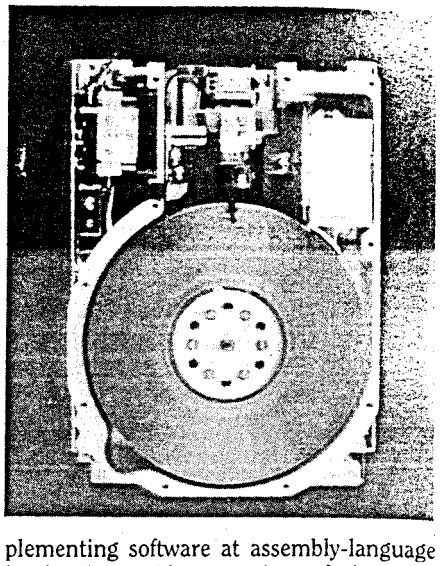
A smaller unit with the outline of a 3.5in floppy-disc drive is the D3126 requires about half the power of the larger drive. As the drives are almost the same price, the smaller one is the best choice, especially where power consumption and cooling need to be considered.

Pronto are offering these NEC drives and WD interface cards to E&WW readers at special prices. The D3126 20-Mbyte 3.5in drive is £341.73, and the DS5126 20-Mbyte drive is £333.26. Cables are included with the controllers, which are the WD1002S-WX2 at £89.63 and the WD1002-WX1 at £99.76.

All prices include v.a.t., UK postage and packing. Please send cheque with order to Barry Rennick at Pronto Electronic Systems, City Gate House, 399 Eastern Avenue, Gants Hill, Ilford, Essex IG2 6LR.

Boards for the SC84 to WX1 interface circuit shown in this article are available from Combe Martin Electronics at King Street, Combe Martin, North Devon EX34 0AD for £8 each inclusive (UK or overseas).

Components are available from John Adams. The complete SC84-to-WX1 interface set is £7.25 excluding v.a.t. to UK readers, £7.75 to readers within Europe and £8.25 to readers outside Europe. Small modifications are needed to the board and kit if the WC2 board is used. John is at 5 The Close, Radlett, Hertfordshire WD7 8HA.



another drive seeking and then return and perform a read or write command. All of this serves to speed up the system.

OPERATING SYSTEM REQUIREMENTS

Application of a hard disc drive requires the hardware to implement it, the driver to operate it and a system which can use it. To an operating system there are only two differences between hard and floppy discs. The most obvious is storage capacity, the other is the cost of the disc.

To expand on the latter point, if a floppy disc is not perfect you can replace it for one or two pounds. Perfection in a hard disc may literally cost hundreds of pounds as it is not just the disc but the whole drive that has to be replaced. For this reason manufacturers of hard drives supply units which are not 100% defect-free. In fact hard drives are graded by the manufacturer. A commercial compromise is struck and so the drive you obtain may have defects which the operating sysem must be able to tolerate. There is no hard and fast rule as to number of defects but a common practice is that cylinder zero of the drive must be perfect.

Imperfections can be handled in three ways. In some systems the controller reserves tracks which it substitutes for faulty ones much in the same way that in the new, larger, dynamic memories, rows of extra storage cells are available to connected in place of faulty ones. This technique reduces disc capacity by the number of reserved tracks and also slows disc access since the reserved tracks may be on a completely different part of the disc.

A technique which maximises good disc space is for the controller to split the disc into logical and physical storage units and then form a translation table between the two, the mapping of which skips over defective physical units. This allows the host computer to talk to the hard drive controller in terms of logical units wihout it needing to know of tracks, heads or sectors, but is makes the controller quite complex.

The third technique is similar to the second except that the mapping-out of defective areas on the disc is handled by the operating system. Being part of the operating system rather than a hardware subsystem enhances overall system flexibility.

Most disc-operating systems maintain files and a directory of those files. The directory contains an entry for each file detailing at the least the name of the file and the point on the disc at which the file begins. Thus the directory links something a human recognizes with something the computer can use to provide access to the named file. It is over this aspect that operating systems falter when presented with a hard disc as, simply due to the disc's enormous capacity the directory may contain hundreds – even thousands of files names.

Operating systems pre-dating large storage media, such as CP/M, were essentially single-directory systems. Later versions of CP/M and CP/M-compatible systems such as ZCPR have allowed extra directories to be placed on discs but the changes have subdivided the hard disc into many smaller units rather than expanding the attributes of the operating system.

Directories created under such systems are almost completely isolated from each other. Post-hard-disc systems, such as MS-DOS also support multiple directories but each directory is part of a hierarchical system. Thus at the highest level is a single directory often called the root, below which may be sub-directories called ACCOUNTS. CAR. ARTICLES. BILLS. etc. Within the BILLS environment for example there may be further directories called GAS. WATER. ELECTRIC and CLEANING and in each of those, directory files such as 1984. 1985 etc., or even other directories.

Structured directories of MS-DOS are more in line with the human approach to filing. You have an office within which are filing cabinets, within which are drawers, within which are files, within which are papers, on which are paragraphs etc., etc., but MS-DOS is not an eight bit system and so users of eight bit systems such as SC84 have not had access to such facilities.

To complete the application of a hard disc to the SC84, a new version of the operating system SCIDOS, version 3.0A, has been developed which offers the same concepts and native commands as MS-DOS but in a CP/M compatible system and in a package one tenth the size of MS-DOS! The difference in size illustrates the benefits of RISC (reduced instruction-set computing) as a means of minimizing code and of implementing software at assembly-language level. The resident portion of the DOS occupies less than 3-Kbytes of logical memory leaving the user with more than 59-Kbytes of working space.

New features introduced in this version of Scidos are a fully-structured, hierarchical directory system with user-defined search paths and file sizes only limited by disc capacity. File (and directory) allocation is on the daisy-chain system, resulting in typically 15% extra storage capability on a floppy disc. CP/M application software is incapable of handling any of the new features so to control them are in the enhanced CCP. This now processes BAT files and commands CHDIR. MKDIR. RMDIR. PATH and COPY in an MS-DOS-like way as well as recognizing command file-specifications including pathways.