PROFILE COMMUNICATION PROTOCOL

This document describes the communication protocol between the Profile hard disk drive and a host computer. Profile is connected to the host by a data bus, a CMD (command) input and a CRES (controller reset) input, a BSY (busy) output, and several other signal lines which are described in the Profile Controller Hardware External Reference Specification and will not be covered in this document.

When Profile is turned on, its processor waits 18 seconds for the disk to come up to speed. It then sequentially reads each block on the disk, using the read and write/verify/sparing routines described below, with a retry count of 105 and a sparing threshold of 53, but without the CMD - BSY handshakes. During this disk scan the hardware blinks the ready light about twice per second. The scan usually takes about 55 seconds, but will take more time if errors are encountered. After the scan is done Profile's ready light stays on without blinking, indicating that Profile is ready for use.

Profile supports three commands. They are: read, write, and write/verify. The host computer initiates all command sequences by raising the CMD line. Whenever Profile's Z8 processor is idle, it stays in a loop waiting for CMD to go high. After 1 1/2 seconds in this loop (except between the second and third handshakes of a write or write/verify operation) the Z8 will move the head to the innermost position, off the data area of the disk, and turn off the stepper motor.

The command bytes for each of the three commands are shown below.

		Block #		Retry Count	Sparing Threshold
READ	00	MS	LS		

		Block #	
WRITE	01	MS	LS

		Block #	
WRITE/ VERIFY	02	MS	LS

Profile interprets CMD high as a request from the host to send it a byte telling it (the host) what Profile expects to do next. When Profile is waiting for a command it sends an '01' in response to CMD high. Profile's other responses are shown in the table below.

PROFILE'S Next Action	PROFILE'S Response
get a command	01
read a block	02
receive write data	03
receive write/verify data	04
do actual write or write/verify on disk	06

Profile indicates that its response byte is on the data bus by raising BSY. It then waits (forever, if necessary, as there is no timeout) for CMD to go low. When that occurs, Profile reads the data bus. If the value read is a '55' (hex), Profile executes the next action, and lets the host know that it is done by lowering BSY. If the response from the host is not a '55', Profile sets the NAK received status bit, resets itself to the idle state and waits for CMD to go high again.

Profile uses only the number of bytes it needs for each command. Any extra bytes sent are ignored. Valid block numbers range from 000000 to 0025FF inclusive. A block number of FFFE will read or write Profile's RAM buffer, while a block number of FFFF will read Profile's spare table from the disk. The retry count parameter of the read command tells Profile how many times to reread a block if it gets a CRC or timeout error (zero is a valid number). If a CRC or timeout error occurs, Profile saves the data the first time it reads the block successfully, but rereads the block the full number of times specified in the retry count. If Profile is not able to read a block during any of the retries, it will attempt to read the block an additional 90 times or until the read is successful, whichever comes first. Each timeout error during these 90 retries counts as 9 retries, since that is how many times the disk rotates before a timeout occurs. If Profile is not able to successfully read the block after all these retries, it enters the block number in its bad block table, sets the appropriate error bits (described in detail later), sets up the bus so the host can read the result of its latest read attempt, and lowers BSY to indicate that the operation is finished. If the bad block table is already full (100 entries), Profile will set that error bit instead of entering the block number in the table. If, during the initial retries (those specified by the retry count), the number of errors is less than the number specified in the sparing threshold, Profile sets the four status

bytes to their appropriate values, sets up the bus for the host, and lowers BSY. However, if the number of errors is equal to or greater than the number specified in the sparing threshold, Profile goes through its write/verify/sparing routine. The w/v/s routine first attempts to write the data on the disk. If the attempt is unsuccessful because there was a seek settle error or because Profile was unable to read its spare table (two conditions which disallow all writes to the disk), Profile will set the operation unsuccessful status bit, set up the data bus for the host, and lower BSY. If the attempt is unsuccessful because of a timeout error, or if the read after write is bad, Profile will retry the whole write/verify routine one more time. If it still is not able to do it, Profile will retry the write/verify/spare routine using a spare sector on the disk. When a write/verify operation is successful, Profile will delete the block number from the bad block table, if it was there, and enter it in the spare table if appropriate. The only difference between a write/verify operation (which uses the write/verify/spare routine described above), and a write operation is that a write operation does not retry on a timeout error, and does not read the block after writing it (and will never spare a block). However, Profile will automatically change a write operation to a write/verify operation if the block being written is in the bad block table.

Profile's 9,728 usable blocks are divided into 152 cylinders of 4 surfaces, with 16 blocks (sectors) per track. The blocks are allocated to sectors sequentially, starting with track 0, head 0, sector 0,1,2, ... 15; track 0, head 1, sector 0,1.. ; .....; track 152, head 3, sector 1,2,...14,15. No blocks are originally assigned to cylinder 77, as it is reserved for the 32 spare sectors and the spare table (which includes some device specific information and the bad block table). Profile's interleave is 5 to 1 for reads, 21 to 1 for writes, and 37 to 1 for write verifies. The latter 2 obviously miss the physical interleave when used with the Apple III. In addition to the wait between successive writes, there is a 30ms wait before the first write after any cylinder change. Profile's rotation speed is 3600 RPM.

When the host requests a read or a write from Profile, Profile first translates the block number into the correct track, head and sector values. It then checks to see if the desired block is in the spare table, and sets the track, head, and sector accordingly if so. If the current block and the last block read or written have the same track and head, the Z8 exits the seek routine. If the track is the same but the head is different, the Z8 waits 750us and then exits the routine. Otherwise, the Z8 waits 24ms for the stepper to settle, then tries up to 64 times to read any 3 consecutive sectors on the disk (actually alternate sectors on the disk, since that is the best the hardware can do). If during these reads it determines that it is on the wrong head or track it will set the appropriate error bit and go back to the beginning of the seek routine. If the Z8 is not able to read 3 consecutive sectors because of a timeout (no header found in 26ms) or CRC error, it will retry the entire seek routine up to twice more after moving the stepper off track first to the innermost track and back, and if not successful, then to the outermost track and back. If it is still not able to read 3 consecutive sectors the Z8 will set the seek

settle error bit, which as mentioned disables all writes to the disk.

Following a read or a write the Z8 provides the host with 4 status bytes. They are placed in the buffer immediately preceding the data just read or written. The significance of the individual bits is as follows:

STATUS 1

7 = 1 if Profile received <> 55 to its last response
6 = 1 if write or write/verify was aborted because >532 bytes of data were sent or because Profile couldn't read its spare table
5 = 1 if host's data is no longer in RAM because Profile updated its spare table
4 = 1 if SEEK ERROR - unable in 3 tries to read 3 consecutive headers on a track
3 = 1 if CRC error (only set during actual read or verify of write/verify, not while trying to read headers after seeking)
2 = 1 if TIMEOUT ERROR (couldn't find header in 9 revolutions - not set while trying to read headers after seeking)
1 = N.C.
0 = 1 if operation unsuccessful

STATUS 2

7 = 1 if SEEK ERROR - unable in 1 try to read 3 consecutive headers on a track
6 = 1 if spared sector table overflow (> 32 sectors spared)
5 = N.C.
4 = 1 if bad block table overflow (> 100 bad blocks in table)
3 = 1 if Profile unable to read its status sector
2 = 1 if sparing occurred
1 = 1 if seek to wrong track occurred
0 = N.C.

STATUS 3

7 = 1 if Profile has been reset
6 = 1 if block number invalid
5 = 1 if block I.D. at end of sector mismatch *
4 = N.C.
3 = N.C.
2 = 1 if Profile was reset *
1 = 1 if Profile gave a bad response *
0 = 1 if parity error *

STATUS 4

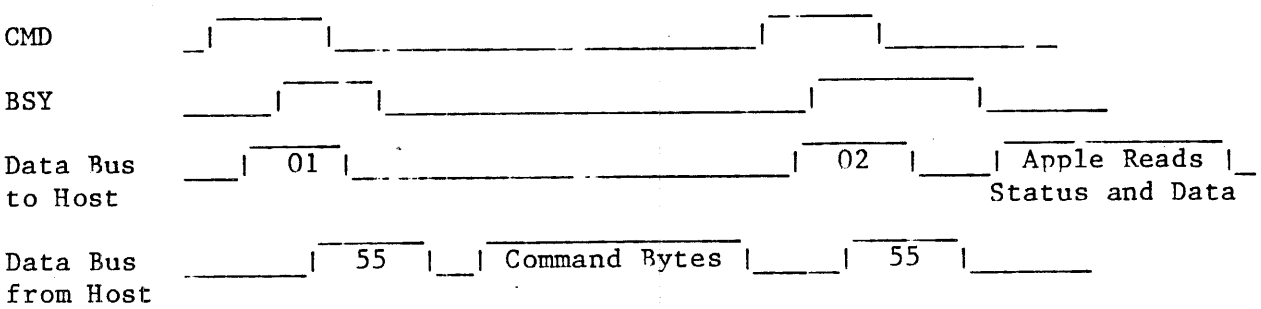
7 - 0 = the number of errors encountered when rereading a block after any read error

* These bits are set by the S.O.S. Profile driver.

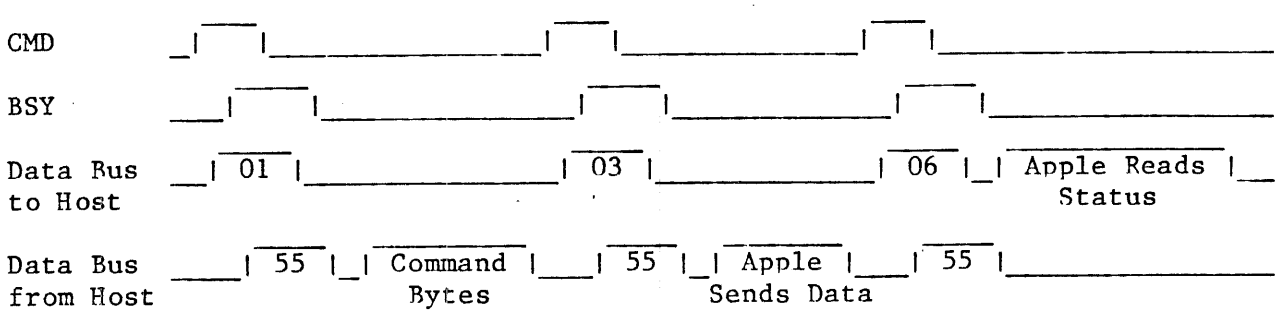
As mentioned previously, reading block FFFFFF gets Profile's spare table. The first 13 bytes are allocated for the device name, which is PROFILE followed by 6 blanks. The next 3 bytes are allocated for the device number, which is 00 00 00. The next 2 bytes are used for the program revision number, which currently is 03 90. The next 3 bytes tell how many blocks are available to the user, with the most significant first. These bytes should be 00 26 00. The next 2 bytes tell how many bytes are in each block. These bytes will be 02 14, which equals 532 decimal (however, Profile doesn't care how many bytes the host reads, nor how many bytes the host sends as long as it's not more than 532). The next byte contains the total number of spare sectors available, which is 20 hexadecimal or 32 decimal. This is followed by the number of spares currently allocated (once a spare is allocated it can never be deallocated, except by reformatting the disk), and then followed by the number of bad blocks currently in the bad block table. Finally the numbers of the spared blocks and the numbers of the bad blocks are listed (3 bytes per block number), with delimiters of FF FF FF between the spare and bad block lists and following the bad block list.

The diagrams below show how the handshaking works for each of the 3 operations supported by Profile.

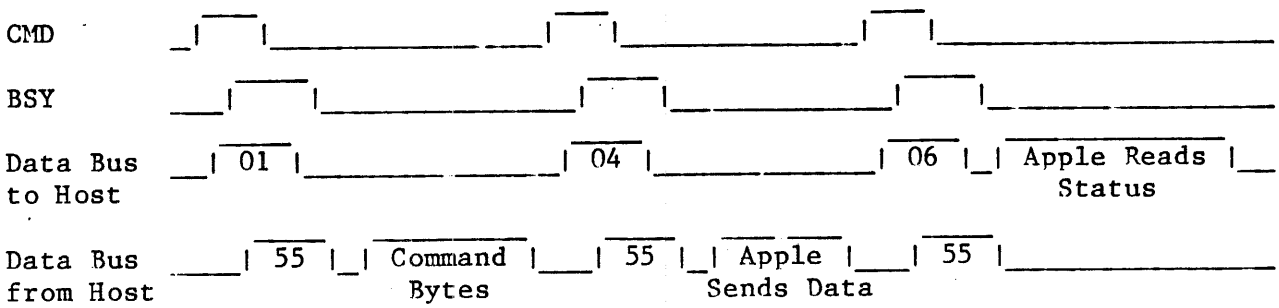
Read Operation



Write Operation



Write/Verify Operation



Important: The host must raise CMD following the last operation requested, since changes in Profile's spare and bad block tables do not get rewritten onto the disk until this occurs.

Addendum for Controller Version 3.96

Revision 3.96 of the Profile controller program has several improvements over revisions 3.90 and 3.92 (3.90 and 3.92 are identical except that 3.92 moves the stepper at half the 1.5ms per track rate used by 3.90). Instead of the Z8 falling directly into the write/verify/sparing routine if the number of errors encountered reading a block is greater than the sparing threshold specified by the driver, it rewrites the block then rereads it 100 times. If the error rate is greater than 30%, the block is spared. This 30% sparing criteria is used anytime a write/verify fails to verify, when doing a write or write/verify of a block that is in the bad block table, and when verifying a write to a spare sector. Another change is that a block is spared if the seek was able to read 3 consecutive sectors OK but a timeout error (because of not being able to find the desired header) occurred while doing a write or write/verify. Because of these changes in the sparing algorithm, the sparing threshold during the initial disk scan is now 30% instead of 50%. The last change in revision 3.96 is that the fast seek algorithm is used if the jumper at P6 on the controller board is cut, and the slow seek algorithm is used if the jumper is intact.