Floppy-disc controller design must begin with the basics

Adding the random-access mass storage afforded by a floppy can turn your μC into a formidable system. Here's a primer on the subject.

Austin Lesea and **Rodnay Zaks,** Sybex Inc

The addition of a floppy-disc drive and accompanying operating software converts the most humble computer into a system that bigger machines must reckon with. Small wonder, then, that floppy-disc interfacing is one of today's hottest topics. This article will shed a little light on this most important subject.

Some operation essentials

A representative floppy-disc drive, the Shugart SA800, consists of read/write control and control electronics (on two pc boards), drive mechanism, read/write-head positioning mechanism and the head itself. The first of these facilities includes index and sector detection, read/write-head position-actuator drivers, read/write-head load-actuator drivers, write drivers, read amplifier and transition detectors, write-protect detector, drive-select circuits and motor control circuits.

The drive accesses and reads/writes data on a track by moving the read/write head over the disc's radius, usually by means of a stepping motor. The following sequence occurs:

- The control electronics activates Drive Select. Usually a controller oversees more than one unit; thus it must enable the Drive Select of the mechanism selected for access.
- The electronics sets Direction Select, which latches the head's direction of movement. As a result, the head moves either toward the disc's center or toward its periphery.
- The Write Gate goes inactive so that no writing occurs during head movement.
- The controller pulses the Step line until the head reaches the desired track. Each pulse produces one track's worth of movement in the direction previously set.

The unit accomplishes reading by activating Drive Select and disabling the Write Gate; writing occurs by activating Drive Select, activating the Write Gate and pulsing data in on the Write Data line.

What signals are needed?

In more detail, six essential signals allow communication with a disc drive **(Fig 1):**

- **Motor On**—This signal also turns the motor off. You allow 1 sec after activation when turning the motor on; conversely, deactivate the drive after 2 sec (or 10 revolutions) whenever no further commands are issued.
- **Direction Select**—As noted above, this signal sets the direction in which the head moves.
- **Step**—This signal moves the head one track toward the disc's center or away from it. Movement occurs on the pulse's trailing edge.
- **Write Gate**—When this line goes active, a write operation is enabled; when it goes inactive, a read occurs.
- **Track 00**—This signal indicates that the head has reached the outside track of the disc and will move no further even if additional step commands are issued.
- **Index/Sector**—This signal results whenever the drive detects either an index hole or (in the case of hard sectoring) a sector hole in the disc. The former marks the beginning of the disc's first sector. (A hard-sectored disc has an additional number of holes that mark the beginning of each sector; 11 or 17 pulses result for each revolution. For a soft-sectored disc, one pulse/rev (every 200 msec) occurs at the beginning of a track.)

A closer look at formatting

In this floppy-disc drive, both clock and data information are encoded in the same signal; each bit creates a clock pulse. If no further pulse occurs in the bit cell, the cell contains a ZERO; a

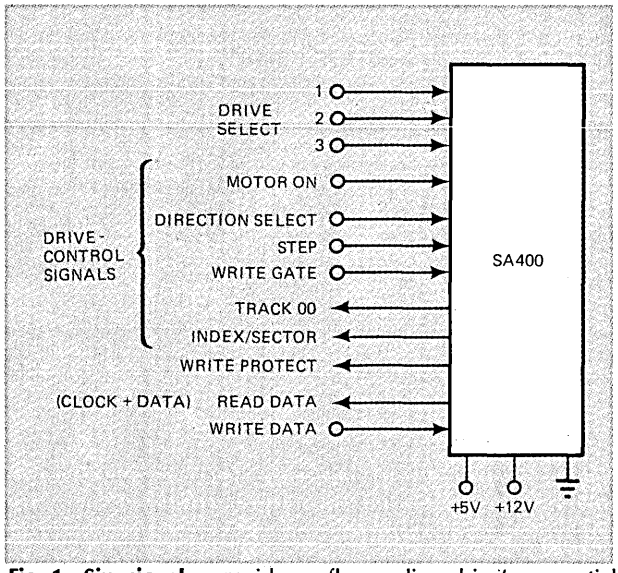


Fig 1—Six signals provide a floppy-disc drive's essential control functions. Additional signals select the proper drive in multidrive systems and control read/write and write-protect functions.

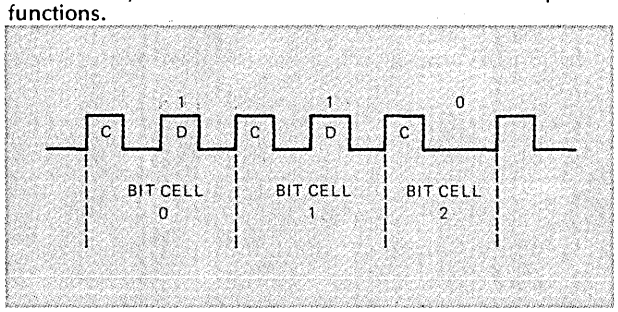


Fig 2—Recording method encodes both clock and data information on the same signal. Thus, each bit creates a clock pulse.

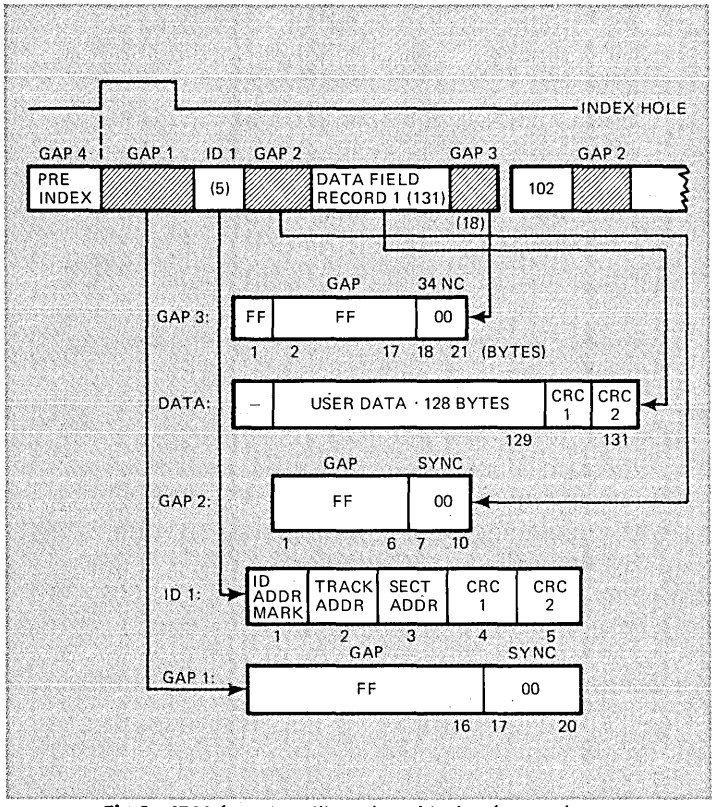


Fig 3—IBM format utilizes four kinds of record gaps.

data pulse in the middle of the cell represents a ONE **(Fig 2).**

In a soft-sectored disc, software divides each track into sectors. Each track starts with an index pulse, which corresponds with detection of the disc's index hole. A unique identifier precedes every record, and successive records are separated by gaps, which allow upgrading of information without erasure of preceding or following records. (Minor speed variations mean that when all or part of a record is rewritten, the end of the new record might extend beyond the previous record end.) In the IBM format **(Fig 3),** four kinds of gaps are used:

- Gap 4, used only once on a track, is the pre-index gap and appears at the end of the track just before the index-hole position.
- Gap 1, the index gap, appears at the beginning of every track. It consists of 20 bytes; the first 16 contain the hexadecimal pattern FF, while the last four signify 00. These latter four bytes provide synchronization for the data separator. The length of gap 1 can never vary; it is followed by ID 1, the identification of the first record. That identification uses five bytes: an ID address mark, the track address, the sector address and two CRC (cyclic redundancy check) check-sum bytes to verify the field's integrity. The track address and sector address verify that the drive has accessed the correct track and sector.
- Gap 2, the ID gap, separates each successive identification field from its data field. It uses 10 bytes; the first six contain the hexadecimal pattern F, while the last four are the usual synchronization bytes, 00. This gap's length can vary after file updating. Following gap 2, the first record (or data field) consists of 131 bytes, of which 128 are actually user data. The two usual CRC check-sum bytes follow.
- Gap 3 terminates the first record. Termed the data gap, it uses 21 bytes, the first 17 of which are set to FF and the last four of which contain 00. Each successive record or sector on the disc starts with ID and gap 2 and terminates with gap 3.

How to detect and correct errors

There are three types of errors: write, read and seek. In the first, data is written incorrectly on the disc. To detect such errors, you use a write-check procedure whereby the drive reads the data again during the next disc revolution. Normally, if this process shows an error, you simply write the data again. If after ten tries the data is still written incorrectly, you must consider the track or sector

damaged and unusable.

Read errors further divide into soft and hard varieties. The former result from transient conditions and are usually corrected by simple rereading or by moving the head back and forth once. If this operation doesn't correct the error, that error is a hard one and results in unrecoverable data.

Seek errors result when the head does not reach the correct track—a process you can verify by reading the ID field at the beginning of the track. Whenever such an error occurs, you must recalibrate the drive's track counter by moving the head back to track 00 and issuing a new Seek order.

Floppy-disc basics

Two types of floppy discs are available today: the full-sized 8-in. variety and the mini floppy, which measures 5.25 in. in diameter. Recording can be either single- or double-density, on one or both sides of the disc. Assuming single-density recording on one side, a full-sized disc provides

- 3.2M-bit total unformatted capacity
- 41.7k-bit unformatted capacity per track.

In IBM 3740 format, a full-sized 8-in. disc stores 2M bits (26.6k bits/track) and achieves a 250k-bps transfer rate.

Typical parameters for a full-sized drive like the Shugart SA800 are

- 8-msec track-to-track access
- 8-msec settling time
- 250-msec average access time
- 35-msec head-load time
- 360-RPM rotational speed
- 3200-bpi recording density (6400 bpi for double density)
- 48-track/in. track density (77 tracks total).

For a mini floppy, unformatted capacity equals 109.4k bytes/disc and 3125 bytes/track. Formatted capacity depends on whether the disc is hard- or soft-sectored:

Soft	Hard
• 80.6k bytes/disc	• 72.03k bytes/disc
• 2304 bytes/track	• 2058 bytes/track
• 128 bytes/sector	• 128 bytes/sector
• 18 sectors/track	• 16 sectors/track.

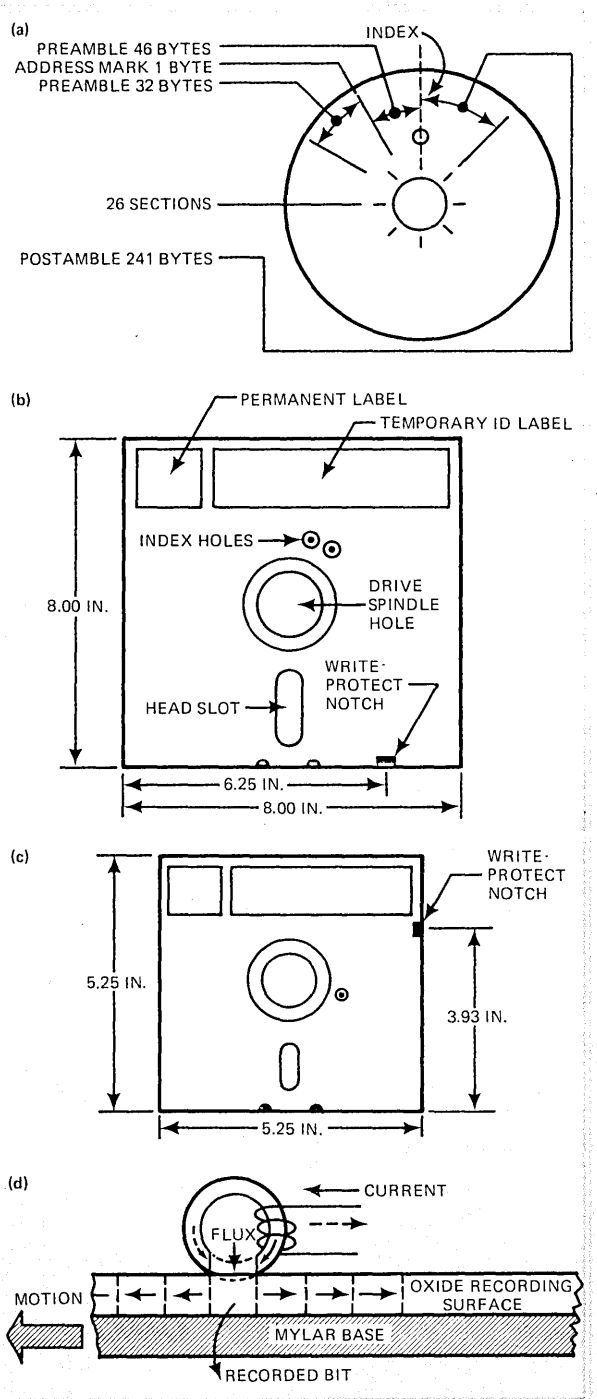
Other key mini-floppy parameters (quoted here for the Shugart SA400) include

- 125k-bps transfer rate
- 40-msec track-to-track access
- 463-msec average access time
- 10-msec settling time
- 75-msec head-loading time
- 300-RPM rotational speed
- 2581-bpi recording density (single density)
- 48-track/in. track density (35 tracks total; some units from other manufacturers record 40 or 77 tracks on a side).

The SA400 records data using an FM encoding technique. Reliability data for the drive includes:

- 3×10^6 passes/track life rating
- 30-min MTTR
- 8000-POM MTBF
- 10^{-8} soft-error rate
- 10^{-11} hard-error rate
- 10^{-6} seek-error rate.

Power consumption for the SA440 equals 15W continuous, 7.5W standby. The drive requires 12 and 5V dc supplies.



Without its protective envelope, a floppy disc **(a)** consists of a round platter of plastic coated with magnetic oxide and formatted in various ways. Two sizes are currently available: the full-sized disc **(b)** and the mini form **(c).** Part **(d)** shows a cross section of a disc to illustrate its composition and a simplified explanation of bit recording.

Error detection for any data written on a disc occurs through the check-sum method, utilizing cyclic redundancy checking. As noted previously, each field is terminated with two CRC bytes, which are the remainder when the system divides the data bits by a generator polynomial G(X). During a read operation, the system again divides the complete sequence of data bits plus appended CRC bytes by G(X); if no error (or an undetectable one) has occurred, this division now yields a zero remainder.

You can implement cyclic redundancy checking either in hardware **(Fig 4)** or software. In **Fig 4,** a shift register with feedback generates $G(X)=X^{16}+X^{15}+X^2+1$; the exclusive-OR feedback accomplishes the required division during successive shifts through the register's flip flops. Single-chip floppy-disc controllers can also accomplish CRC generation and checking.

Implementing a real-world controller

Now consider a specific floppy-disc controller, the Shugart SA4400, which oversees operations of the firm's SA400 Minifloppy drive. Shugart has implemented this board with the SMS/Signetics 300 bipolar controller chip; the board can control one, two or three SA400's.

Compatible with the IBM 3740 format, the SA4400 uses a modified gap structure (gap 4, the pre-index gap, is shorter than the 3740's). It provides a 128-byte data buffer and supplies eight control functions:

- INIT—Resets the controller in the disc drive
- SEEK—Steps ahead to the specified track
- READ—Reads a 128-byte sector
- READ ID—Reads the next sector identification
- WRITE—Writes a 128-byte sector with data AM (address mark)
- WRITE DDL—Same as WRITE, but with deleted data AM
- FORMAT—Writes address marks, gaps and data on an entire track in 3740 format
- STATUS—Obtains status for the drive.

The READ, WRITE and READ ID functions can send data between the host processor and the disc buffer or between the buffer and the disc itself.

The signals used by the SA4400 to communicate with the host μP system appear in **Fig 5.** The basic sequence of events is

- Seek track
- Find sector
- Shift/transfer desired number of sectors
- Check CRC.

Few commands are necessary to implement controller operation, and most controllers provide between six and ten such commands.

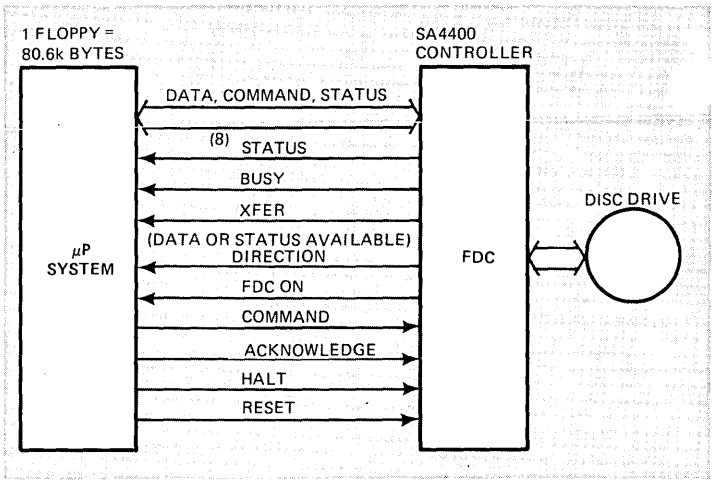


Fig 5—A typical mini-floppy controller, the Shugart SA4400, generates ten classes of signals to interface with a μP system.

A look at a one-chip controller

The Western Digital FD1771B floppy-disc controller/formatter (FDC) interfaces with most drives and is IBM 3740 compatible. It provides

- Automatic track seek with verification, a feature that all FDC's must offer

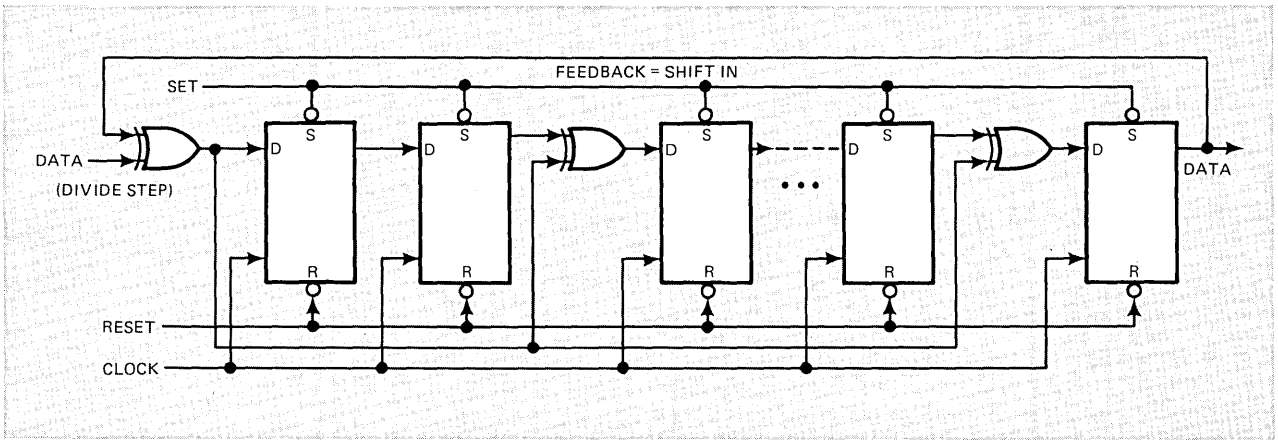


Fig 4—Flip flops and exclusive OR's generate a cyclic redundancy check. This particular circuit implements generation of the polynomial $G(X)=X^{16}+X^{15}+X^2+1$; the result of dividing a bit string by this polynomial is the CRC check sum.

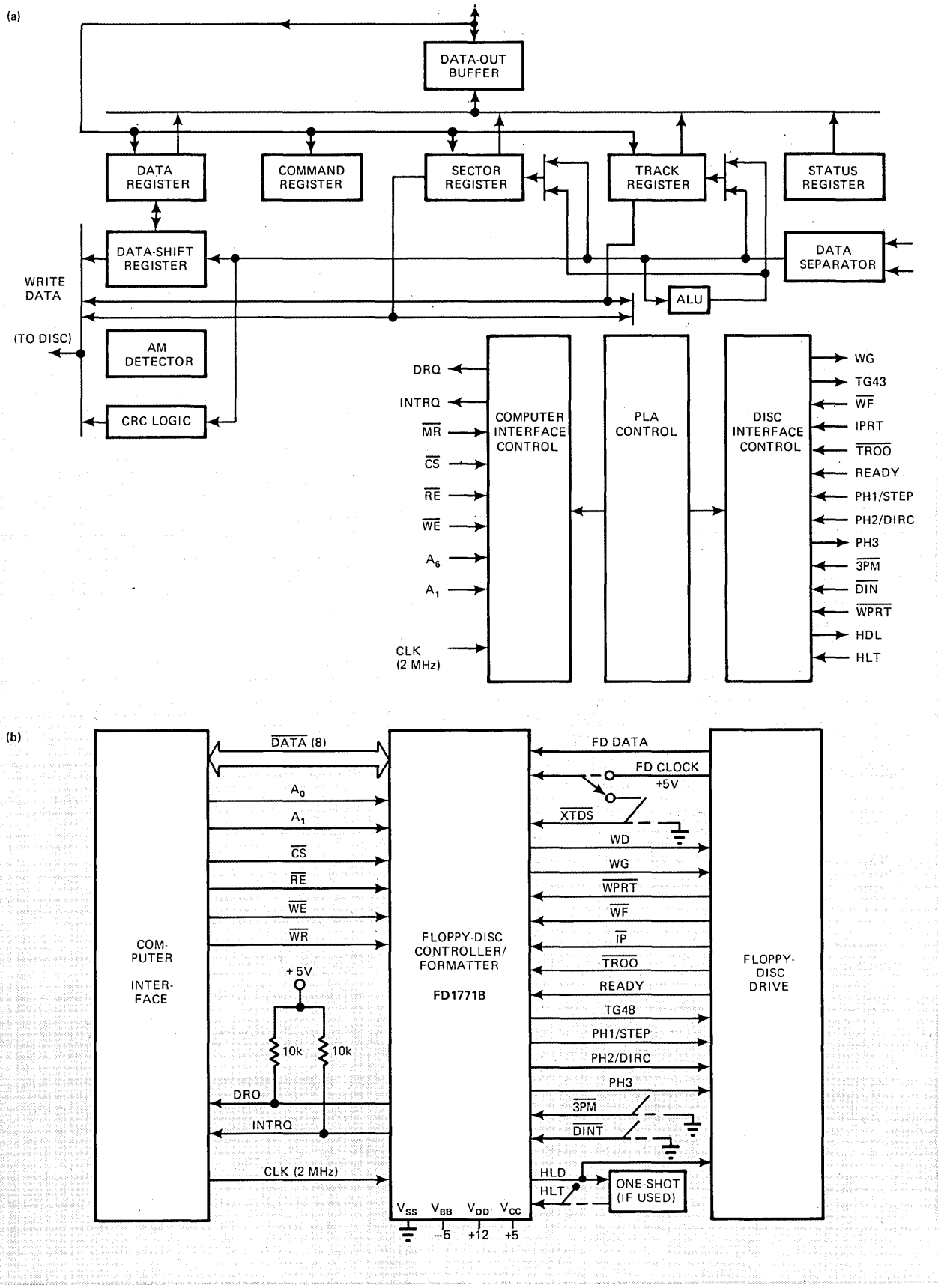


Fig 6—Internal architecture of the Western Digital FD1771B floppy-disc controller chip **(a)** consists of five functional blocks, six registers and two interfaces. A complete μP-to-floppy interface utilizing this chip appears in **(b).**

- Soft-sector format compatibility
- Read or write with single or multiple records, automatic sector search and entire-track capability
- Programmable controls for track-to-track stepping time, head-settling time, head-engage time, 3-phase or step-plus-direction motor control, and DMA or program transfers.

All of these features are standard for all FDC's; the differences between various chips usually lie in the number of drives that one chip can simultaneously control.

The FD1771B's internal architecture appears in **Fig 6a;** the chip consists of five functional circuits, six registers and two interfaces: one for the processor and one for the floppy disc.

In addition to the PIA control, the other four functional circuits are

- CRC logic—Generates check characters
- ALU—Implements arithmetic functions, including comparing characters for incrementing or decrementing contents
- Disc-interface control
- Computer-interface control.

The six internal registers and their functions are

- Data-shift register—Assembles eight bits from the floppy-disc data or serializes eight bits received from the μP data bus at the floppy-disc data line
- Data register—A simple holding register that contains a byte during read or write operations. It communicates with the data-out buffer and can receive data directly from the μP data bus.
- Command register—Holds the 8-bit command currently being executed. This register is loaded by the programmer and specifies the disc drive's operation mode.
- Sector register—Holds the address of the desired sector position
- Track register—Holds the track number of the current head position. This register is incremented when the head moves toward the disc's center (up to track 76 on a full-sized disc) and decremented otherwise.
- Status register—Holds the controller's status information.

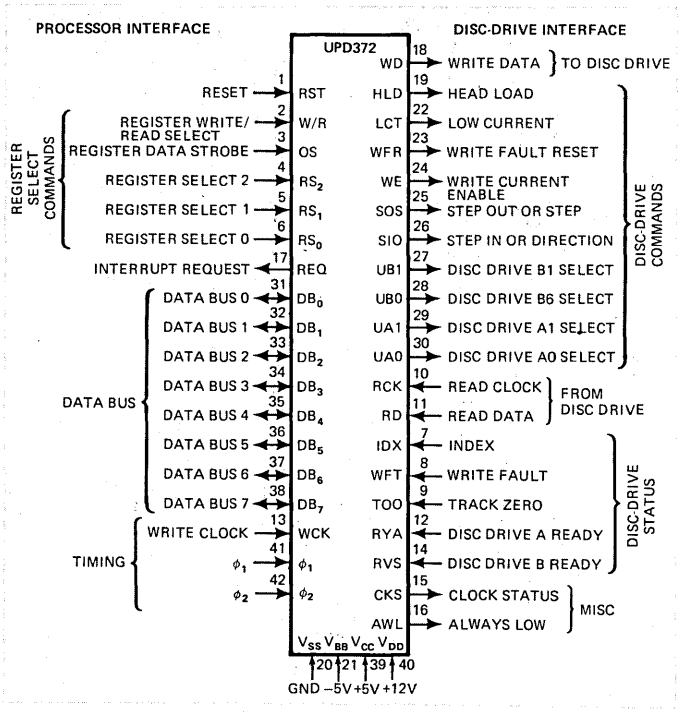


Fig 7—Another controller chip, the NEC UPD372, can handle up to four drives simultaneously.

The chip's two interfaces appear in **Fig 6b.** The FDC communicates with the μP via eight bidirectional data lines, labeled DAL. An input is specified when $\overline{CS}$ and $\overline{WE}$ (Write Enable) are active, and an action occurs when $\overline{CS}$ and $\overline{RE}$ (Read Enable) are active. The internal destination is specified by A_1-A_0; the data-request output (DRO) serves DMA. Interrupt Request (INTRT) is activated by various conditions.

The FDC-to-floppy-disc signals appear on the right side of **Fig 6b;** they provide head-positioning controls, write controls and data transfers. The clock is a 2 MHz square wave, internally divided by four to yield 500 kHz. It provides three programmable stepping rates, controlled by bit 1 and bit 0 of the command word. Head-settling time equals 10 msec.

A μP-to-FDC read operation takes five steps:

- Load the track register
- Generate the Seek command
- Wait for verification
- Transfer data to the μP under interrupt control
- Check for interrupt after the correct number of transfers.

A write operation, on the other hand, requires seven steps:

- Load the track register
- Generate the Seek command
- Wait for verification
- Generate the Write command
- Load the first data after receiving the data request
- Load the remaining data
- Check BUSY and CRC-error flag.

Some other FDC's

The NEC UPD372 is compatible with the IBM 3740 as well as the Shugart Minifloppy. It provides the usual facilities—CRC generation and programmable step pulse, track-stepping rate, sector size and data-transfer rate—and controls up to four drives. But read/write is limited to one drive at a time; the chip provides simultaneous track

seek for the others. The UPD372 also controls Calcomp 140, CDC BR 803, GSI 050 and 110, Innovex 210, PerSci 75, Pertec FD400 and Sycor 145 drives. A diagram of the chip's innards appears in **Fig 7.**

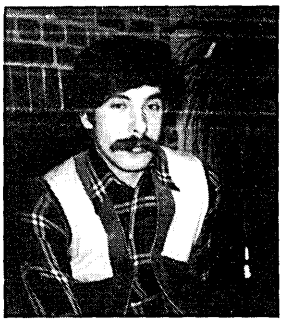
Another FDC, the Motorola 6843FDC, provides direct interface with the 6800 μP. It implements ten macro commands:

- Seek Track 0 (STZ)
- Seek (SEK)
- Single-sector Write (SSW)
- Single-sector Read (SSR)
- Read CRC (RCR)
- Multiple-sector Write (MSW)
- Multiple-sector Read (MSR)
- Free-format Write (FFW)
- Free-format Read (FFR).

This chip has two programmable delays for seek time and settling time, and it requires three channels of DMA. It uses about 3% of the μP's time; assuming a 256k-bps transfer rate, the maximum CPU load is 12.5%. □

Authors' biographies

Austin Lesea, senior engineer at Sybex Inc, Berkeley, CA, holds a BSEE and an MS in EE and computer science (1974 and 75, respectively) from the Univ of California at Berkeley, where he designed that school's microcomputer laboratory, among other projects. A member of IEEE and ISEC, he has a special interest in communication theory and spends his free time dabbling in photography, skiing and hang gliding.



Rodnay Zaks is president of Sybex and has published more than 20 books. A former professor and graduate (PhD in computer science, 1972) of the Univ of California at Berkeley, he is a member of the ACM, IEEE, SIGMICRO and SIGARCH and chairman of Euromicro, the European Association of Microprocessing.

Article Interest Quotient (Circle One)
***High* 479 *Medium* 480 *Low* 481**

Sloppy floppy

Dear Editor:

The article "Floppy-disc controller design must begin with the basics" (May 20, pg 129) contains the following factual errors or omissions:

1. The Shugart SA800 has a single pc board and not two (pg 129).
2. On pg 130, a soft-sectored disc format is described as the "IBM format." Instead, this is a format specifically designed for the Shugart Minifloppy and was, in fact, based on the IBM 3740, as it has the same recording technique (FM), the same sector size (128 bytes) and the same address marks (for sector ID and data/delete-data field). However, the two formats are *not* compatible, as the authors suggest on pg 134, primarily because of different number of sectors/track (18 in SA4400 vs 26 in IBM 3740) and different sector ID length (five bytes in one vs seven in the other).
3. The SA4400, in addi-

tion to the host-buffer and buffer-disc transfer modes for READ, WRITE and READ ID functions, has a direct host-disc mode for fast processors that significantly improves system throughput.

4. The CRC polynomial used in the IBM format, and similarly in the SA4400, is:
   $G(X)=X^{16}+X^{12}+X^5+1$
   and not:
   $G(X)=X^{16}+X^{15}+X^2+1$.
5. The correct sequence of events for data transfers between host and disc for read operations should be:
   a. Seek track
   b. Find sector
   c. Transfer sector
   d. Check CRC—if CRC error repeat steps b through d for the same sector up to 10 times.
   e. Repeat steps b through d until the whole file has been transferred.

Sincerely,
Chris Georgiou
Consultant
Sacramento, CA