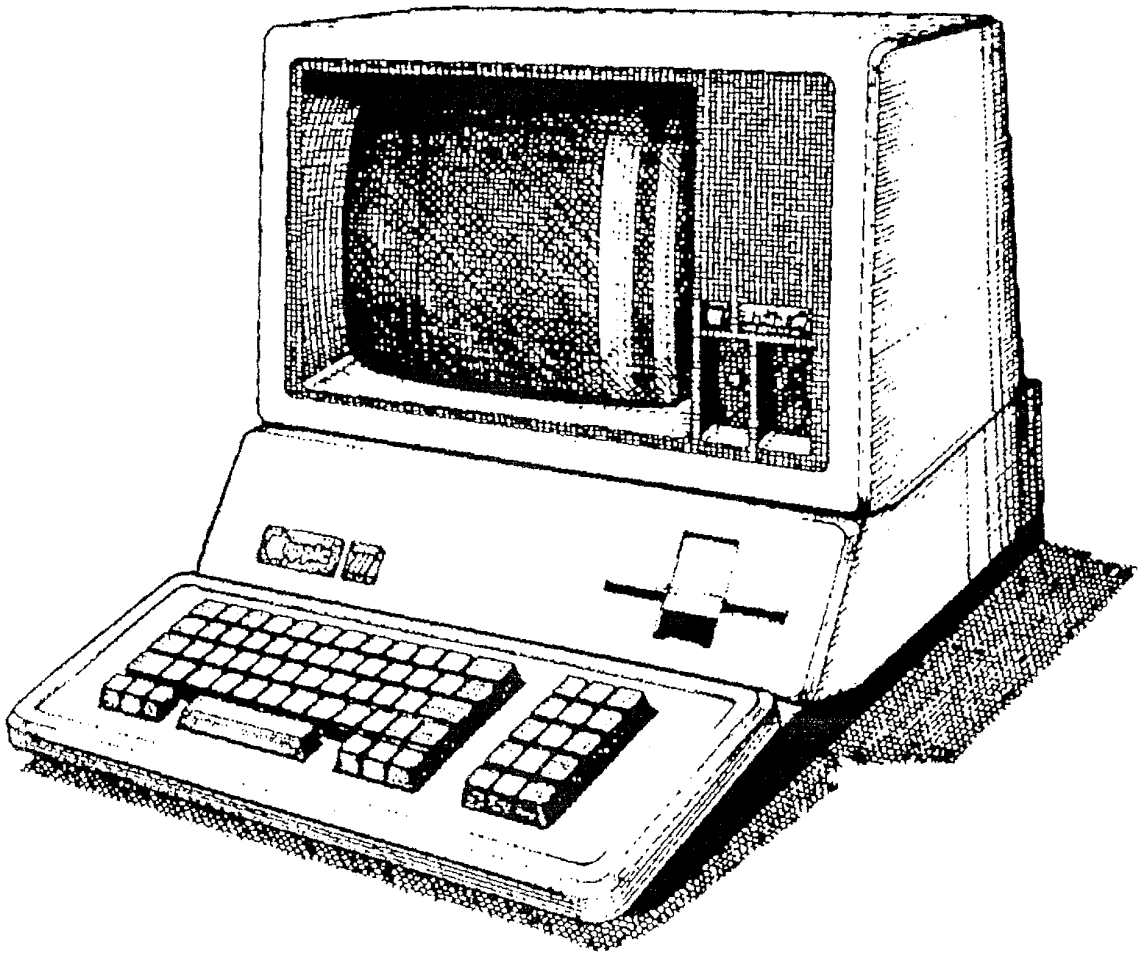
Apple /// Computer Information



DOCUMENT NAME

APPLE /// PROFILE HARD DISK REPAIR DOCUMENTS

#

78

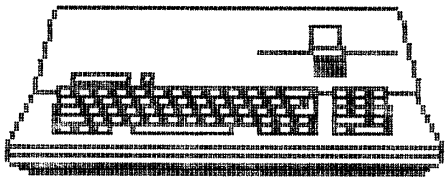
Ex Libris David T. Craig

1

PAGES: 549

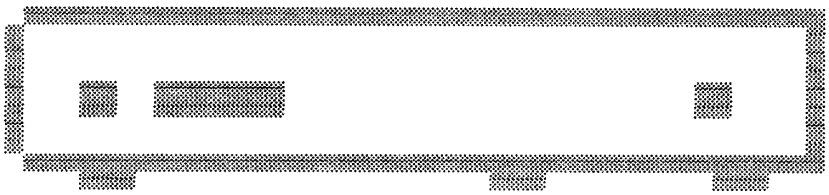
Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Documents



Apple ///

Apple ///+



ProFile

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]
71533.606@compuserve.com

2

APPLE /// COMPUTER

REPAIR INFORMATION: PROFILE HARD DISK

[DAVID T. CRAIG - 736 EDGEWATER, WICHITA, KANSAS 67230 U.S.A.]

TABLE OF CONTENTS

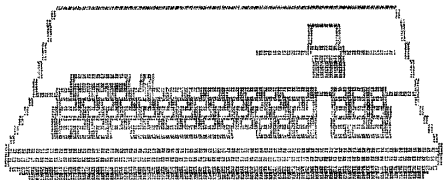
TOTAL PAGES: 579

3

1

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///

Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Sales Kit:
ProFile Technical Review

Author: Apple

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

4

Apple III

ProFile Sales Kit

ProFile Technical Review

This section of the ProFile Sales Kit will give you an overview of the technical details of the ProFile Personal Mass Storage System for the Apple ///. It is intended to be a summary only, and is not a detailed explanation of the engineering aspects of the ProFile. This information is presented to give you a better understanding of the quality, reliability, and performance that Apple has built into ProFile, and to prepare you to answer technical questions raised by your customers. Additional information regarding the ProFile can be found in "Section D" of this Sales Kit.

What is ProFile?

ProFile is a Winchester technology hard disk drive designed to operate with the Apple /// personal computer. It has a formatted storage capacity of 5 Megabytes, which is essentially the same as 35 floppy disks (the Apple /// disk drive uses floppies that have a capacity of 140K Bytes). The ProFile will, as you can see, enable you to store large amounts of data on a single device. You are also able to store files that are larger than the 140K floppies that you usually use. In addition, you are able to access your data much faster with the ProFile than with the Disk ///.

The ProFile physically consists of three main assemblies. They are the ProFile drive, the Apple /// interface card, and the interconnect cable. The ProFile has its own power supply, and as such does not rely on the Apple /// for operating power. Functionally, the ProFile consists of five major modules:

1. The Controller Card
2. The Power Supply
3. The Analog Card
4. The Head Disk Assembly (HDA) with motor control board
5. The Interface Card

The following sections describe the function of these modules in detail.

The Contoller Card

Functionally, the controller provides communcations with the Apple ///, provides signals to read and write serial data on the disk, moves the heads to the proper track, and monitors error conditions. The controller consists of a Z8 microprocessor, 2K bytes of RAM, error detection logic, and read/write control logic.

1-8

5

The Z8 provides an intelligent interface to the Apple ///. High level commands, such as read, write, and status, are passed to the Z8 through the RAM. The Z8 executes the command and passes the result of the operation back to the Apple /// through the RAM.

The controller also interfaces to the analog card to pass head control information to it. In this way the controller determines when read/write operations will take place. The Z8 controls the stepper motor to move the heads from track to track (called "seeking"), selects one of four read/write heads, and writes sector marks on the disk during formatting.

Read /Write functions are performed by the read/write logic on command from the Z8. This logic in turn controls the parallel to serial data conversion when writing, and the serial to parallel data conversion when reading. To ensure the proper transfer of data, the controller does a CRC (cyclic redundancy check) of the serial data. If an error occurs, the Z8 will automatically perform an error recovery routine and try to relocate the data onto a different section of the disk. The sector causing problems will be removed from use to prevent future errors.

To prevent heat build-up in the drive, the Z8 removes power from the stepper motor if no commands have been received for 0.75 seconds. After 3 seconds, the head is moved to a non data area of the disk to prevent accidental damage to data if a failure (such as power loss) occurs. The READY light on the front of the ProFile is lit whenever the controller is idle (not busy).

Power Supply

The power supply provides the +5VDC, +12VDC, and -12VDC needed by the ProFile for operation. The supply also contains monitoring circuitry to detect a power failure. Once a failure is detected, the head current is shut off to prevent the accidental writing of false data that would otherwise occur if a write operation were in process when the power failed. The power supply is completely shielded to eliminate the effects of electro magnetic radiation.

Analog Card

The analog card serves as the interface between the controller and the Head Disk Assembly (HDA). It consists of a data encoder, a data decoder, write driver, head select logic, automatic gain control (AGC) preamplifier, read detector, phase lock loop (PLL), and sector detector.

The head select matrix selects one of the four heads for a read or write operation. The ProFile has two fixed disks in its HDA, and there are two heads for each disk (one for each side, since the disks are double sided). Thus you have a choice of four heads, depending on which section of the disk you are trying to access. It is not necessary for you to know which section of the disk you are trying to access; the controller

2-8

6

and the analog board take care of that for you.

During a write operation, the serial data is encoded using a technique known as MFM. It is not important that you know what this is, just that it allows the maximum data storage with low formatting overhead. In otherwords, it lets you get a lot of data in a small amount of space.

During a read operation, the AGC circuit amplifies the low level head signal (.6 to 2.0 mv) to a fixed signal level (1.0 volt). The read detector simply shapes the signal so that it appears in a standard fashion. The PLL and data decoder then convert this signal back into serial data, which is passed to the controller, which in turn converts it to parallel data and passes it to the Apple ///.

When the disk is initially formatted, the sector boundaries are written to the disk (this is done by removing all read signals from certain sections of the media). During the read operation, the sector detector looks for these areas of no read signal, and signals the controller that a sector boundary has been found.

Head Disk Assembly (HDA)

The ProFile HDA is a random access storage device with two non-removable 5 1/4 inch discs as storage media. Each disk surface employs one movable head to service 153 data tracks. The total formatted capacity of the four heads and surfaces is 5 Megabytes (16 sectors per track, 532 bytes per sector, 612 tracks).

High reliability is achieved through the use of a band actuator and open loop stepper head positioning mechanism. The read/write heads are mounted on a ball bearing supported carriage which is positioned by the band actuator connected to the stepper motor shaft. The inherent simplicity of the mechanical design and electronic control allows maintenance free operation for the life of the drive (designed for over 10,000 hours MTBF). All PCB's are mounted outside the HDA for easy serviceability.

Mechanical and contamination protection for the heads, actuator, and discs are provided by an impact resistant aluminum enclosure. A self contained recirculating system supplies clean air through a 0.3 micron filter. a special spindle pump assures adequate air flow and uniform temperature distribution throughout the head and disk area. Thermal isolation of the stepper and spindle motor assemblies from the disc enclosure provides significantly greater "off track" margin (temperature changes are less likely to cause read errors). Additionally, read and write operations can be performed immediately after power on without waiting for thermal stabilization.

A brushless DC drive motor rotates the spindle at 3600 RPM. The spindle is driven directly with no belt or pulley. The motor and spindle are dynamically balanced to insure a low vibration level. A brake is used to provide a fast stop to the spindle motor when power is removed.

3-8

7

The recording media consists of a lubricated thin magnetic oxide coating on a 130 mm diameter aluminum substrate. This coating formulation, together with the low load force, low mass Winchester type "flying heads", permits reliable contact start/stop operation.

Interface Card

The Apple /// interface card serves primarily to buffer the data and decoded control lines of the Apple /// for transmission to the ProFile controller card. The interface card may be plugged into any of the four Apple /// expansion slots. Eight bi-directional data lines and five control lines are connected to the ProFile controller card with a 25-conductor cable. Each signal is buffered by an RC network fo EMI/RFI suppression. Bytes may be transferred either one at a time or by DMA at 1 Megabyte per second.

4-8

8

Apple III

ProFile - The Reliability Story

With thousands of ProFiles already installed, we have been asked to explain what Apple has done to make the ProFile so reliable. This is how we do it:

Design Simplicity:

Experience shows that the fewer parts there are, the less that can go wrong. ProFile was designed to work in harmony with the personal computer, resulting in fewer electronic and mechanical parts than any comparable Winchester design. There are fewer things that can go wrong with ProFile.

ProFile is designed to prevent problems:

Throughout the design process, the question was asked: "What can go wrong?" The answers were taken into account as the design proceeded. For instance, ProFile moves the heads to a "parking position" off the data zone after three seconds of no activity. This prevents the loss of data if, for instance, the ProFile is dropped or jarred. ProFile constantly checks for errors during operation. After any change in tracks, ProFile verifies that the operation has been performed correctly. ProFile also checks that the heads are positioned accurately on a track before any read or write operation is performed. Unless the system requests that the ProFile do otherwise, data is always verified after a write operation. In all these cases, ProFile will correct the problem so that no errors occur.

Many types of systems are prone to failures when the power is turned on or off. The ProFile power supply is designed to sense a power failure well before the internal DC voltages drop. This allows the intelligent controller in ProFile to prevent any data loss if the power fails or is turned off accidentally. The system will not begin any operation until the power is on for at least one second.

Superior System Margins:

For the user, a system with greater operating margins results in superior reliability. ProFile will operate correctly and handle data correctly under adverse environmental and mechanical conditions. For example, the packaging of ProFile enables it to withstand a one inch (2.5 cm) drop while operating, or a three inch (7.6 cm) angled drop while not operating. The HDA (head disk assembly) can be operated from 10˚C to 60˚C. This wide operating range, combined with a 75% efficient switching power supply allows ProFile to operate from 10 to 40˚C in still air without a fan!

High speed, low noise ECL (emitter coupled logic) provides wide margins for the analog electronics section of ProFile. Motor speed is kept accurate within 4%. Automatic gain control combined with a unique "gated detector" can compensate for off track operation up to 20%. Data can be read correctly with up to 50% signal degradation.

5-8

9

ProFile can Detect and Even Correct Problems:

The intelligent controller which is built into ProFile is continually checking the operation of the disk. In addition to the "problem prevention" functions already described, ProFile performs additional operations to assure that the user will never see a problem. These operations start the moment ProFile is turned on. After power-up, ProFile does a scan of the entire disk surface, and checks for any errors. Upon any data error, an extensive analysis of the error is performed to determine whether a media error exists. If that is the case, the data will be moved to a spare sector of the disk. That part of the disk with a media error will no longer be used. In most cases, the data recovery routines in ProFile will be able to extract the data even from a bad sector. The recovery operation includes more than 300 retries under various conditions. Maps of the bad sectors are redundantly recorded on ProFile so that an error in the map will not cause a problem in operation.

ProFile is Designed for Performance:

Data can be transferred from the ProFile to the system at up to one Mbyte per second DMA rate. Data is interleaved at a 5:1 ratio, which allows three 512 byte sectors to be transferred on each rotation. MFM encoding allows the maximum data storage capacity with low formatting overhead.

ProFile has been Exhaustively Tested in Design and Manufacturing.

During product development, over 200,000 hours of testing were logged. Over forty systems were customer tested for three months before volume production began. Each ProFile is tested for 184 hours prior to shipment, 60 of which are at 60˜C. This commitment to quality ensures a 10,000 hour MTBF with typical performance of over 24000 hours MTBF. ProFile is truly the most reliable product of its type.

6—8

10

PROFILE:

APPLE'S 5 MEGABYTE PERSONAL MASS STORAGE SYSTEM for the APPLE ///

FEATURES:	BENEFITS:
1. ProFile has 5 Megabytes of storage capacity	1. Simplifies large data processing and programming tasks.
2. Less than 44 cents per 1000 bytes	2. Provides cost-effective data storage.
3. ProFile accesses data 10 times faster than the conventional floppy disk drive.	3. Increases productivity.
4. ProFile's intelligent controller automatically scans for error conditions and relocates marginal data blocks elsewhere on the disk, if necessary.	4. Assures data integrity.
5. ProFile has a flexible backup scheme using the Backup /// Utility.	5. Assures data security.
6. ProFile is fully supported by Apple ///'s Sophisticated Operating System.	6. Apple /// software utilizes ProFile without requiring any changes. Different types of programs can often use the same data base.
7. Many different application programs can be stored on ProFile. Using Catalyst software from Quark Engineering, various software programs stored on ProFile, can be selected and then loaded from the hard disk without re-booting floppy diskettes when you want to go from one program to another.	7. Saves time normally lost when booting up numerous floppies. Reduces media wear because programs are booted directly from the ProFile.
8. ProFile is compact, lightweight, styled like the Apple /// and Monitor ///, simple to install, and extremely quiet.	8. ProFile is usable in any work environment.

7-8

11

MARKETS AND APPLICATIONS

ProFile is an ideal tool for....

* Financial Planners: who can make better decisions, because ProFile lets them draw on a greater data base in generating answers to questions about pricing, market share, profits, etc. In addition, all financial modeling files created with programs such as VisiCalc can be stored in one place allowing the planner to switch quickly from one file to another.

* Software Developers: ProFile lets developers keep all successive versions of programs on the same disk, thus making development time less tedious and time-consuming.

* Graphics Designers: Thanks to ProFile's fast access time, plots, charts, and finished graphics can be displayed many times faster than with conventional diskettes. ProFile also makes it possible to create highly sophisticated graphics programs that require large amounts of storage space.

* Professionals: ProFile lets doctors, dentists, lawyers, consultants, and other professionals store large client record files all in one place.

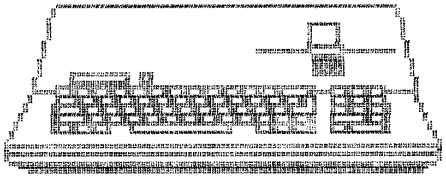
* Managers in small to medium size businesses: ProFile has the storage capacity to hold many letters, memos, reports, and other documents in a single place.

12

2

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///

Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Limited Data Recovery

Author: Apple

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

apple computer inc.

6 pages

13

ProFile Technical Procedures

Section 1

Apple /// ProFile Limited Data Recovery

Contents:

apple computer inc.

14

ProFile Technical Procedures

Apple /// ProFile Limited Data Recovery and Exchange

Contents:

Section 1

Section 2

15

apple computer inc.

A. INTRODUCTION

This ProFile Limited Data Recovery Program is used with Apple /// ProFiles. It attempts to copy customer's files from a damaged ProFile to the exchange unit. Although there is no guarantee that data can be transferred, in most cases data recovery is possible if the ProFile passes the self test after being turned on.

If the ProFile READY light does not come on to a steady state, data cannot be recovered using this program. Special arrangements must be made with Level 2 for data recovery.

The importance of regular backups should be emphasized to the customer. The customer should have a copy of the Backup /// diskette (part number 681 0032). The documentation for this is "Apple /// Backup User's Manual" (part number 030 0381).

CAUTION: Before continuing with this procedure use the "Apple /// Confidence Program" (part number 681 0031) to ensure that the Apple /// is functioning properly. Failure to do this could possibly damage the ProFile!

B. WHAT YOU SHOULD DO FIRST

First run the confidence program and make sure that the Apple /// is functioning properly. Then make a back-up copy of the Recovery Program diskette! You will be using a system with known bad hardware attached to it, so don't take a chance of destroying the software accidentally. Put the original in a safe place.

C. SOFTWARE OVERVIEW

The diskette accompanying this document contains software designed to recover a large portion of data found on a damaged ProFile. In order to use this program the customer's ProFile must complete the power-up cycle, which includes a testing sequence. That is, the red "READY" indicator must be on and steady. If this does not occur, data recovery is not possible.

The program copies data from the customer's ProFile to the exchange unit. If the program has difficulty writing to the Exchange ProFile, it should stop. Something could be wrong with the Exchange unit, so try another Exchange unit. As data is copied, those blocks which the program had problems reading are identified. After the blocks are copied, all further operations are performed on the exchange unit.

apple computer inc.

The program examines each file to see if any of its blocks were among those which couldn't be copied. A printed record indicates the files that couldn't be copied as being suspect and, most probably, unusable. While this is going on, the master allocation map for the entire ProFile is also being rebuilt. The allocation map tells which blocks are used and which are free for use.

Our testing has shown that of the ProFiles that become "READY", the program is able to recover most all of the customer's data.

D. EQUIPMENT REQUIRED

Recovery Program Diskette
Apple /// System (256k)
Monitor
Compatible Printer
Printer Interface Card (if not using the Silentype printer)
ProFile Interface Card (2 required)
ProFile Interface Cable (2 required)

E. CONFIGURING THE SOFTWARE

The Recovery Program software is configured for two ProFile drivers and the Silentype printer. If you use the Silentype printer no configuration is necessary. If you wish to use a printer which uses the built-in serial port or an interface card, use the System Configuration Program to add the appropriate driver. (Refer to the Standard Device Drivers Manual.) Delete the .PRINTER driver already configured. Whatever printer driver you use must be named **.PRINTER** as the program looks for this name for its output.

Do not under any circumstances, make any changes to either of the two ProFile drivers. These are named **.GOOD** and **.BAD**, and must remain that way for the program to run correctly.

When you have the software properly configured, make another copy and use it as your working diskette. There is really no point in taking the chance of destroying the one you just configured.

17

apple computer inc.

F. SETTING UP THE SYSTEM

1. Turn the power off on the Apple /// and remove the cover.

2. If necessary, install an interface card for the printer in the slot for which you configured its driver.

3. Configure the Apple /// with two ProFiles as follows: Install the ProFile interface cards in slots three (3) and four (4). These are the two right-most slots as you look at the Apple from the keyboard side. Seat the cards firmly in the slots.

4. The ProFile Electronic Module contains two circular markers, one red and one green. Place the red marker on the cable connected to the ProFile card in slot 4. Place the green marker on the cable connected to the card in slot 3. (Addition markers are included in the software package.)

   NOTE: You can leave this set-up and whenever you want to use the data recovery procedures all that is necessary it to connect the ProFiles (see next section).

5. Connect the printer and monitor.

G. RUNNING THE PROGRAM

NOTE: Before you running the Limited Data Recovery Program, attach the LED to the Exchange ProFile. (See Section 2 for directions.)

1. Turn on the monitor.

2. Turn on the printer and make sure it is on-line.

3. Connect the (good) Exchange ProFile to the cable marked with the green marker.

4. Connect the customer's ProFile to the cable marked with the red marker.

   CAUTION: Check that the cables are connected correctly or you will copy files from the Exchange ProFile to the customer's, eliminating any chance of saving any files.

5. Connect the power cables to both ProFiles and turn them on.

18

apple computer inc.

6. **Wait** until the "READY" lights on both ProFiles show steady red. This will take a minute or so.

7. Place the Recovery Program Diskette in the Apple /// built-in drive.

8. Turn on the Apple /// and the program will start.

9. When the program starts you will see the message:

   Watch the ready light on the EXCHANGE ProFile.
   Please press the RETURN key when you're ready.

   When you press RETURN, the READY light on the Exchange ProFile should flash once, indicating that the cables are connected correctly.

   If the light does not flash, you may have connected the cables incorrectly. Press the ESCAPE key (to prevent the recovery process), check the cables and try again.

10. As a final check, you will see the following displayed:

    Are you sure you want to overwrite PROFILE? (PROFILE is the name of the Exchange ProFile.)

    Enter "Y" if the ready light flashed, or
    ESCAPE if the ready light did not flash:

    Answer yes by entering a **Y** to begin the recovery process.

11. An asterisk (*) in the upper right hand corner of the screen will blink to indicate that the recovery operation is working correctly. During the operation, different messages will appear on the screen to let you know what is happening within the program. The operation can take anywhere from 15 to 30 minutes, depending upon how much data can be recovered.

12. As sections of the program are completed, messages will be listed to the printer.

13. When the program is done, the Apple bell will beep about every five seconds. Press the ESCAPE key to stop the bell and the program.

14. Power down both ProFiles. The exchange unit goes to the customer, and the other unit goes to Level 2.

15. Turn the equipment off, remove the program diskette and put it away.

19

apple computer inc.

H. INTERPRETING THE RESULTS

Give your customer the printed results of the data recovery procedures and a copy of how to interpret these results. In the following explanation, examples of the printout are shown in bold face printing. These statements are interpreted as follows:

Apple /// Limited Disk Recovery Program
Results:
Trouble accessing **0 blocks** (a number other than 0 might appear)

All accessible directories are listed below. Individual files which contain problem blocks are also listed.
The resolution of the individual files on the ProFile is shown as a running status, displaying the directory names as they appear on the ProFile. If a file contains one or more "suspect" blocks, its name will appear.

These files are not deleted as they may, in fact, still be partially usable. An example of this would be as ASCII file, such as one created by AppleWriter. An attempt should be made by the customer to access these files. If the attempt results in "unusual" things happening, then delete them.

Block Conflict Report
This message appears if two or more files claim use of the same block .Another pass through the ProFile directories is made to resolve it. Along with this message the directory names are listed on the print-out. Those files which conflict with each other will be listed. An attempt is made to repair the problem. If the conflict is resolved, no further mention is made of the files, otherwise the following message appears: **Unable to repair all block conflict errors. Try your files.**

Summary of Allocation Map Changes
This summarizes the changes made to the master allocation map. This will include the total number of blocks available on the ProFile (**blocks on volume**), the number of blocks in use (**blocks used**), and the number still available (**blocks available**).

Block(s) released from allocation map
A statement will also appear regarding the number of blocks, if any, released from the allocation map. If blocks were released, the original allocation map "thought" that more blocks were in use than really were. The ProFile returned to your customer contains the updated, correct allocation map.

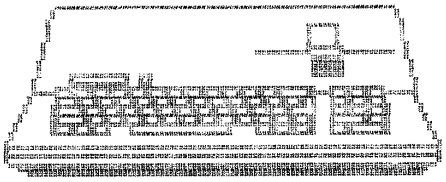
(The End)

20

3

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Exchange Procedures

Author: Apple

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

21

apple computer inc.

11 pages

ProFile Technical Procedures

Section 2

ProFile Exchange Procedures

Contents:

22

apple computer inc.

A. INTRODUCTION

When a customer returns a ProFile for servicing, the unit will be exchanged for a spares kit ProFile. Before you exchange the customer's ProFile, run the Limited Data Recovery program (described in Section 1) if there is data the customer wishes to try to recover. Exchanging the ProFile consists of removing the cover, the L.E.D. Cable Assembly and rear plates from the customer's ProFile, then replacing the LED assembly and placing the rear plates and the customer's cover on the Service Spares Kit (Exchange ProFile) unit.

The shipping cover marked "SHIPPING FIXTURE" is attached to the ProFile to be serviced and the entire unit is placed in the same packaging as was the spares kit, and sent back for servicing.

B. MATERIALS REQUIRED

Diagonal cutters ("dikes")
Tie Wraps
Medium Phillips Screwdriver
Protective Pad
Small Flatblade Screwdriver

CAUTION: The ProFile is a mechanical device with motors and moving parts. Rough handling such as dropping the drive, sharply jarring it or allowing heavy objects to fall on it can cause a malfunction. Whenever it is necessary to turn the ProFile over, be sure to rest it on a protective pad.

23

apple computer inc.

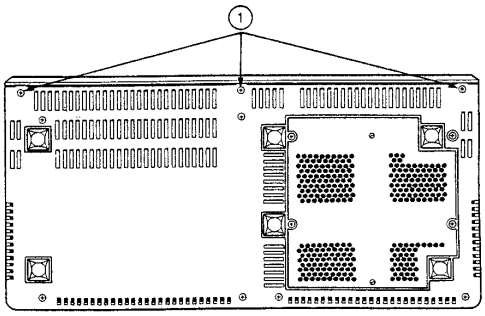


FIGURE 1

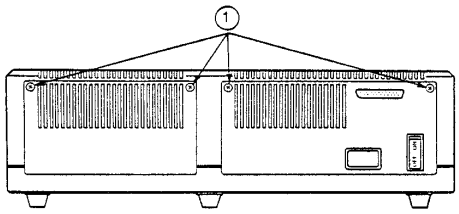


FIGURE 2

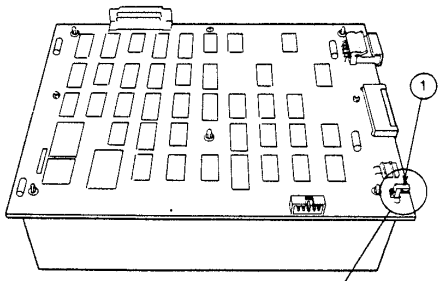


FIGURE 3

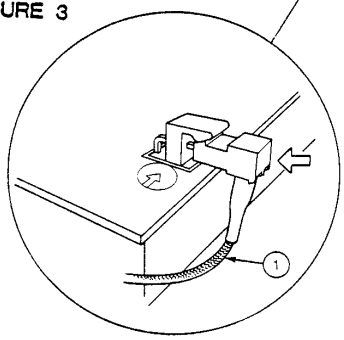


FIGURE 4

24

apple computer inc.

C. REMOVING THE COVER

1. Make sure the customer's ProFile is turned off. Disconnect the power cord and interface cable (ribbon cable) from the back of the ProFile.

2. Turn the ProFile over, lay it on the protective pad, and remove the three Phillips-head screws from beneath the front panel (Figure 1, #1).

3. Turn the ProFile right side up; loosen and remove the four screws on the back of the unit and the two rear plates (Figure 2, #1).

4. Lift the cover off carefully and rest it on the far side of the case, taking care not to pull on the LED cable.

5. Unplug the LED cable from its socket on the controller board (Figure 3, #1).

6. Remove the cover marked "SHIPPING FIXTURE" from the Service Spare parts module. Note the LED assembly is not attached.

25

apple computer inc.

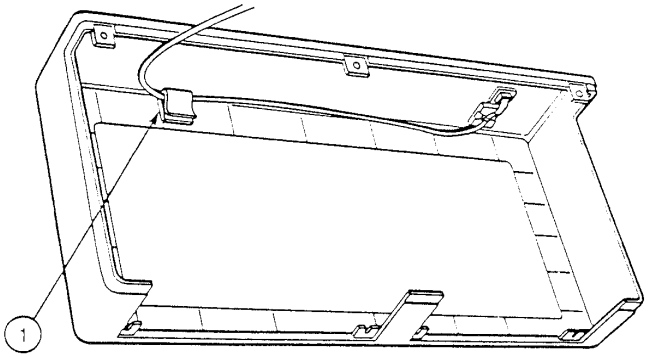


FIGURE 5

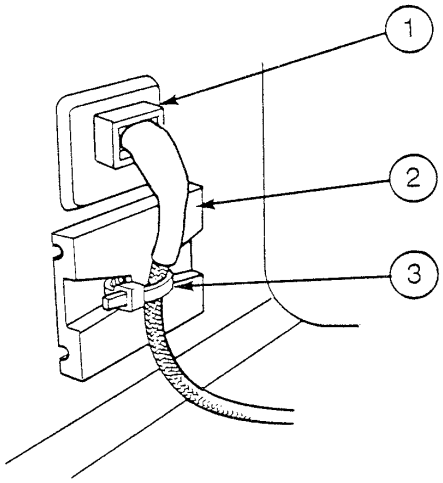


FIGURE 6

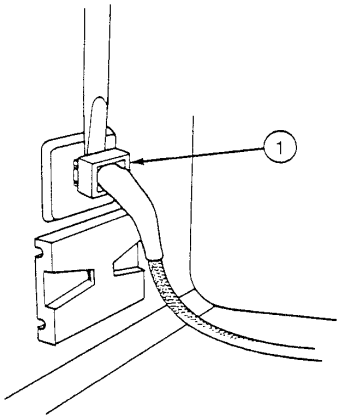


FIGURE 7

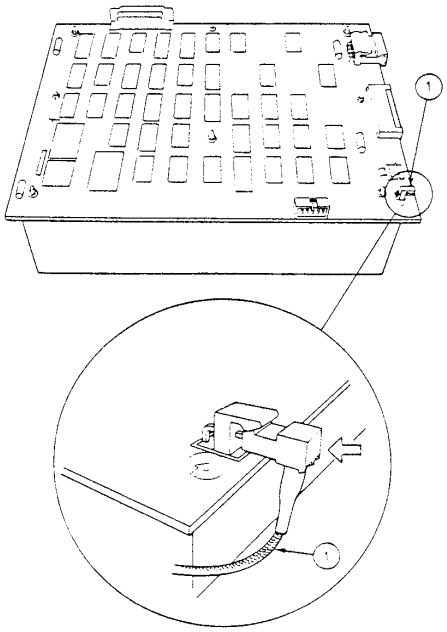


FIGURE 9

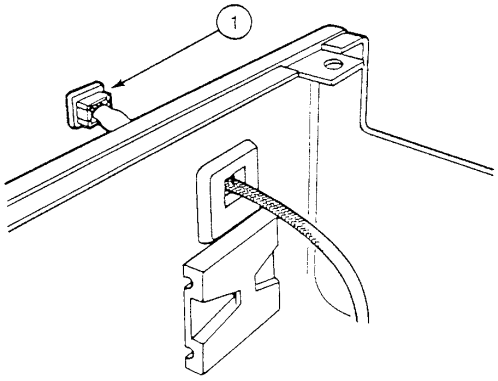


FIGURE 8

26

apple computer inc.

D. REMOVING THE LED ASSEMBLY

1. The ProFile cover is removed and the LED cable is disconnected from the controller card.

2. With the customer's cover laying flat, as in Figure 5, cut off the white plastic tie (Figure 6, #3) that holds the LED cable to the white holder in the cover (Figure 6, #2).

3. Remove the other end of the LED cable from the other holder in the cover. (On some ProFiles, this holder will be a clamp like the one shown in Figure 5, #1; on others, it will be like the one shown in Figure 6.)

4. With a flathead screwdriver, pry the cable clamp off the back of the LED (Figure 7, #1) and slide it down the cable, out of the way.

5. Firmly push a few inches of the cable out through the slot in the cover (behind the "Ready" label) as shown in Figure 8. You will have to remove the "Ready" label around the LED opening on the cover to free the LED.

6. Around the red LED is a small black plastic mount. Remove the mount (Figure 8, #1) by pushing out its side flaps and sliding it off the LED.

7. Pull the cable back through the hole in the case.

E. INSTALLING THE NEW LED ASSEMBLY

8. Take the LED from the Service Spares Kit and thread it through the front opening in the cover. Place the small black plastic mount on the LED (see Figure 8, #1), then pull the cable back through the opening until the LED fits in its slot. Replace the "ready" label with the new label found in the spares kit..

9. Push the cable clamp (Figure 6, #1) up to the cover until it holds the cable steady.

10. Place the LED cable against the white plastic holder and fasten it with a tie wrap, contained in the spares kit, (Figure 6, #2 and 3). Cut off excess tie wrap.

11. Place the cable in the other holder (Figure 5, #1), using a tie wrap if necessary. Cut off excess tie wrap.

12. Connect LED cable to controller card (Figure 9). Place Customer's LED Cable Assembly and old "Ready" label in the bag, from which you got the new assembly, to be returned with the ProFile to be serviced. Put the customer's cover on the Spares Kit ProFile.

27

apple computer inc.

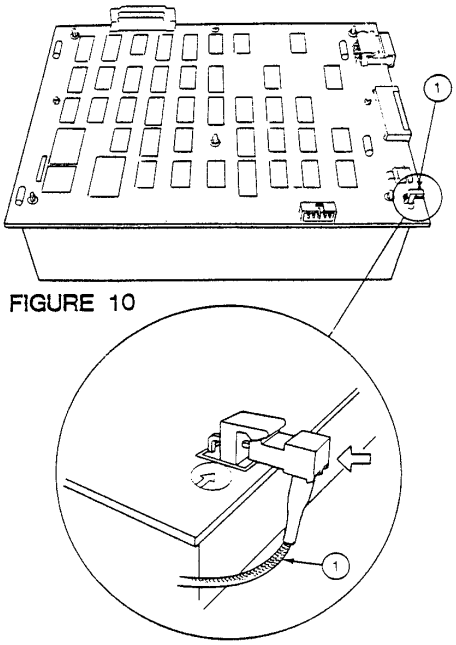


FIGURE 10

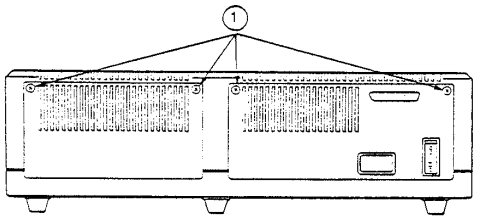


FIGURE 11

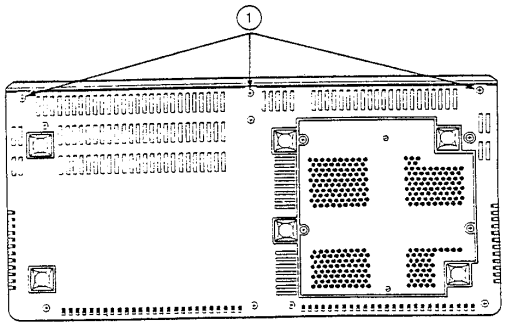


FIGURE 12

28

apple computer inc.

F. REPLACING THE COVER AND REAR PLATES

1. Attach the LED cable to its connector on the controller card (Figure 10). Make sure the LED cable exits down and away from the card (Figure 10, #1).

2. Replace the two rear plates and the four screws. Do not tighten the screws at this time (Figure 11).

   NOTE: The serial number is stamped on one of the rear plates and must be transferred to the ProFile which the customer will keep.

3. Replace the customer's ProFile cover on to the spares module. (Hint: The four slots on the back of the cover fit between the inner and outer rear plates. Line up the back first; then pull the cover gently forward and down. Check around the cover to make sure the LED cable isn't caught between the cover and the base.)

4. Tighten the four rear-plate screws.

5. Turn the ProFile over and replace the three screws on the front edge (Figure 12).

6. Turn the ProFile right side up. Reinstall the power cord and the interface cable.

   NOTE: Before replacing the cover on the unit to be serviced, lay the used LED Cable Assembly from the customer's ProFile on top.

7. Replace the cover marked "SHIPPING FIXTURE" on the unit to be returned for servicing.

29

apple computer inc.

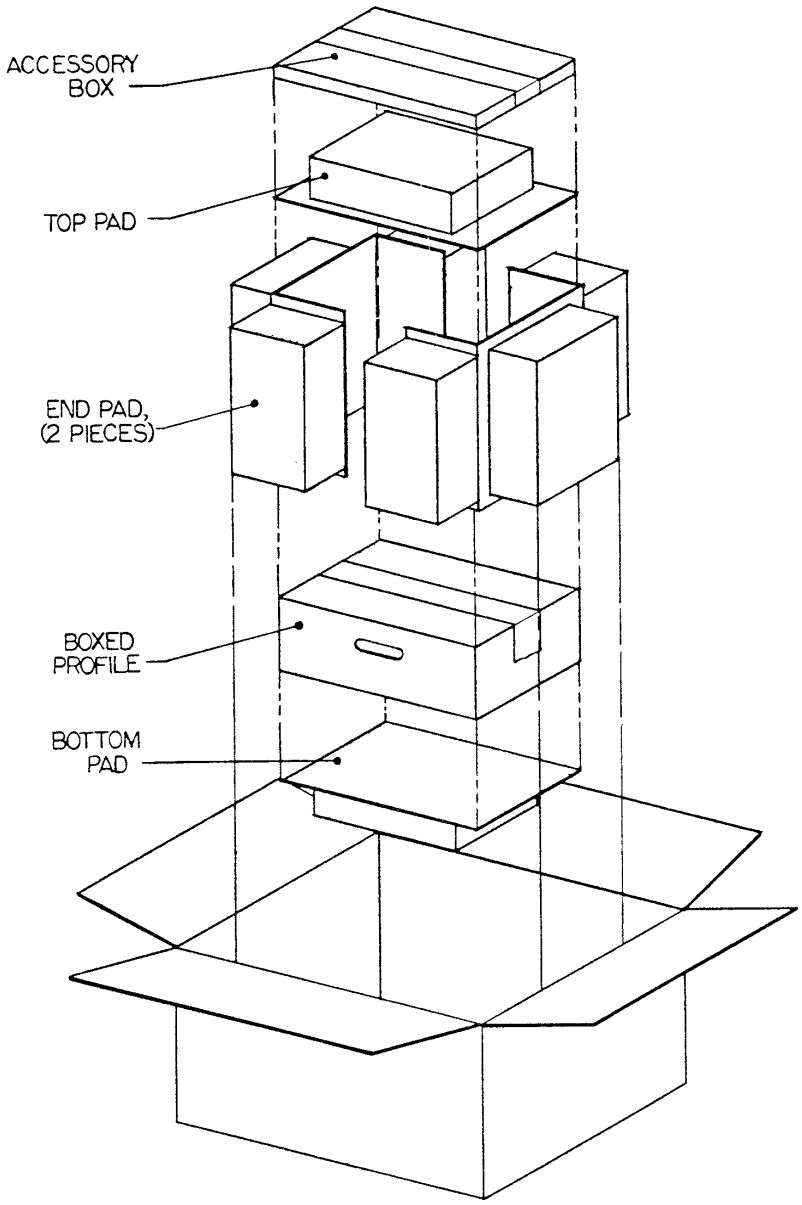


FIGURE 13

30

apple computer inc.

G. PACKAGING FOR SHIPMENT

The following instructions refer to Figure 13 unless otherwise stated.

1. Set Packing Box on the floor with all flaps facing outside the box.
2. Bend the two End Pads and slide them into the box with the foam against the sides of the box.
3. Slide Bottom Pad, with the foam side toward the bottom of the box, into position between the end pads.
4. Place End Caps snuggly onto the ends of the ProFile (Figure 14).
5. Place the boxed ProFile on top of the Bottom Pad.
6. Lay the Top Pad with the foam padding facing away from the ProFile, on top of the ProFile.
7. Place the Accessory Box, which contains the used LED Assembly, on top of the ProFile.
8. Close the end (short) flaps; then the side flaps.
9. Secure the flaps using shipping tape.
10. Secure shipping documentation to the top of the box.

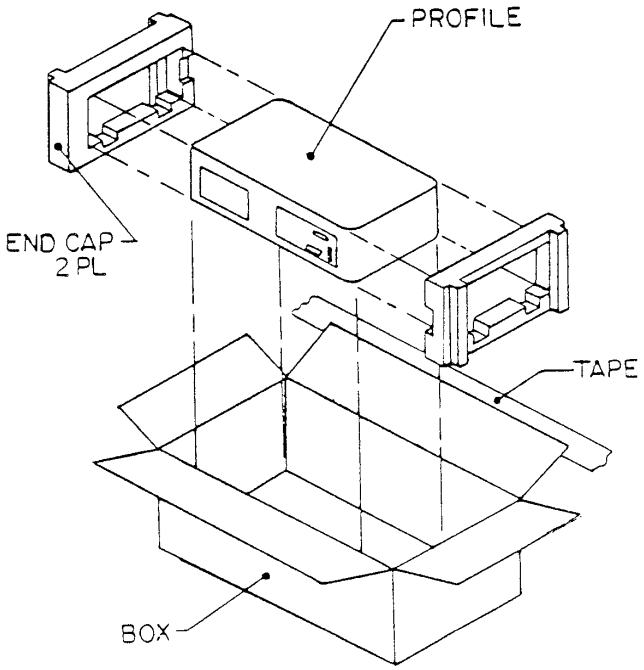


FIGURE 14

31

apple computer inc.

H. SUMMARY OF PROCEDURES CHECKLIST

___ Remove the spares kit from its packaging. Keep the shipping materials to package the unit to be sent in for servicing.

___ Remove cover marked "SHIPPING FIXTURE" from the spares kit and verify that all modules are present, all connections are proper, and that there are no loose or broken wires or foreign materials.

___ Turn on monitor.

___ Turn on printer and make sure it is on-line. (The driver is already configured for the Silentype printer.)

___ Connect LED on Exchange ProFile.

___ Connect Exchange Profile (use slot 3 for the Interface card).

___ Connect customer's ProFile (use slot 4 for the Interface card).

___ Turn on Apple /// and run the Limited Data Recovery program.

___ Power down and disconnect both ProFiles.

___ Remove both rear plates from the customer's ProFile.

___ Remove the cover from the customer's ProFile.

___ Remove and replace the LED assembly on the cover of the customer's ProFile, using the new LED front panel label.

___ Attach the two rear plates from the customer's ProFile onto the Exchange (spares kit) ProFile). Note that the serial number is on one of these plates.

___ Replace the customer's cover onto the Exchange ProFile fitting it into place over the two rear plates.

___ Place the used LED assembly along with the old front panel label in the assembly box.

___ Attach the cover marked "SHIPPING FIXTURE" onto the unit to be serviced.

___ Pack the unit to be repaired and the box containing the used LED assembly in the spares kit packaging.

___ Complete all shipping documentation.

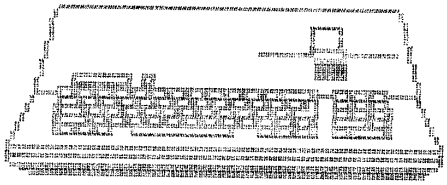
(The End)

32

4

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Pro-File Level II Phase 1 Service Manual (Preliminary)

Author: Apple

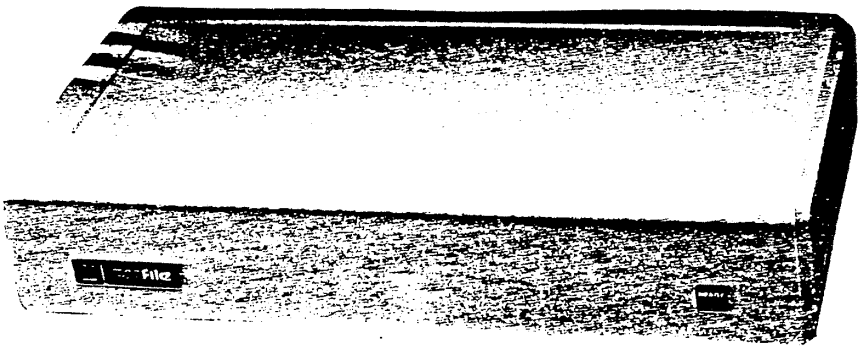
DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

33

Pages: 238

PRO-FILE LEVEL II
PHASE 1 SERVICE MANUAL

PRELIMINARY



34

PROFILE EQUIPMENT LIST

1) DEBUGGER:
   FIRMWARE.....889-9007
   Z8.....338-8603

2) SOFTWARE DUPLICATION:
   FST.....889-0029
   QUICK DEBUG (Ver. 7).....889-0004
   FORMAT CERTIFY.....889-0013
   BIG DEBUG (Ver. 17).....(INCLUDED)

3) UPGRADE KIT:
   MASKED Z8 MICROPROSSOR.....341-0171
   6.36 x 5/16 STANDOFF.....860-0213
   5.1K OHM RESISTOR.....101-4512
   0.1 MICROFARAD CAPACITOR.....130-0007
   1K OHM RESISTER.....101-4102
   330 OHM DIP PACKS.....112-0105
   3.9K OHM RESISTER.....101-4392
   26 TO 30 GAUGE INSULATED WIRE.....N/A

4) MISCELLANEOUS TOOLS:
   TEST LED.....590-0047
    (CUT DOWN)
   JUMPER
    (USED DURING FORMAT)
   ALLEN (hex) DRIVERS
    (BRAKE ADJUSTMENT 5/64)
    (TRACK ADJUSTMENT .050)

35

PRO-FILE PHASE 1 SERVICE MANUAL TABLE OF CONTENTS

36

PRO-FILE PHASE 1 SERVICE MANUAL TABLE OF CONTENTS

Section 3 Appendices

37

HOW TO USE THIS MANUAL

The Pro-File Phase 1 Service Manual is divided into 3 sections:

SECTION 1 TROUBLESHOOTING:

This section contains a flowchart procedure and corresponding explanation for troubleshooting a problem Pro-File to the faulty module.

SECTION 2 SERVICE PROCEDURES:

This section contains the various procedures needed to test, adjust, and remove or replace the modules on the Pro-File.

SECTION 3 APPENDICES:

This section contains various descriptions referred to by the other sections for the purpose of providing general and specific information on Pro-File operation.

38

SECTION 1
TROUBLESHOOTING

39

SECTION 1 TABLE OF CONTENTS

40

IMPORTANT READ THIS

HOW TO USE THIS SECTION

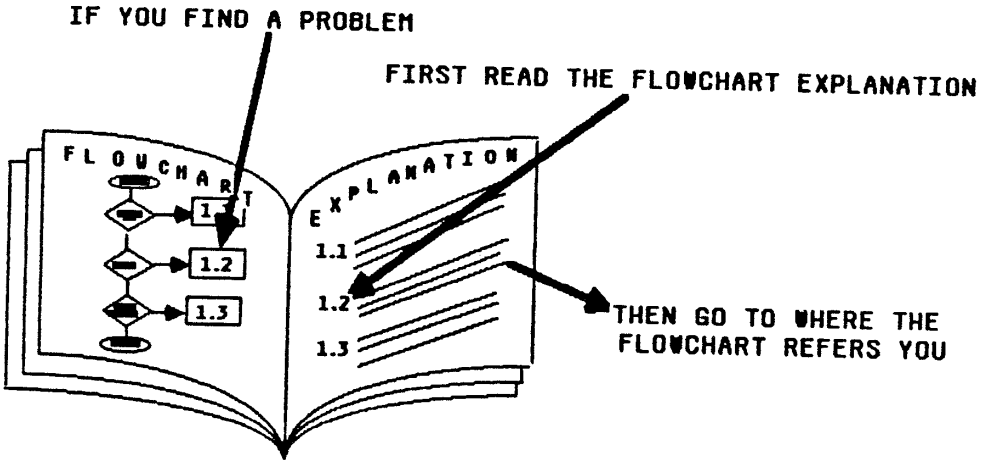
The procedure in this Troubleshooting Section is concerned with isolating a problem in the Pro-File to a malfunctioning module (i.e. Controller PCB, Analog PCB, Hard Disk Assembly (HDA), or Power Supply).

CAUTION: The Hard Disk Assembly (HDA) is a very expensive and fragile device, handle it gently to avoid damage.

To troubleshoot the Pro-File use the flowcharts in this section beginning with the main flowchart on the next page.

If you find a problem, go to the opposite page and read the explanation for the check that failed. This explanation will provide you with valuable information on the reasoning behind the flowchart, and will also provide you with references to sections 2 and 3 for needed procedures and additional information. After reading the flowchart explanation, you may if you wish take the action indicated at the NO branch of that check.

* If you are not familiar with Pro-File operation you may wish to read the Pro-File HDA description and/or the Pro-File Overview in the Appendices Section.



41

START HERE

SHEET 1

10/19/83

REMOVE THE COVER FROM THE PROFILE
INSERT TEST LED AT P4 ON THE CONTROLER PCB
(SERVICE PROCEDURES: IF INSTRUCTIONS ARE NEEDED)

READ 1.1

DOES CONTROLLER PCB HAVE SYSTEM Z8 WITH MASKED ROM ?

NO

INSTALL A SYSTEM Z8 WITH MASKED ROM ONTO THE CONTROLLER PCB

YES

OBSERVE THE SCAN SEQUENCE (BELOW) AS YOU TURN ON THE PROFILE:

NOTE: SCAN SHOULD TAKE NO LONGER THAN 90 SEC

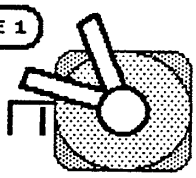
1. THE READY LAMP SHOULD COME ON FOR LONGER THAN 0.5 SECOND THEN GO OFF
2. AFTER ABOUT 20 SECONDS, THE INTERRUPTER ARM ON THE HDA SHOULD MOVE TO THE "TRACK 0" POSITION (SHOWN IN FIGURE 2)
3. THE READY LAMP SHOULD BLINK AS THE INTERRUPTOR ARM STEPS THROUGH EACH TRACK FROM TRACK 0 TO PARK POSITION

NOTE:
a. THE INTERRUPTER ARM MAY MOMENTARILY GO TO THE SPARES TABLE AT THE CENTER OF ITS TRAVEL FOR SPARED SECTORS. THIS IS NORMAL (FOR MORE INFORMATION ON THE SPARES TABLE REFER TO THE FORMAT EXPLANATION IN APPENDICES SECTION)

b. IF AN ERROR IS DETECTED WHILE READING A SECTOR, THE INTERRUPTER ARM MAY HESITATE AT A SINGLE TRACK FOR SEVERAL SECONDS TO VERIFY THE MEDIA THIS IS NORMAL

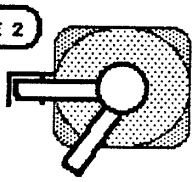
4. AT THE END OF SCAN THE INTERRUPTOR ARM SHOULD RETURN TO PARK POSITION (FIGURE 1) AND STAY THERE THEN THE READY LAMP SHOULD COME ON STEADILY

FIGURE 1



INTERRUPTOR ARM IN PARK POSITION

FIGURE 2



INTERRUPTOR ARM IN TRACK 0 POSITION

READ 1.2

SCAN OK ?

NO

GO TO NO SCAN FLOWCHART ON PAGE 1.14

YES

BOOT AND RUN THE QUICK DEBUGGER ON THE PROFILE, SAVE THE HARD COPY PRINTOUT. (IF INSTRUCTIONS ARE NEEDED REFER TO SERVICE PROCEDURES)

GO TO SHEET 2

42

1.1 When the Pro-File first came out the firmware program used for the Z8 microprocessor on the Controller PCB was contained on a ROM chip located piggyback on the Z8. After this version had been out in the field awhile, it was found that due to the heat expansion that occurred when the machine was turned on, the leads from the ROM's piggyback socket were intermittently separating from the Z8.

Since the Z8 would be looking for program instructions when this happened, the temporarily open input would be interpreted as a bit in an instruction code and cause the Z8 to do strange things including putting the Pro-File into Write mode as the heads were doing a seek to a target track. This of course destroyed any sector header and/or data block fields that happened to be passing under the heads as they were on their way to the target track.

Later when a read of the damaged sectors was tried, either as a result of the next power up Scan operation, or a read command from the host computer system, the Z8 would detect the problem and ultimately spare the sector which of course made the malfunction look like a media problem.

The logical solution for the imagined media problem (the Z8 problem was not known at the time) was to replace the now badly formatted (though undamaged) HDA with a new HDA. This action would cause the Pro-File to work fine until the piggyback ROM again separated from the Z8. The HDA would be replaced again, ... etc.

Needless to say this process was not very effective. The **masked ROM Z8** has been developed to be an effective solution to this problem. All **piggyback Z8s** are to be replaced with this version. It is projected that most of the current Pro-File problems will be remedied by this upgrade. **(Now return to the point in the flowchart which referred you to this discussion.)**

1.2 All functions performed by the Controller and Analog PCBs and the HDA are ultimately controlled by the firmware program in the Z8 microprocessor on the Controller PCB. During power up on the Pro-File the Z8 will have the Pro-File go through a Scan operation in which certain specific things are checked for. A detailed explanation of the Scan operation may be found in Section 3 of this manual. **(Now proceed to the NoScan flowchart.)**

43

SHEET 2

SHEET 2

LESS THAN 10 SPARED SECTORS AND 5 BAD BLOCKS ?

NO

READ 2.1

GO TO DP FLOWCHART IN THIS SECTION

YES

TURN OFF THE PROFILE THEN:

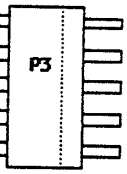
1. EXAMINE THE PCB'S FROM THE PROBLEM PROFILE FOR BENT PINS, BROKEN LEADS, CRACKED BOARDS, ETC.
2. INSTALL UPGRADES ON THE CONTROLLER AND ANALOG PCB'S, IF NECESSARY
3. CHECK & ADJUST THE HDA BRAKE AND INDEX SENSOR, IF NECESSARY

(IF INSTRUCTIONS ARE NEEDED REFER TO SERVICE PROCEDURES)

TURN THE PROFILE ON AND MEASURE THE POWER SUPPLY VOLTAGES AT P3 OF THE CONTROLLER PCB (REFER TO FIGURE 3)

FIGURE 3

PIN #	VOLTAGES
1	+5V ± 0.1V
2	+12V ± 1.2V
3	POK +5V ± 0.3V
4	-12V ± 1.2V
5	GROUND



ARE THE VOLTAGES WITHIN TOLERANCE ?

NO

READ 2.2

WHILE LEAVING EACH MODULE IN PLACE, DISCONNECT CONNECTORS J4, J3, AND J2 (J4 CONNECTS THE HDA TO THE POWER SUPPLY, J3 CONNECTS THE CONTROLLER PCB TO THE POWER SUPPLY, AND J2 CONNECTS THE CONTROLLER PCB TO THE ANALOG PCB) THEN CONNECT A KNOWN GOOD CONTROLLER PCB INTO J3 AND RECHECK EACH VOLTAGE

YES

ARE THE VOLTAGES WITHIN TOLERANCE ?

NO

REPLACE THE POWER SUPPLY

YES

ONE AT A TIME, RECONNECT THE FOLLOWING:

IF THE PROBLEM REDCCURS AFTER THE RE-ATTACHMENT OF A MODULE THEN THAT IS PROBABLY THE PROBLEM MODULE

1. J3 TO THE OLD CONTROLLER PCB
2. J2 TO THE OLD CONTROLLER PCB (POWER TO ANALOG PCB)
3. J4 TO THE HDA

GO TO SHEET 3

44

2.1 The Quick Debugger program reads the Spares table on the Profile and prints the original sector location of every spared sector listed in that table. Additional information on sparing sectors may be found in the Special Function Tracks explanation in the HDA format description in the Appendices Section of this manual. At this point in the flowchart you know that Scan worked OK, but that the HDA has spared too many sectors. Spared sectors can be caused by either:

1. **Electronic problems** causing a sector to be unreadable or somehow alter its data bit pattern at the original sector location on the disk. For example the Z8 may have malfunctioned as it was performing a Seek, and instructed the electronics to write as the heads were moving across the media to a target track. This of course would destroy the parts of all the sectors that the head passed over on its way to the target track.
2. **Media problems** making it impossible to record magnetic information on a sector where it occurs. This could be a manufacturing defect in the media (the magnetic coating on the disk), or a ding in the media caused by head contact.

One way to determine whether a spared sector was caused by a media problem or an electronic problem is to continuously read the questionable sector and examine its analog signal with an oscilloscope. The Quick Debugger printout gives you a list of spared sectors that you might want to look at. The DP (Diagnostic Procedure) flowchart that you are being referred to, will explain how you can continuously read a spared sector and interpret its analog signal. **(Now proceed to the DP Flowchart.)**

2.2 A missing or out of tolerance voltage can be caused by any of the modules in the Pro-File. The method used in this flowchart to isolate the problem module is to disconnect the power supply from all the modules, plug in a known good Controller PCB and monitor the voltages at P3 (the power supply connection to the Controller PCB). If the power supply is at fault then the voltage/s will still be missing or out of tolerance.

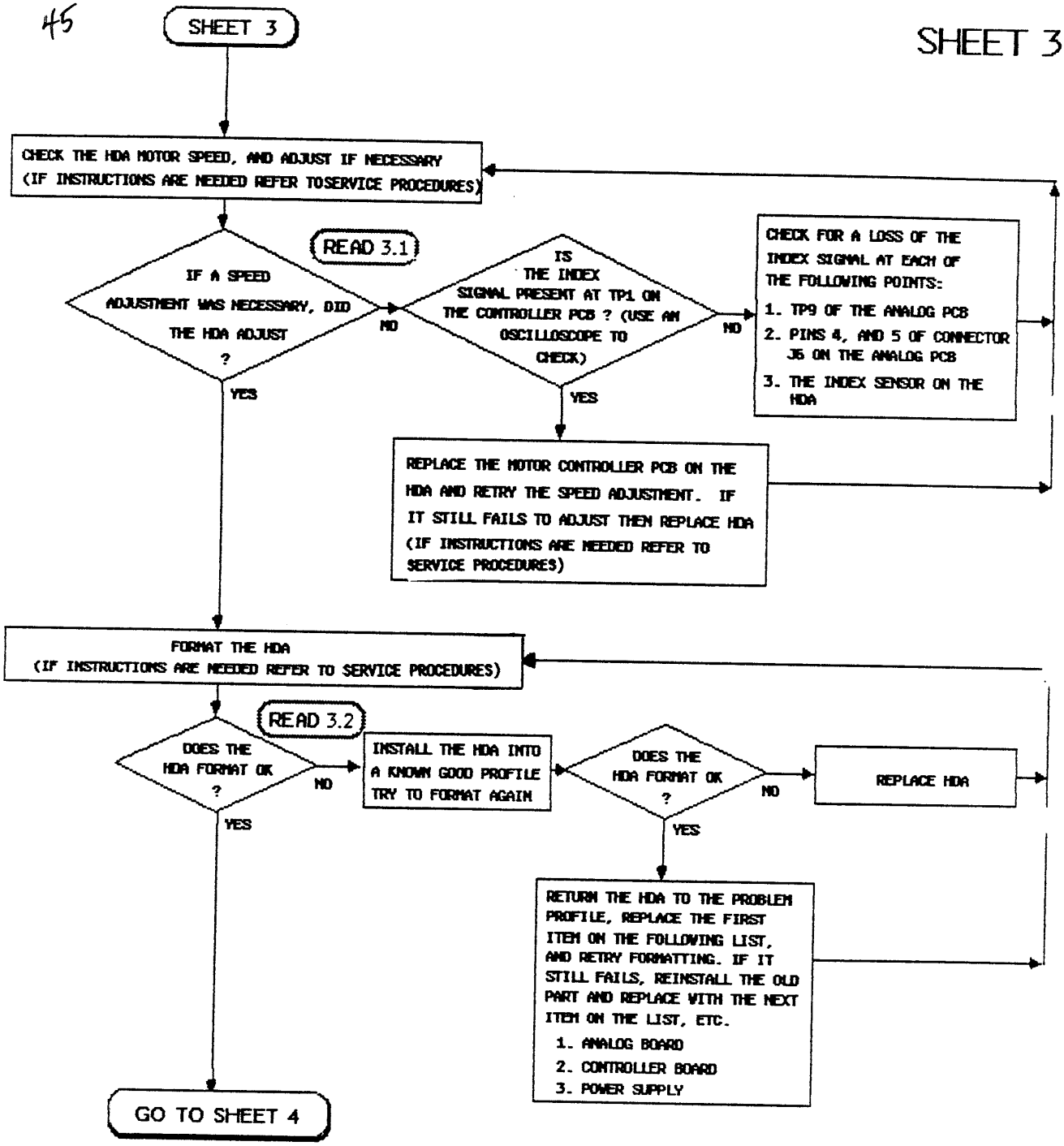
WARNING: At least one module must be connected to the Power Supply at all times to provide it with a load.

If the voltages are OK with the known good Controller PCB installed, then the problem must be with a module other than the power supply. To determine which of these modules is causing the power problem, each module is reconnected one at a time and P3 is checked after each reconnection. **(Now return to the flowchart.)**

45

SHEET 3

SHEET 3



46

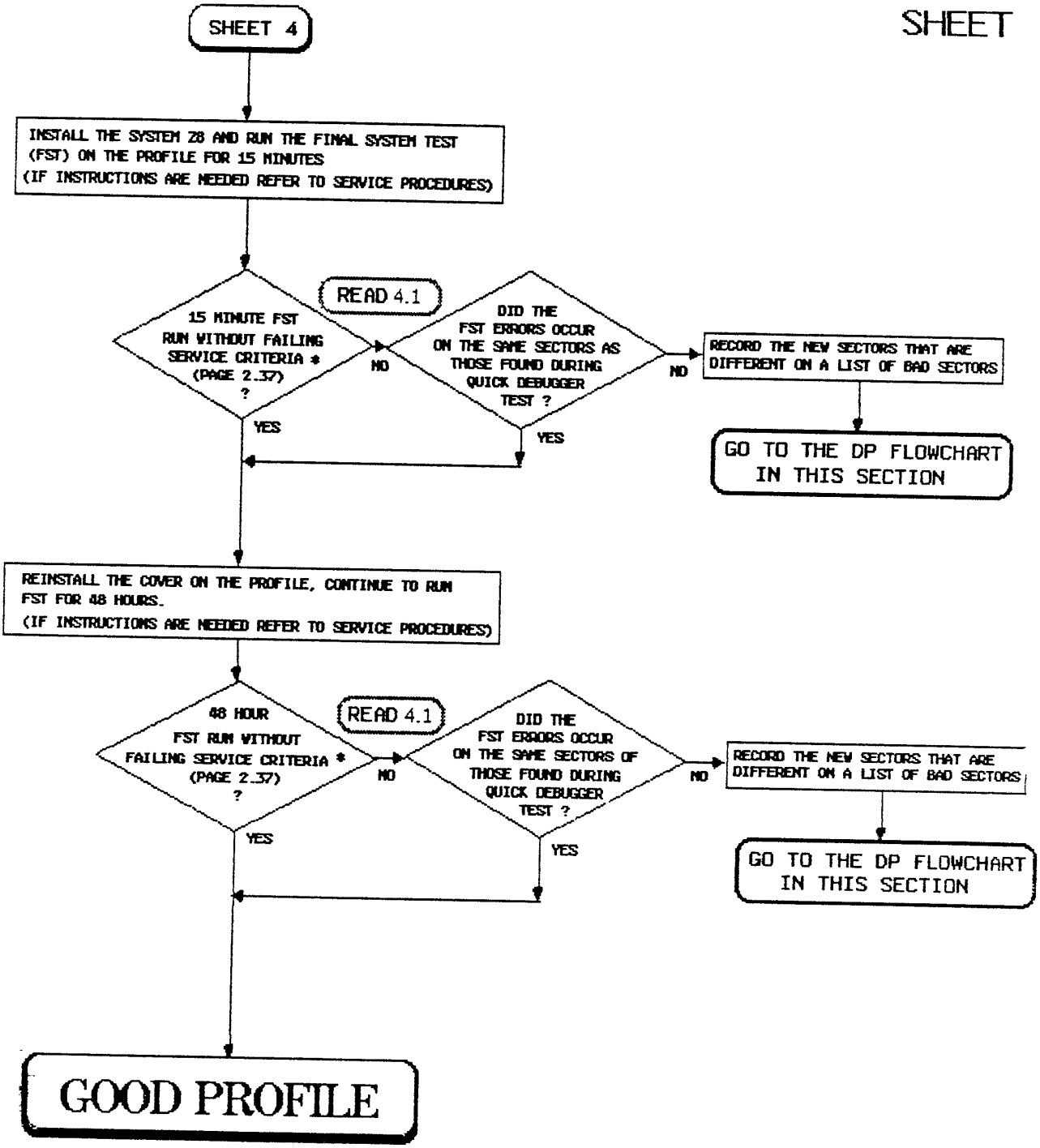
3.1 The Index signal is never actually used by the Pro-File electronics after the HDA has been formatted. During the format process, Index is used to provide a reference for the sectoring scheme to be put on the disk. (For an explanation of the Format used on the Pro-File refer to the Pro-File HDA Description in the Appendices section of this manual.)

But the Index does provide the technician with an indication of actual disk speed. It should occur once per revolution just before sector 0 on disk surface 0. Since the disk speed is supposed to be 3600 RPM this means that the frequency counter should measure a 16.666 ms period between each Index pulse.

If the Index signal is present at TP1 of the Controller PCB, but will not adjust as per directions in the Index Check and Adjustment procedure in the Service Procedures Section of this manual, then the problem could be with the electronics on the Motor Controller PCB on the HDA, or with the disk motor on the HDA. If the signal is absent at TP1 on the Controller PCB, then flow should be traced back to the Index sensor on the bottom of the HDA. **(Now return to the point in the flowchart which referred you to this discussion.)**

3.2 During the Format process all sector headers and marks are erased and the spares tracks (track 77 on all disk surfaces) are erased. An inability to format could be caused by an electronic problem in the Pro-File, or by the HDA. If the HDA formats OK after putting it into a known good Pro-File, then the problem is probably with one of the electronics modules in the Pro-File. **(Now return to the point in the flowchart which referred you to this discussion.)**

47



48

4.1 The FST (Final System Test) is an exersizer/tester program used to determine whether or not the Pro-File is operating within acceptable limits.

For approximately the first 10 minutes of the FST, the data block field of every sector is written to and read to detect any CRC read errors.

The system Z8 is used during this test, so the same firmware routines (i.e. Scan, Seek, Read, Retry, Diagnostic, etc.) will be used. After the initial 10 minute write/verify has been performed, the FST exercises the Profile by doing random Seeks, Reads, etc. until its operation is terminated when the operator presses the <ESCAPE> key.

The reason the flowchart has you check for problems after the first 15 minutes of FST is because most hard failures will probably produce errors within 5 minutes after the FST begins to exercise the Pro-File. At this point in the flowchart errors can be caused by 2 types of problems:

1. An **Electronic problem** which has caused a sector to be unreadable by altering the data bit pattern at its original sector location on the disk. For example the Z8 may malfunction as it performs a Seek, and instruct the electronics to write as the heads move across the media to a target track.

   This of course would destroy the parts of all the sectors that the head passes over on its way to the target track (the head should never write during a seek).

2. A **Media problem** which has made it impossible to record magnetic information on the sector where it occurs. This could be a manufacturing defect in the media (the magnetic coating on the disk), or a ding in the media caused by head contact during rough handling.

The big question at this point is which of the 2 types of problems described above is the real culprit. If the error was caused by an itermittent electronic problem producing a bad format on the HDA, then this looks like an HDA (media) problem. However, if the HDA is replaced the problem will probably reoccur in time. (This discussion is continued on the next page.)

49

4.1 (continued) The method this flowchart uses to determine which type of problem caused the error is to compare the sector locations of sectors that had errors before formatting (a list of these sectors was printed out when you ran the Quick Debugger program on the problem Pro-File), with those produced after formatting (these will be shown by the FST).

If they are the same, then the problem which caused the errors, is probably bad media because bad media locations don't change, so the HDA should be replaced.

If they are different, then the problem is probably electronic. This is because it is highly unlikely that an intermittent electronics problem would occur on the same sector twice.

(If you would like further information on spared sectors, firmware routines, the Pro-File's HDA format, etc., you may refer to the descriptions in the Appendices Section of this manual.)

If the faulty sectors were the same, and the number of errors exceeds those specified by the FST service criteria sheet (found in the Service Procedures Section), then replace the HDA and return to the point in the flowchart which referred you to this discussion.

If they are different, then the problem is probably electronic and you need to look at some of the different faulty sectors to determine what is bad about them and so determine which module is faulty. The DP (Diagnostic Procedure) flowchart that you are being referred to, will explain how you can continuously read a spared sector and interpret its analog signal. **(If there were different sectors, then proceed to the DP Flowchart.)**

50

THIS PAGE LEFT BLANK INTENTIONALLY

51

NO SCAN FLOWCHART
SHEET 1

WHEN THE PROFILE WAS TURNED ON DID THE READY LAMP COME ON FOR 0.5 TO 3 SECONDS AND THEN GO OFF?

NO

READ NS1.1

MEASURE THE POWER SUPPLY VOLTAGES AT P3 OF THE CONTROLLER PCB
(REFER TO FIGURE 3)

FIGURE 3

PIN #	VOLTAGES
1	+5V ± 0.1V
2	+12V ± 1.2V
3 POK	+5V ± 0.3V
4	-12V ± 1.2V
5	GROUND

P3

YES

ARE THE VOLTAGES WITHIN TOLERANCE ?

NO

WHILE LEAVING EACH MODULE IN PLACE, DISCONNECT CONNECTORS J4, J3, AND J2 (J4 CONNECTS THE HDA TO THE POWER SUPPLY, J3 CONNECTS THE CONTROLLER PCB TO THE POWER SUPPLY, AND J2 CONNECTS THE CONTROLLER PCB TO THE ANALOG PCB) THEN CONNECT A KNOWN GOOD CONTROLLER PCB INTO J3 AND RECHECK EACH VOLTAGE

YES

IS THE READY LAMP ALWAYS LIT?

YES

REPLACE THE FIRST ITEM ON THE FOLLOWING LIST. IF IT STILL FAILS, REINSTALL THE OLD PART AND REPLACE WITH THE NEXT ITEM

1. SYSTEM Z8
2. CONTROLLER PCB

NO

REPLACE THE FIRST ITEM ON THE FOLLOWING LIST. IF IT STILL FAILS, REINSTALL THE OLD PART AND REPLACE WITH THE NEXT ITEM

1. THE READY LED
2. CONTROLLER PCB

REPLACE THE POWER SUPPLY

NO

ARE THE VOLTAGES WITHIN TOLERANCE ?

YES

ONE AT A TIME, RECONNECT THE FOLLOWING:

IF THE PROBLEM REOCCURS AFTER THE RE-ATTACHMENT OF A MODULE THEN THAT IS THE SUSPECTED MODULE

1. J3 TO THE OLD CONTROLLER PCB
2. J2 TO THE OLD CONTROLLER PCB (POWER TO ANALOG PCB)
3. J4 TO THE HDA

IS THE HDA SPINNING ?

NO

READ NS1.2

IS THERE +12 VOLTS AT HDA CONNECTOR P4?

NO

REPLACE THE POWER SUPPLY

YES

YES

TURN OFF THE PROFILE REPLACE THE MOTOR CONTROLLER PCB, IF THAT DOESN'T WORK, THEN REPLACE THE HDA

GO TO SHEET 1 OF THE MAIN FLOWCHART PG 1.4

GO TO NO SCAN FLOWCHART SHEET 2

52

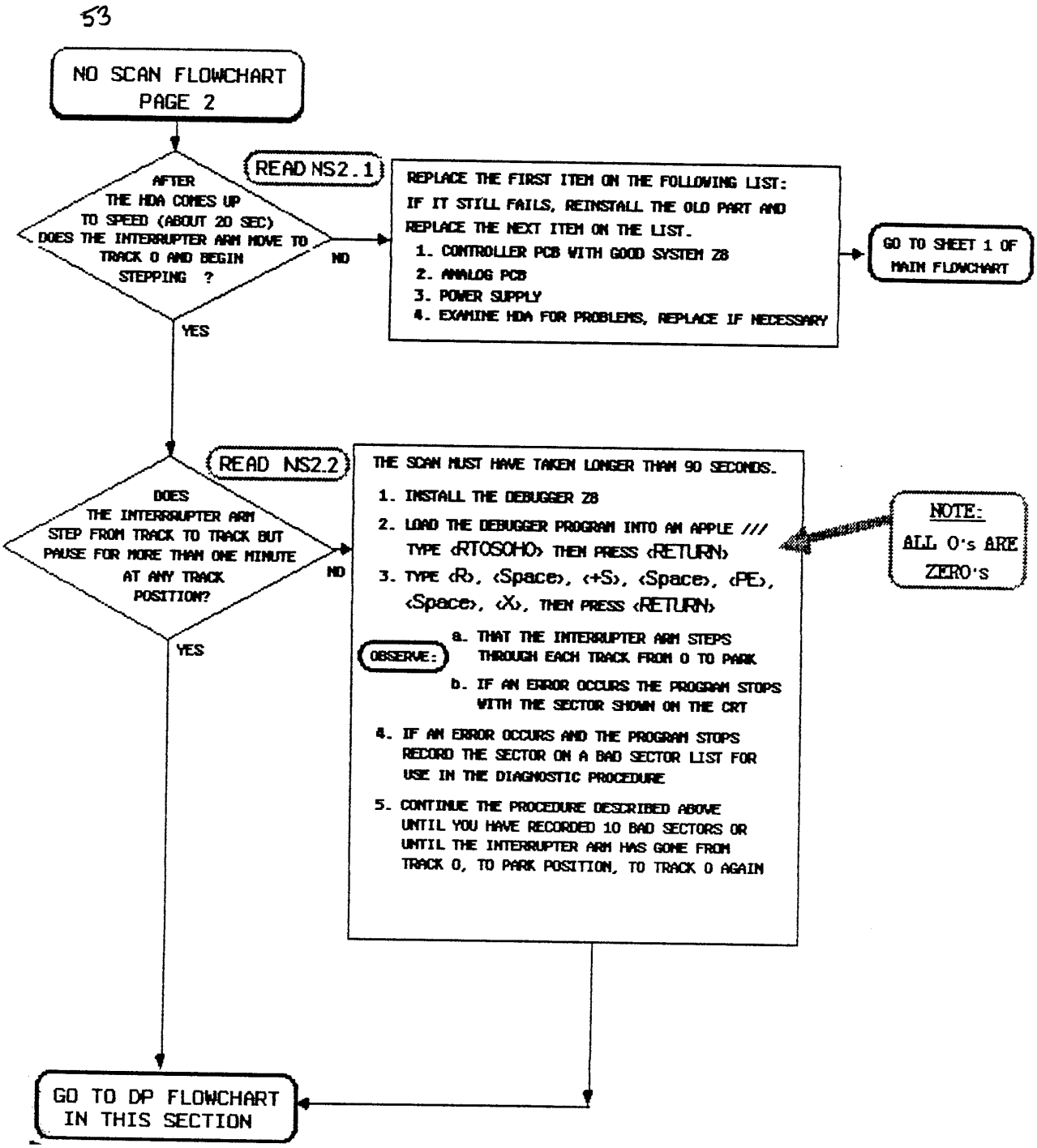
NS1.1 Upon power up on the Pro-File the Z8 will wait for the POK (Power OK) signal to occur. This signal is generated by the power supply when it determines that the proper voltage levels for each voltage have been reached and stabilized. When it happens the Z8 will cause the Ready lamp to extinguish. This should happen within 1/2 to 3 seconds of power up.

A missing or out of tolerance voltage can be caused by any of the modules in the Pro-File. The method used in this flowchart to isolate the problem module is to disconnect the power supply from all the modules, plug in a known good Controller PCB and monitor the voltages at P3 (the power supply connection to the Controller PCB). If the power supply is at fault then the voltage/s will still be missing or out of tolerance.

If the voltages are OK with the known good Controller PCB installed, then the problem must be with a module other than the power supply. To determine which of these modules is causing the power problem each module is reconnected one at a time and P3 is checked after each reconnection. **(Now return to the point in the flowchart which referred you to this discussion.)**

NS1.2 Approximately 20 seconds after the Ready lamp goes out indicating that the power has stabilized, the disk motor should be rotating at 3600 RPM (its operating speed). If the HDA isn't spinning the Pro-File can't read data and that's probably why the Scan operation was unsuccessful. **(Now return to the point in the flowchart which referred you to this discussion.)**

53



54

NS2.1 The firmware program in the system Z8 waits approximately 20 seconds after a successful powerup (a successful powerup is indicated by the Power OK signal becoming active and causing the Ready lamp to extinguish) before it initiates the Scan operation. During Scan operation the Z8 will cause the following things to occur.

Note: Turn off the Pro-File and gently position the interruptor arm to the Park position (if it is not already there). Then turn on the Pro-File while you observe the following.

The interruptor arm will move to approximately the center of its travel and pause there briefly while the spares table is read from track 77 on disk surfaces 2 or 3.

If there is a problem which disables the Pro-File from being able to read anything, then this is the first place that it would be evident.

If the Z8 is unable to read sector headers on track 77, then the interrupter arm may go to track 0 and back to 77 to try again. If it still can't read sector headers it may go to track 155 (Park position) and back to track 77 to try again to attempt read the spares table. If it still can't read the spares table the Z8 will abandon the Scan operation and move the interruptor arm to Park position.

(If you would like a detailed description of the Scan operation, or further information on the spares table, error recovery routines, the Pro-File's HDA format, etc. refer to the descriptions in the Appendices Section of this manual.)

(Now return to the point in the flowchart which referred you to this discussion.)

NS2.2 If the Seek to track 77 was successful (as described above in NS2.1),then the read circuitry must be at least marginally OK, so the Z8 will go to track 0 to begin to read each sector in sequence beginning with sector 0. During this time the Ready lamp will blink. If there is a problem in this procedure the Pro-File firmware may go to an error recovery routine.

(If you would like a detailed description of the Scan operation, or further information on the spares table, error recovery routines, the Pro-File's HDA format, etc. refer to the descriptions in the Appendices Section of this manual.)

(Now return to the point in the flowchart which referred you to this discussion.)

55

DP - DIAGNOSTIC PROCEDURE

NOTE: The Diagnostic Procedure (DP) flowchart begins on page 1.22. However if this is your first time through this procedure, it is strongly recomended that you read the following introduction.

Introduction:

The main objective of this procedure is to familiarize the technician with some common failure modes within the Pro-File electronics. Each of these modes usually shows up as an alteration, of the analog waveform of the sector being read from the HDA.

The modified patterns usually do not reflect actual physical damage to the disc surface, but rather a problem in the Pro-File electronics.

The technician's job will be to examine the faulty sector's analog signal as it is being read to determine which module within the Pro-File caused or is causing the failure.

The faulty sector's analog signal waveform can be monitored directly by placing oscilloscope probe(s) on either Test Points 1 & 2 (TP1 & TP2 are the same signal differentiated) on the Analog PCB and externally triggering the scope with the Index signal at TP9.

An easy way to access these test points is to remove the metal plate located underneath the Pro-File, and tilt the Pro-File up on its side to expose the Analog PCB (refer to page 2.22 for the location of the Test Points). This way the Pro-File can be checked quickly with a minimum of disassembly.

Following this introduction are examples of analog signals PCB taken on TP1 & TP2 along with the VCO charge pump data locking error voltage at TP8.

The signal read at either TP1 or TP2 is the analog data read directly from the head after passing the data through a preamplifier, lo-pass filter, and automatic gain control circuit.

The signals at TP1 and TP2 are really just head signals amplified by the circuitry on the Analog PCB. A loss of signal at these test points may indicate that there is a problem with the connection between the HDA and Analog PCB, or the Analog PCB circuitry.

(continued on the next page)

56

Introduction: (continued)

The signal obtained from TP8 is the feedback voltage that a Phase Locked Loop (PLL) control circuit uses to lock to the analog data signal on the HDA.

The feedback either increases or decreases the voltage to a **Voltage Controlled Oscillator** (VCO) causing it to increment or decrement its output frequency thereby syncing that frequency to that of the data being read.

When the Pro-File is idle, the VCO locks to about 20 Mhz, or twice the Controller PCB clock frequency (there is very little signal at TP8).

When reading, the frequency varies in proportion to the data frequency written to the disc.

So, whenever the Pro-File is reading the data from the HDA, the signal will change in amplitude to reflect data on the disc. (refer to page 1.26, Example of Normal VCO charge pump Signal {TP8 voltage})

Possible Problems:

A. Z8 microprocessor
Of the problems existing in the Pro-File, most of them can be attributed to the system Z8 microprocessor on the Controller PCB. (Refer to page 1.5, part 1.1).

The older system Z8 was a "piggyback" type chip, and the new version is the masked system Z8.

The masked version is a much improved version less likely to fail over a normal Pro-File lifetime, yet, there is small percentage of fallout within these chips too.

One malfunction that commonly occurs is when the Z8 turns on the write circuitry while seeking to a different track. (Refer to the following pages on Damaged or Cleared Data Areas)

(continued on the next page)

57

Introduction: (continued)

Possible Problems: (continued)

B. Controller and Analog PCB Upgrades

In the past some lines on the Controller and Analog PCBs were being spuriously activated. Sometimes this would cause the Z8 to go into its write routine when it was supposed to be seeking or idle.

The Controller and Analog PCB upgrade procedures (described in the Service Procedures Section), were designed to be a solution to these problems.

C. Controller PCB

The Pro-File Controller PCB functionally provides control signals to the Analog PCB and HDA to move the heads to the proper track, select the proper sector, channel data, and monitor error conditions during a read or write.

If a malfunction occurs in any of these functions, the Controller PCB can be suspected as the bad module. However, do not forget to verify the Z8 microprocessor first.

D. Analog PCB

There are 2 main Analog PCB problems:

1. PLL (Phase Lock Loop) problems may show up as an abnormal VCO (Voltage Controlled Oscillator) charge pump signal at TP8 when the analog data signal at TP2 and TP2 appears normal. In this case, the PLL circuitry would not be able to lock to the analog data signal coming from the HDA. (Refer to page 1.40, Example of Abnormal VCO charge pump).

2. AGC (Automatic Gain Control) circuitry problems may the analog data signal to be low or non-existant at TP1 and TP2. (Refer to page 1.42, Example of Abnormal AGC signal)

E. Hard Disk Assembly (HDA)

Of the many possible problems which can occur on the HDA, there are four major ones.

(continued on the next page)

58

Introduction: (continued)

Possible Problems:

E. Hard Disk Assembly (HDA) (continued)

1. **Media damage** (Refer to page 1.38, Example of Bad Media) can be found by finding an area on the media with a loss of signal (not a sector mark), where repeated attempts at rewriting data does not return the area to normal. (Rewriting the sector can be accomplished with the Big Debugger program, using the <W> command, refer to page 2.40 for instructions)

   There also can be damage to the media where one of the disc heads has struck the surface of the disc. Remember that there is a space for 32 bad sectors in the Spares Table, so a few bad sectors existing due to a media problem is acceptable and a normal part of Pro-File operation.

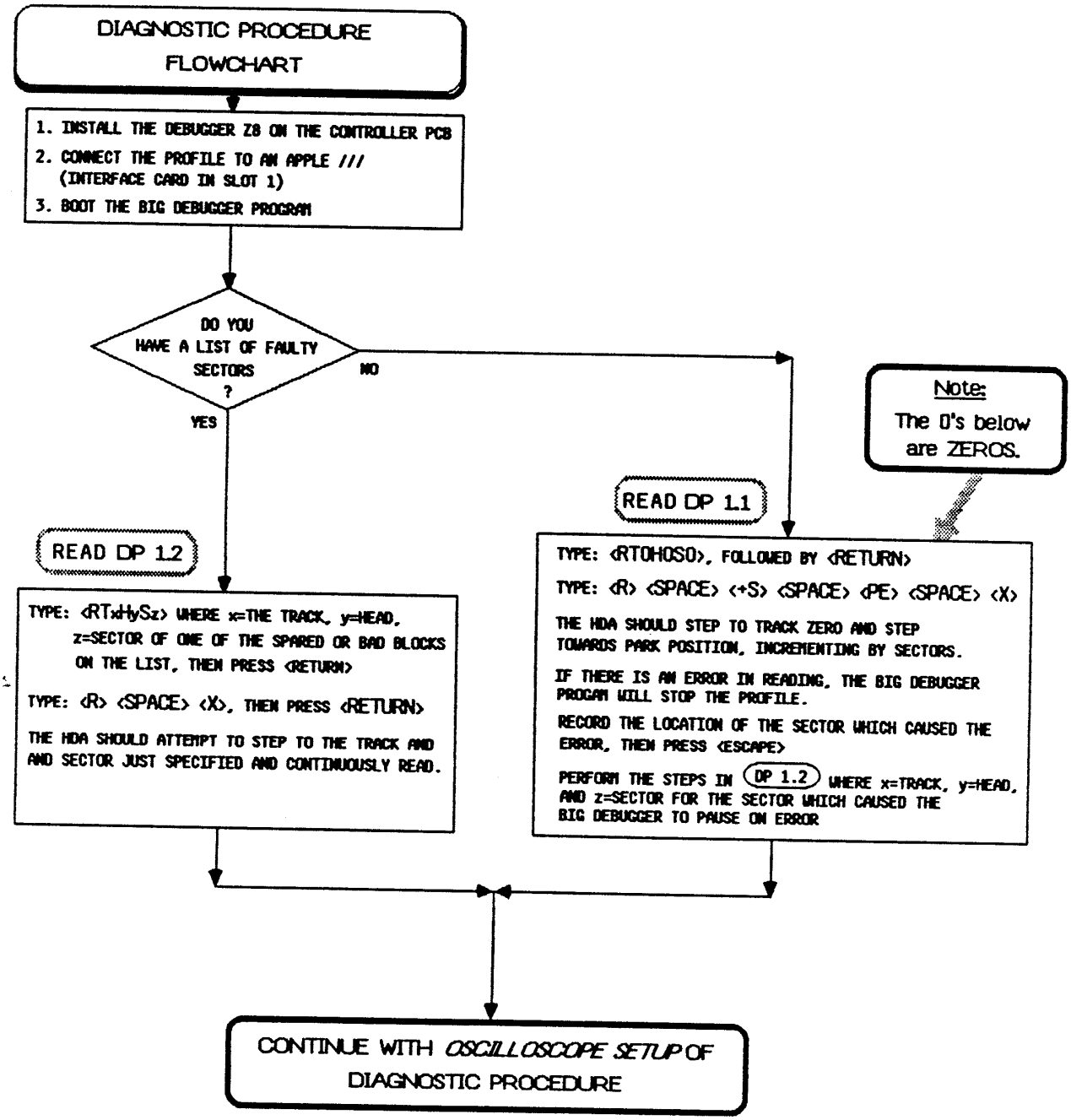
2. **Motor Speed problems** usually occur because of a defective Motor Control PCB on the HDA. A widely oscillating speed might adversely affect the operation of the Pro-File. The analog signals will be present at the Test Points, but the analog to digital circuitry will probably not be able to lock to the data on the HDA. And the Pro-File probably will not pass reformatting without errors occurring.

3. **Disc Head problems** are usually limited to a complete loss of signal at one or more of the heads. It is because of damage to the heads or a bad connection to the heads. Either way, the HDA needs replacement, as there is no way to repair a problem internal to the HDA in the field.

4. **Stepper Motor problems** may be caused by a bad connection between the Controller or Analog PCB's, or a bad stepper motor winding. Either way, the stepper will not move smoothly or have accuracy at any track step. To confirm the problem use the Read command from the Big Debugger program in the Service Procedures Section to perform a read on some track other than where the heads are now. If the interrupter arm rotates in the direction opposite of the target track, or shakes back and forth there may be a problem with the stepper motor.

   Another problem that can occur is stepper motor **hysteresis.** Hysteresis is where a magnetic field is built up inside the stepper motor coils causing the motor to settle with the head positioned at a slightly off track position. In this case the HDA should be replaced since this is a track alignment problem internal to the HDA and not repairable by the technician.

59



60

DP1.1 If you do not have a list of faulty sectors, then the interrupter arm must have spent 1 minute or more over a track during Scan. This probably means that the Pro-File had difficulty reading a sector or sectors on that track. By entering the commands specified at this step in the flowchart into the Big Debugger program, you will cause the Big Debugger program to read every sector in sequence and stop in the event of an error.

When the program stops the test operation, the heads will be positioned over the problem track, and the head for the disk surface containing the problem track will be enabled. This means that the analog signal from that track should continuously be sent to TP1 and 2 on the Analog PCB. Perform the steps in the DP1.3 procedure to view the signal from this track and determine if its display can provide you with a clue as to what the problem is.

If you don't feel that the display of this track is providing you with the syptoms you need to diagnose the problem, you can press the <**SPACE**> bar to cause the Big Debugger program to test the rest of the sectors until it finds another error. View that tracks signal for symptoms in the same way .... etc.

DP1.2 By entering the commands specified at this branch in the flowchart into the Big Debugger program you will cause the Big Debugger program to read one of the sectors that you determined earlier in the flowchart might contain clues to electronic problems in the Pro-File.

While the program continuously reads the specified sector, the heads will be positioned over the problem sector's track, and the head for the disk surface containing the problem sector will be enabled. In DP1.3 you will find a procedure to view the signal from this track.

If you position your timebase delay hilight over the analog data read signal for this period and enter timebase delay mode on your oscilloscope you will see an expanded display of the questionable sector. Compare the display of the faulty sector by itself with the faulty sector examples later in this procedure to help you find the problem.

If you don't feel that the display of this sector is providing you with any of the symptoms described in DP1.3, you can enter the commands necessary to have the Big Debugger program continuously read the next faulty sector on your list. View that sectors signal for symptoms in the same way .... etc.

61

DIAGNOSTIC PROCEDURE
OSCILLOSCOPE SETUP

READ DP 1.3

TO VIEW THE TRACK BEING READ:

1. TURN ON THE OSCILLOSCOPE AND ALLOW IT TO WARM UP FOR 15 MINUTES
2. *INITIALLY* SET CHANNEL A (OR 1) ON THE OSCILLOSCOPE TO THE FOLLOWING:
   A. VOLTS PER DIVISION CONTROL SETTING TO 1 VOLTS/DIV
   B. AC COUPLING
   C. CONNECT TO HDA ANALOG DATA - TP1 OR 2 ON ANALOG PCB (GROUND THE PROBE)
3. SET CHANNEL B (OR 2) TO:
   A. 0.5 VOLTS/DIV
   B. AC COUPLING
   C. CONNECT TO HDA VCO SIGNAL - TP8 ON ANALOG PCB (DO NOT GROUND THE PROBE)
4. USE EXTERNAL TRIGGER AND SET TIMEBASE TO:
   A. 2 mSEC/DIV
   B. CONNECT EXTERNAL TRIGGER TO INDEX PULSE - TP1 ON CONTROLLER PCB
   C. AUTO TRIGGERING - ADJUST TRIGGER LEVEL TO LOCK SIGNAL ON SCOPE

NOTE: IF THERE IS NO INDEX PULSE ON TP1 (CONTROLLER PCB), TRY USING TP9 (ON ANALOG PCB). IT IS ACTUALLY THE INDEX SIGNAL FROM THE HDA, IF IS NOT PRESENT, CHECK THE INDEX SENSOR ADJUSTMENT DESCRIBED IN SECTION 2

OBSERVE: THE SIGNAL YOU ARE VIEWING SHOULD BE SIMILAR TO THAT SHOWN IN THE PHOTOGRAPH ON THE OPPOSITE PAGE.

TO VIEW THE SECTOR BEING READ:

5. WHILE IN THE MAIN TIMEBASE, POSITION AND ADJUST THE DELAYED TIMEBASE HIGHLIGHT OVER THE PERIOD OF THE VCO CHARGE PUMP SIGNAL THAT IS ACTIVE (PROBABLY LOW AND OSCILLATING, SIMILAR TO THE LOWER SIGNAL IN THE PHOTOGRAPH ON THE OPPOSITE PAGE).
6. ENTER THE DELAYED TIME BASE TO VIEW THE QUESTIONABLE SECTOR

OBSERVE: THE SIGNAL YOU ARE VIEWING MAY BE COMPARED WITH THE NORMAL SIGNAL ILLUSTRATION SHOWN ON PAGE 1.28.

7. VIEW THE EXAMPLE SIGNALS IN THE FOLLOWING PAGES TO DETERMINE A PROBLEM WITHIN THE PROFILE

CONTINUE WITH DIAGNOSTIC PROCEDURE
SIGNAL EXAMPLES

62

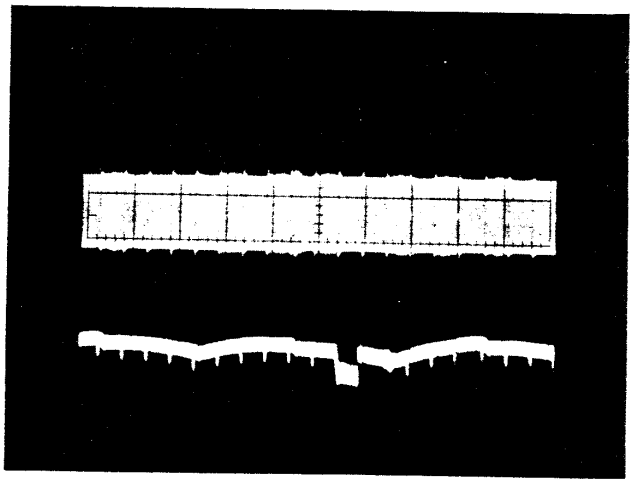
DP1.3 At this point in the flowchart a head should be enabled over a track containing a faulty sector. On your oscilloscope display you should see a signal similar to that in the figure shown below.

If you had a list of faulty sectors, then the Big Debugger program should be continuously reading a faulty sector. Notice the period in the VCO charge pump signal where it is low and oscillating. The sector where the VCO charge pump signal goes low is the sector that is being read.

If you position your delay trigger hilight over the analog data read signal for this peri,od and enter delay trigger mode on your oscilloscope you will see an expanded display of the questionable sector. Compare the display of the faulty sector by itself with the faulty sector examples later in this procedure. If you find one that is similar, read the corresponding explanation and if you wish perform the recommended module replacement.

If you are viewing a questionable track and you don't know which sector is bad, then use the delay trigger function on your oscilloscope to view the sectors individually to determine why the Pro-File had a problem with this track.

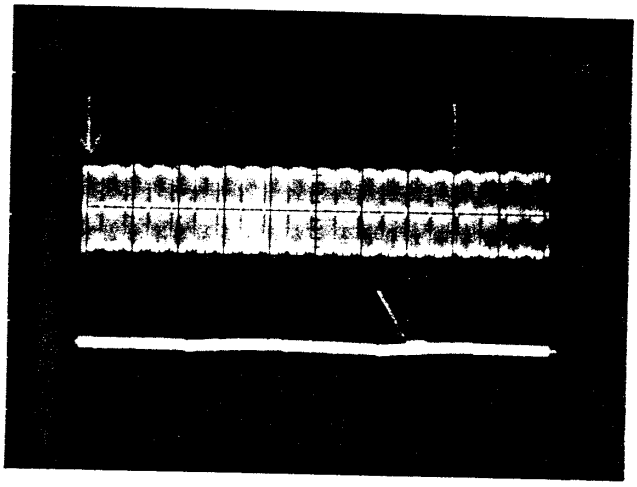
Note: A sector mark (sometimes called sector gap) separates each sector as shown below. If you would like information on the format used for the Pro-File refer to the Pro-File HDA description in section 3.



63

Diagnostic Procedure

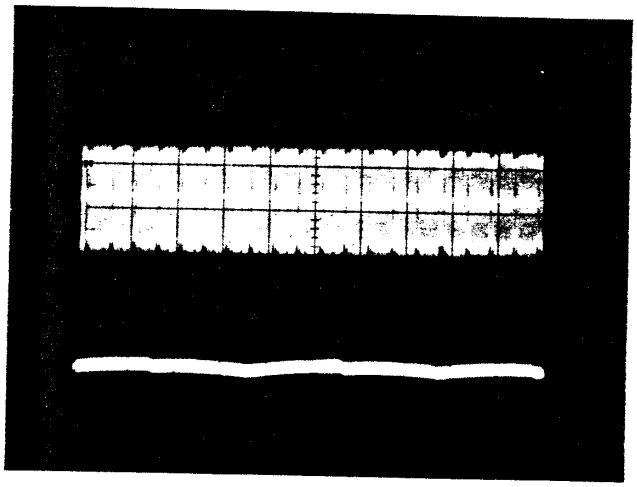
Example of a Normal Track



TP1 displays a view of a track of data. An entire track is shown between the arrows. Amplitude is 2 v peak to peak.

TP8 (VCO) is shown locking in to read a sector by the lower arrow.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays another look at a normal track with a higher amplitude of 2.4 v. Note that there are 16 areas of no signal corresponding to the sector marks.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div

64

Examples of a Normal Track

The amplitude voltage of an analog data signal from a normal track can vary depending on the physical location of the track, the Analog PCB, and what actual data has been written to the HDA.

The photos shown at the left are two characteristic normal tracks which show a **possible** variation in amplitude voltage. The upper photo displays an amplitude of about 2 volts peak to peak, the lower amplitude 2.4 volts.

Each track contains 16 sectors separated by a sector mark, but at this timebase the photo actually shows more than the 16 sectors.

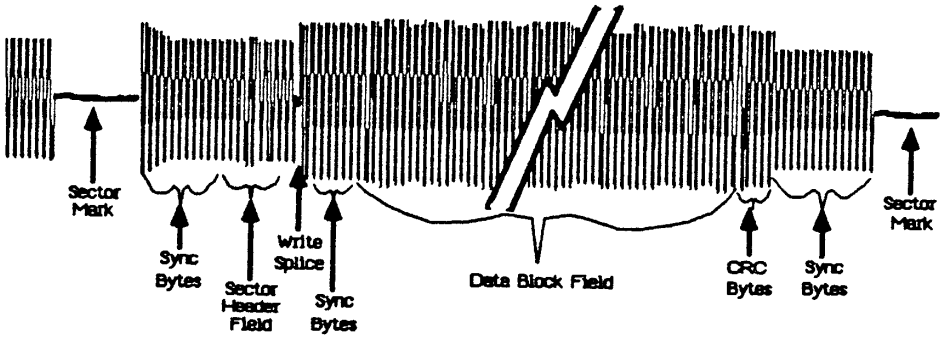
This is because the period of signal displayed at this timebase is longer than the period of time that it takes to read one complete track, so the beginning of the same track is displayed as the disk begins its next rotation.

One complete track is shown between the arrows.

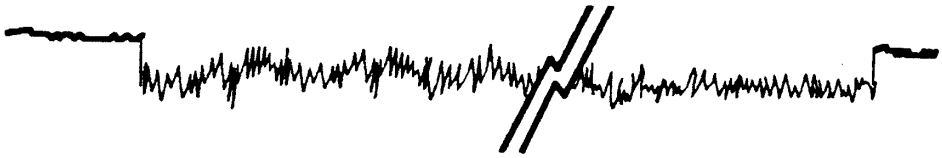
Note: Careful examination of the upper photo shows the VCO charge pump circuitry locking in on the sector indicated by the lower arrow.

65

EXPANDED VIEW OF ONE SECTOR (TP 1 or 2)



EXPANDED VIEW OF VCO CHARGE PUMP SIGNAL (TP8)



THE OSCILLATIONS SHOULD OCCUR ONLY FOR THE SECTOR BEING READ. THE VOLTAGE LEVEL THEY OCCUR AT DEPENDS ON THE SPEED ADJUSTMENT ON THE HDA MOTOR CONTROL PCB (THE ILLUSTRAION DEPICTS A SIGNAL RIDING ON A NEGATIVE LEVEL, AS MOST WILL)

66

Examples of a Normal Sector

As explained in the previous example of a normal track, the amplitude voltage of an analog data signal from a normal track can vary depending on the physical location of the track, the Analog PCB, and what actual data has been written to the HDA.

The illustration at the left shows a typical view of a normal sector's analog signal. The amplitude of a sector can vary from about 1.1 volts to over 2 volts.

The flared "bell" shape at the beginning of each sector, is a normal characteristic caused by the AGC (Automatic Gain Control) circuitry as it adjusts for the gain of the analog read signal from the head. If the bell shape is missing, the technician might suspect a faulty Analog PCB.

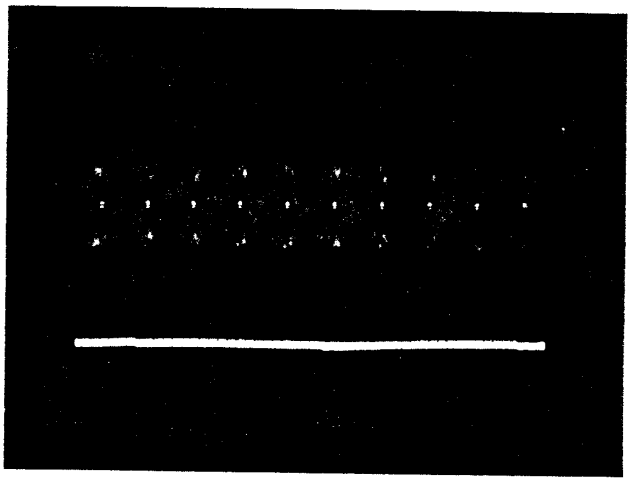
The Sync bytes and Sector Header field at the beginning of the sector are written during formatting, and should never be written over after formatting. The Write Splice is the point at which the Pro-File will normally begin to write a data block into the sector.

When the Pro-File reads a data block from a sector the data block field may not be in exact sync with the Sector Header field, so additional sync bytes are needed after the Write Splice, for the Pro-File to sync to the data field.

Notice that the VCO charge pump signal at TP8 goes low only for the sector being read. The oscillations that occur while the charge pump signal is low reflect the changes the VCO (Voltage Controlled Oscillator) makes to sync with the clock in the analog data signal read from the HDA (TP 1 or 2).

67

Diagnostic Procedure

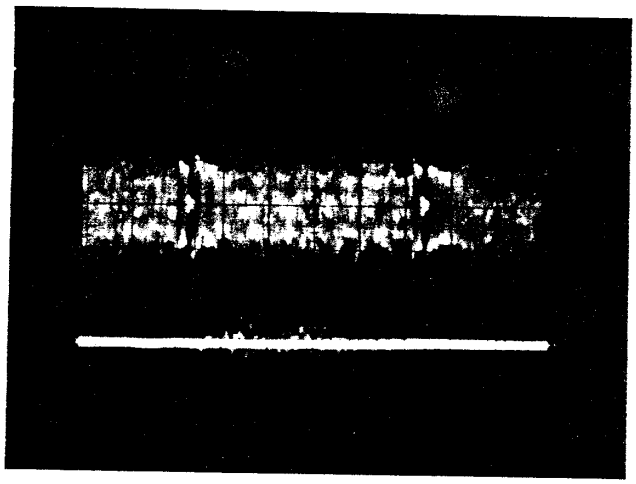


Example of a Normal VCO Signal

TP1 displays a partial view of a track of data. The Profile has been told to read the sector of data shown by arrow.

TP8 (VCO) shows a signal corresponding to the sector read. This signal "locks" to the data.

Scope Settings:
TP1 - 1 volt/div
TP8 - .2 volt/div
Timebase: 1 msec/div



TP1 displays a larger view of the sector being read.

TP8 (VCO) shows a more detailed look at the signal produced at TP8. Each increase and decrease in signal amplitude corresponds to the data signal at TP1.

Scope Settings:
TP1 - 1 volt/div
TP8 - .2 volt/div
Timebase: .2 msec/div

68

Example of a Normal VCO charge pump Signal

As explained in this flowchart's introduction, the VCO charge pump signal at TP8 enables the VCO (Voltage Controlled Oscillator) on the Analog PCB to synchronize to the data written on the disk.

The voltage level at TP8 causes the VCO to increase or decrease its output frequency.

When the Z8 wishes to read a sector on a track this signal will go low and then oscillate. This means it is now controlling the correction of the VCO as needed during the reading of the sector.

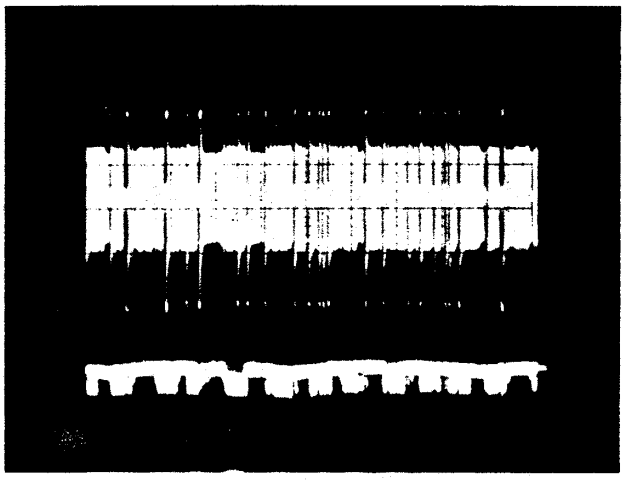
After the sector is read, the Z8 will direct the signal to go high again since there is no longer any data to be synced to.

The upper photo shown at the left shows a view of about half of a normal track. The Pro-File was told (using the Big Debugger Program) to read the sector continuously. No read errors occurred despite the large amplitude signal seen on TP8.

The lower photo displays an enlarged view of the sector being read above. Note that the changes in peaks in the data signal at TP1 correspond to the peaks in the VCO charge pump signal in TP8. An even larger view of the signals would show this more clearly.

69

Diagnostic Procedure

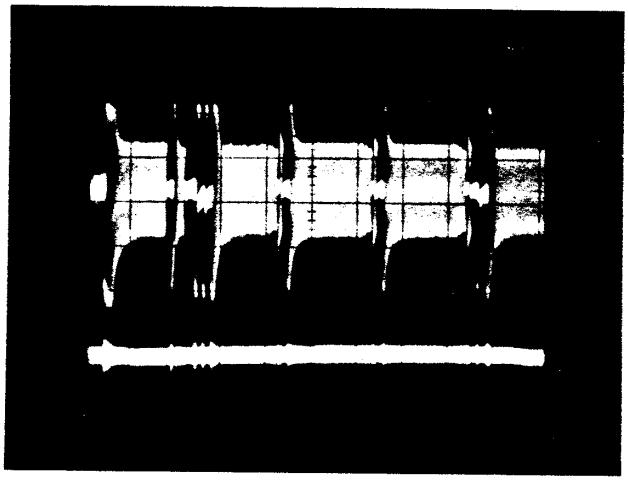


Example of a Damaged Track

TP1 displays a view of a track of data. There appears to be extra sectors if you count the extra peaks.

TP8 (VCO) attempted to lock to the data present on the track, but was unable to confirm any good sectors.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays a closer view of some of the sectors of the track. Several sectors appear to be normal, but have actually been damaged.

TP8 (VCO) could not lock to the damaged the sectors.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .5 msec/div

Suspect:
1. System Z8
2. Controller Board
3. HDA

70

Example of Damaged Track

As described in the Pro-File HDA Format description in section 3, sector marks divide each track into 16 separate sectors. The Controller PCB expects the sectors to be in a certain location with respect to the sector marks.

The upper photo shown at the left shows an entire track of data with uneven length sectors (they should all be the same length), and extra pulses (TP1). A normal sector would occur where the VCO charge pump (at TP8) shows the amplitude in the negative direction (down).

The Pro-File was instructed to read all the sectors of the track with the Big Debugger program, what is shown is every other sector being read. The lower photo shows an enlarged view of the track analog signal displaying areas which appear to be similar to a normal sector, but are actually damaged.

There are extra areas of no signal similar to a normal sector mark at the end of some of the sectors, and areas with multiple sector marks at the beginning of the sector (refer to lower arrow).

At this point, the technician would normally suspect the system Z8 as the cause of the damaged tracks. This is because the Z8 has historically been the most common failed component in the Pro-File. Yet, this Pro-File already had a masked version of the system Z8 installed on the Controller PCB.

So it is a possibility that the Controller PCB is really at fault, not the Z8. Since the Z8 had already been replaced during the upgrade procedures, it was decided to reformat the HDA (correcting the track damage in the format), and run FST for 15 minutes to see if the track damage would reoccur. A replacement system Z8 was installed, the system upgraded, and the HDA reformatted.

After running FST for 10 minutes, the Pro-File began to produce errors: (Refer to page 2.37 for a list of errors and criteria for failure)

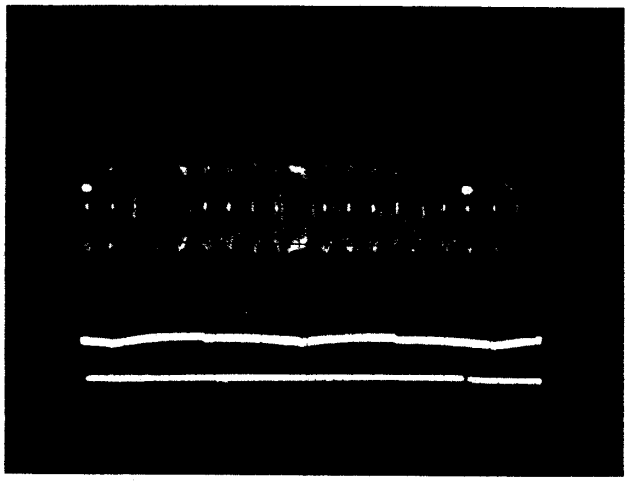
```
04 00 00 00
05 00 00 00
15 80 00 00
```

The Pro-File was stopped and the Controller PCB replaced. (The original Z8 was reinstalled into the new Controller PCB) and the 15 minute FST rerun. No additional errors occurred even after 48 hours, so it was assumed that the Controller PCB somehow produced the damaged tracks.

71

Diagnostic Procedure

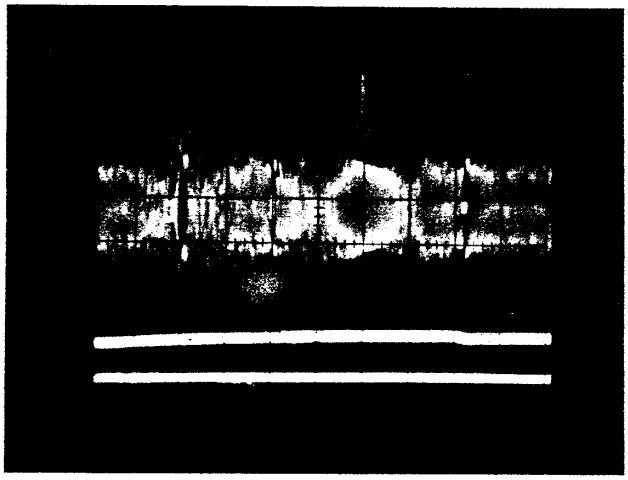
Example of Damaged Data Areas on Track



TP1 displays a view of a track of data. There are missing sector marks and other strange areas on the track.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays a closer look at one of the areas of the damaged track. The arrow points to where a sector mark is missing.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .5 msec/div

Suspect:
1. System Z8
2. Controller PCB
3. HDA

72

Example of Damaged Areas within a Track

As described in the Pro-File HDA Format, page 3.11, sector marks divide each track into 16 separate sectors. The Controller PCB expects the sectors to be in a certain location with respect to the sector marks.

The upper photo shown at the left shows an entire track of data between the two arrows with missing sector marks and other abnormal areas (TP1).

The damaged track was quickly found by using the Big Debugger program to scan from track 0 to park position, (type the command <R +T PE X>, refer to the Big Debugger description in section 2 if more instruction is needed).

The Debugger program stopped the heads at this track when it detected a read error. By looking at the signal at TP1, some abnormalities were seen.

The lower photo shows a enlarged view of a portion of the track in the upper photo. The arrow points to where a sector mark has been written over.

This kind of signal pattern is another probable characteristic of a system 28 failure. The sector mark was written over and the data put down in such a way so that it could not be read again.

Once the sector mark information is damaged in this way, there is no possible way for the Pro-File to rewrite the sector mark without reformatting.

The diagnostics internal to the Pro-File would probably spare this area under normal circumstances. The HDA must be reformatted to establish the normal sector information (however, this will destroy any data).

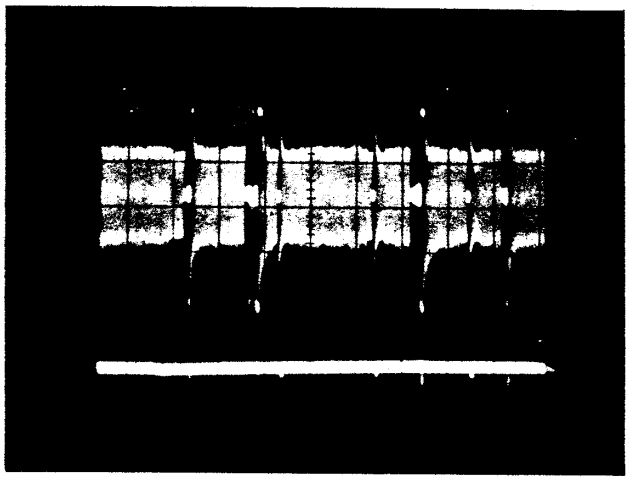
During a scan after powering on, the interrupter stopped momentarily at this track location while the 28 made attempts to read the damaged sectors.

The sectors on this track were not present in the spares table bad sector list after running the Quick Debugger program.

In a case such as this the only way to continuously find and read the faulty track is to run the Big Debugger program and have it pause on any error. At that point, look at the signal and try to determine the problem.

73

Diagnostic Procedure

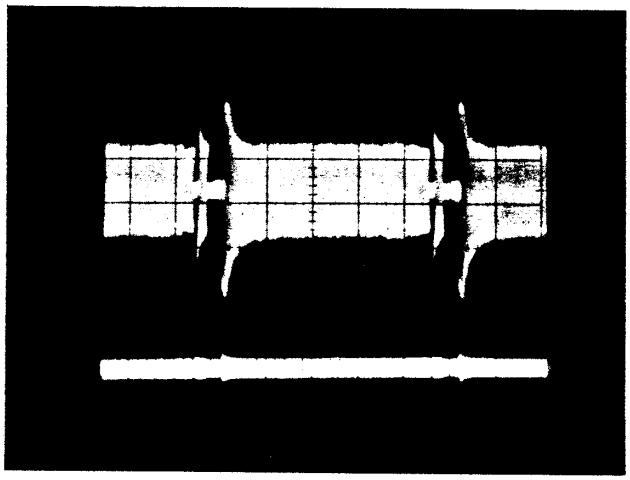


Example of a Damaged Track and Sectors

TP1 displays a partial view of a track of data. There is one normal sector of data shown by the arrow. The other sectors have been overwritten with an abnormal pulse in various places.

TP8 (VCO) could not lock to the abnormal sectors.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .5 msec/div



TP1 displays a larger view of the damaged sector.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .2 msec/div

Suspect:
1. System Z8
2. Controller PCB

74

Example of Damaged Track and Sectors

The upper photo shown at the left shows a partial track of data with one normal sector occurring in the center of the photo.

The other sectors have what appears to be an extra sector mark written in the middle. Not only is the extra sector mark in the wrong position, the width of the sector mark is about three times normal.

The Z8 became confused when reading this sector because of the extra sector marks.

The lower photo shows another track on the HDA with a similar problem (an extra apparent sector mark).

At this point, one should normally suspect the system Z8 as the cause of the damaged sectors.

The problem Pro-File originally had a piggyback system Z8 installed, so there was a very good chance that the Z8 would be the problem.

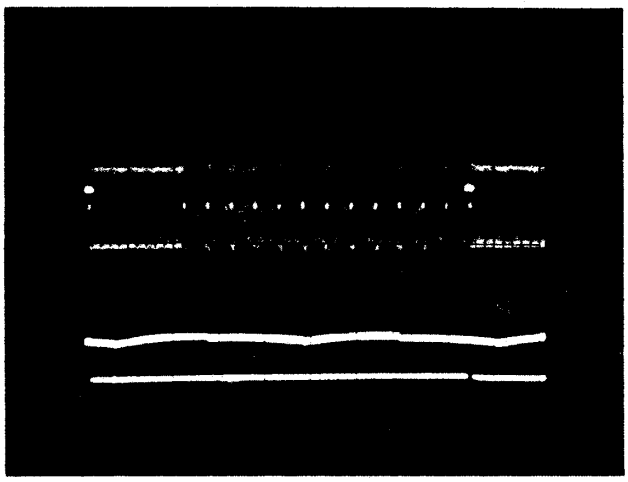
The Pro-File PCBs were upgraded and the HDA reformatted normally. A 15 minute FST run produced no errors, so the test was continued for 48 hours. No errors occurred.

(except an occasional read error - 08 00 00 00; refer to page 2.37 for a list of errors and how many of a type are acceptable)

75

Diagnostic Procedure

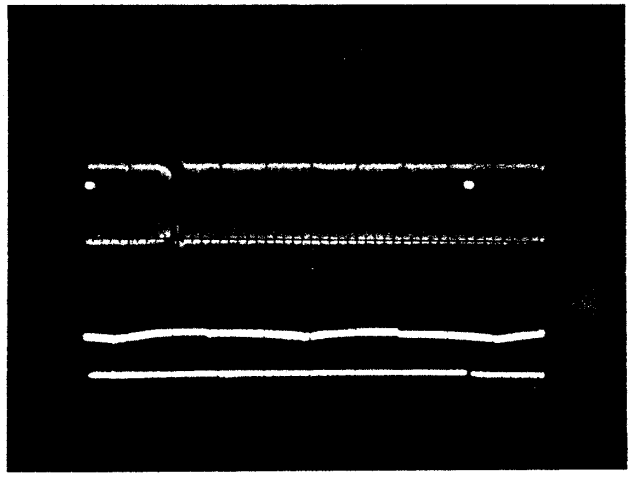
Example of Cleared Data Areas on Track



TP1 displays a view of a track of data. About a 1/4 of the track has been wiped to a constant AC signal. There are no signs of data in this area.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays another track which has been completely cleared of any data accept for one small disturbance. An normal track would be between the arrows.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div

Suspect:
1. System Z8
2. Controller PCB
3. Analog PCB
4. HDA

76

Example of Cleared Data Areas on Track

The upper photo shown at the left shows an entire track of data between the two arrows with about 1/4 of the track damaged.

The sectors in the damaged area have been cleared to a continuous level AC signal by some unknown source, probably the system Z8. Both the sector marks and data areas are missing.

The lower photo shows another track on the same HDA which has been cleared completely of any data and sector marks.

Each signal displays a condition where the HDA wrote a continuous AC signal without using the sector marks as an indicator of where to write.

There is no way for the Pro-File to repair the damaged areas without the technician reformatting the HDA.

Chances are very good that the system Z8 functioned abnormally and caused the Pro-File to write over the data signal; so this would be a good choice for the bad module.

It is also possible that the Controller or Analog PCB could be the cause of the problem because there could have been a short which may have caused the write circuit to malfunction.

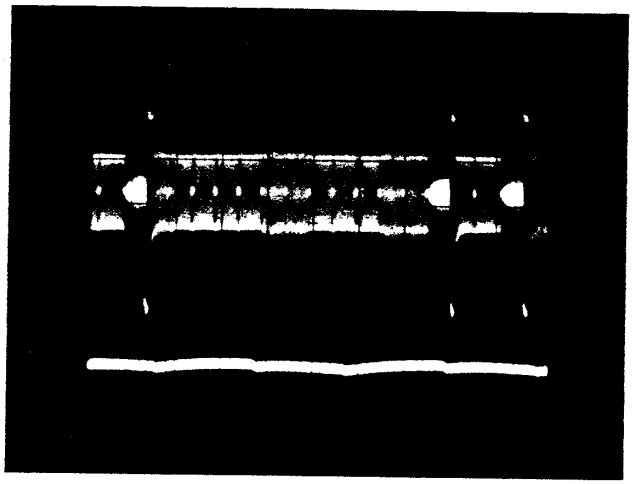
For example, a DC voltage shorted to one of the disc heads on the HDA, while the heads were positioned over a track might cause the head to perform a DC erase of the entire track.

This kind of erase would only take about 17 milliseconds to happen (one disc revolution). But if this happened, the signal would be a low amplitude signal, much like a continuous sector mark, etc.

77

Diagnostic Procedure

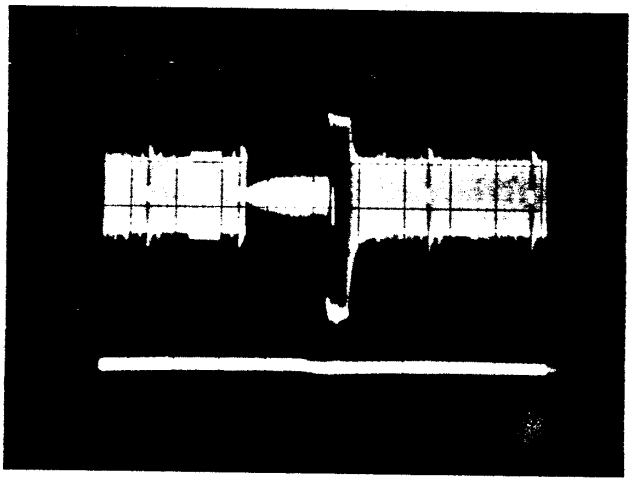
Example of Offtrack Signal



TP1 displays a view of a track of data. There are 3 sectors which have been damaged.

TP8 (VCO) was not able to lock on to this kind of abnormal data signal at TP1

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays a closer view of some of the sectors of the track, including the bad area. The signal shows that two sectors have actually been damaged.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .5 msec/div

Suspect:
1. System Z8
2. Controller PCB
3. HDA

78

Example of Offtrack Signal

As described in the physical description of the Pro-File HDA format in section 3 of this manual, the data areas are written on the disc surfaces in tracks, and the Z8 performs the Seek routine to cause the stepper motor to move the heads over the target track to access data. The number of step pulses the Z8 sends to the stepper motor determines how far the head will move and so which track it will be over. (For an explanation of the Seek routine refer to the Firmware Routines description in section 3.) Because the control system is of open loop nature, (i.e. there is little feedback telling the Controller PCB where the disc heads are physically located) both mechanical and electronic offtrack problems can occur.

A mechanical offtrack problem within the HDA can be caused by several things. The stepper motor controlling the position of the heads can become inaccurate placing the heads slightly offtrack either way. The metal band which pulls the heads back and forth can become torn or damaged resulting in poor placement of the heads. The mechanical assembly holding the bearings for the movement of the heads can become distorted causing every track to be shifted one way. Or the HDA stepper motor could have a hysteresis problem. Any of these problems can only be corrected by replacing the HDA.

An electronic problem can resemble that of a mechanical one, except that there is generally no permanent damage done to the HDA. After reformatting, the HDA can be returned to normal service, as long as the problem has been eliminated from the system. If the system Z8 was an older revision piggyback type, then it is probably the bad component. This is because the Z8 has historically been the most common failed component in the Pro-File.

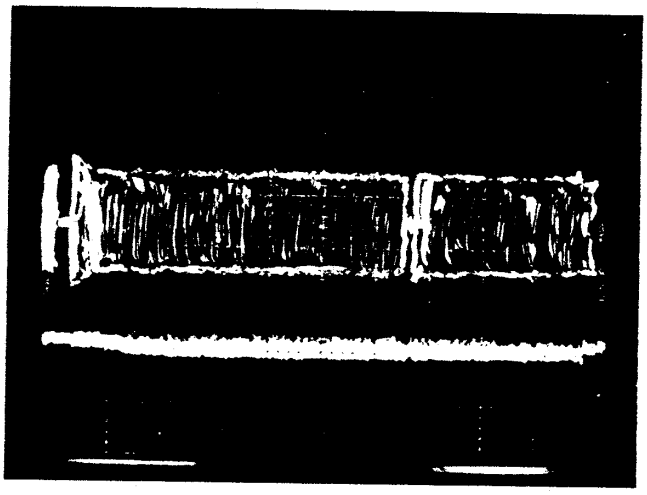
Another electronic failure that can cause an offtrack problem is the Controller PCB.

The upper photo shows three sectors which have been damaged by the system Z8 microprocessor by writing the data in an offtrack position. The lower photo shows a closeup of the analog signal where two sectors have actually been damaged. By telling the system to attempt to read the sector continuously, and carefully and gently moving the interrupter arm towards the target track, the signal became normal. This confirmed the fact that the sectors were written offtrack. By replacing the piggyback system Z8 with a masked ROM version, reformatting the HDA the problem disappeared. This was confirmed by running the Pro-File on FST for 48 hours.

79

Diagnostic Procedure

Example of Bad Media



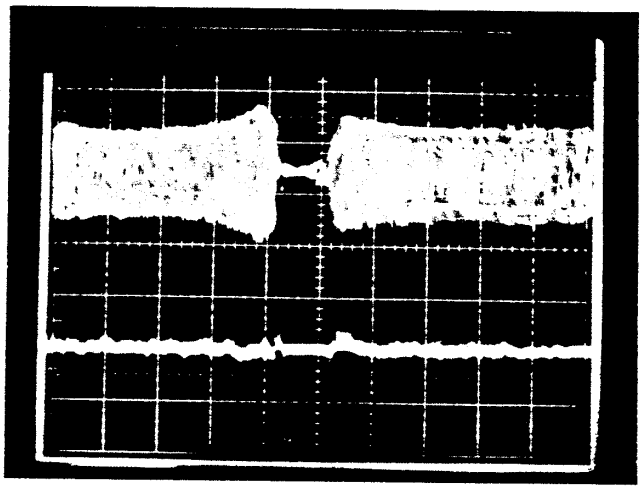
TP1

TP1 displays a closer view of a suspected bad area of a track. In about the middle of the sector there appears to be a section missing some magnetic material.

TP8

TP8 (VCO) does not show any particular abnormality.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .1 msec/div



TP1

TP1 -displays a larger view of the sector shown above. On closer examination, the dead area appears as a sector mark, except much smaller in width.

TP8

TP8 (VCO) displays as flat signal, as it would between normal sector marks

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .01 msec/div

Suspect:
1. Media Problem

80

Example of Bad Media

Bad Media is an example of where the HDA disk surface coating has been damaged by either a defect in the magnetic coating or a scratch due to head contact with the media. There are a limited number of allowable defects per disk surface. These defects (hard errors) are defined as ≤ 2 bytes in length. The total number of defects is:

Per Drive	32 errors total
Per Surface	8 errors total
Track 0	No errors allowed

The HDA disk drive uses two double sided 5-1/4 inch discs coated with an iron oxide base material as the recording media. The disc dimensions are 40mm inside diameter and 130 mm outside diameter. The thickness of the magnetic coating increases linearly from 20 microinches to 40 microinches at the outside diameter. The disk surface is coated with a Teflon lubricant 40 to 60 angstroms in thickness. (an angstrom = 0.0000000004 inches)

The media defect shown at left was detected by looking at the spares list of supposedly bad sectors, created by the Quick Debugger program and then telling the Big Debugger program to continuously read the suspect sector.

An initial view of the track looked completely normal, except for the fact that one of the sectors produced an error when reading.

By displaying the whole sector, there appeared to be a small area of no signal present (Refer to upper photo). Closer examination (by using the delay trigger function of the oscilloscope) of the suspect area of the sector, shows a total loss of signal in this area (lower photo).

If bad media is suspected an attempt should be made to write data to the sector(s) in question before the conclusion can be made that the suspect area is actually a media problem.

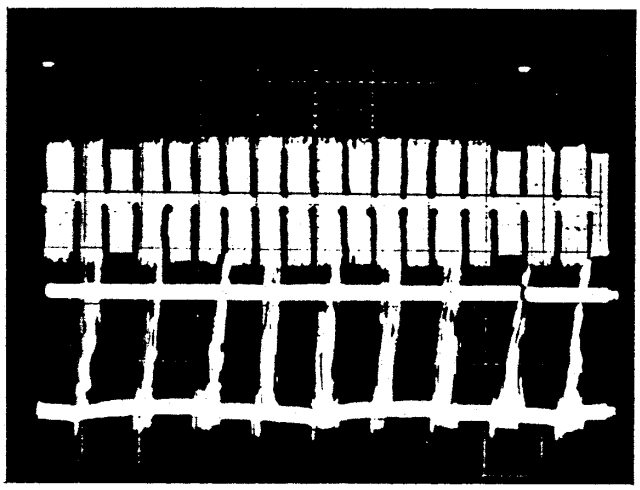
The Big Debugger allows the technician to read a sector into the computer buffer and write the sector back out to the 'bad' sector on the Pro-File. (Refer to page 2.40, for instructions on the Big Debugger program. Use the command <W> to write.)

If the number of total errors incurred by reading the media is greater than or equal to the total number of errors shown above, then the HDA should be rejected as defective.

81

Diagnostic Procedure

Example of Abnormal VCO Signal at TP8



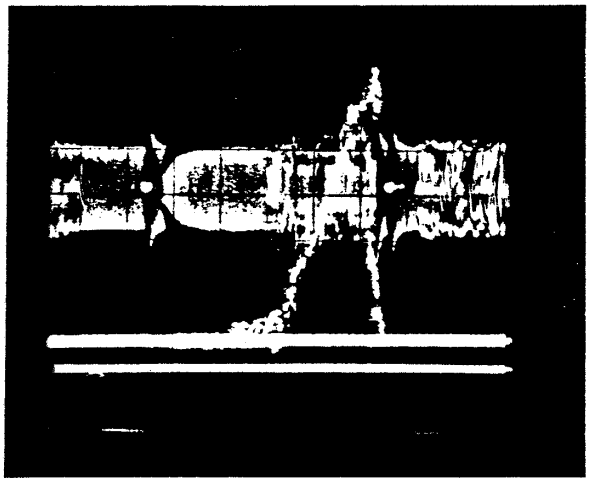
TP1 displays a view of a track of data. (The entire track appears between the two arrows)

TP1

TP8 (VCO) displays a very high amplitude signal (2v). This abnormal voltage is due to the errors occurring when reading.

TP8

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays a closer view of one of the sectors of a track. This sector (and all others of the track) has been written offtrack.

TP1

TP8 (VCO) tries to correct for the offtrack condition by boosting its amplitude voltage. This effect is seen at the left.

TP8

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: .2 msec/div

Suspect:
1. Analog PCB
2. HDA

82

Example of Abnormal VCO charge pump Signal

During a normal read operation, the Pro-File locks to the analog data signal by comparing the phase of the clock in the analog read signal with the phase of the VCO. If the VCO leads or lags the analog data clock a positive or negative correction voltage is stored in the charge pump capacitor. The charge on this capacitor is the voltage controlling the VCO, and can be seen at TP8 on the Analog PCB (usually ± 1 volt Peak to Peak).

To change the frequency, the Pro-File changes an input voltage to the VCO.

If the VCO charge pump signal (TP8) is observed along with the data signal (TP1) and a large deviation in VCO charge pump voltage appears (similar to the spikes in the upper photo), the VCO circuitry on the Analog PCB could be causing the abnormal VCO charge pump voltage to occur.

By changing the timebase to read a single sector (the lower photo), notice that both the analog data signal at TP1 and the VCO charge pump voltage at TP8 appear abnormal.

TP1 displays an offtrack condition (noted by the pinching of the signal following the sector mark), where the head is not positioned directly over the written data signal.

This offtrack condition can be either caused by an electronic or HDA related failure. (There are more examples of an offtrack data signal earlier in this procedure.)

As it turned out, every track on the particular HDA which produced this picture showed an analog data signal similar to TP1 (in the lower photo).

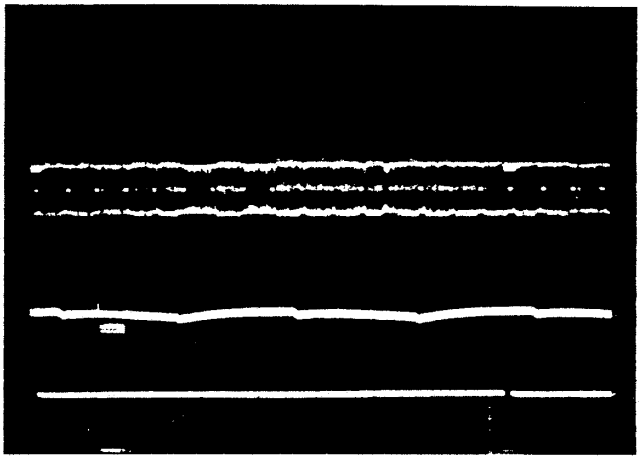
Because every sector of every track on the HDA was offtrack, it might be concluded that the HDA is bad because of the consistency in head misalignment.

The problem could be remedied by reformatting, but this would not be a good thing to do as the reason why the Pro-File went offtrack is not known. (Unknown damage internal to the HDA itself)

83

Diagnostic Procedure

Example of Abnormal AGC
(Automatic Gain Control)

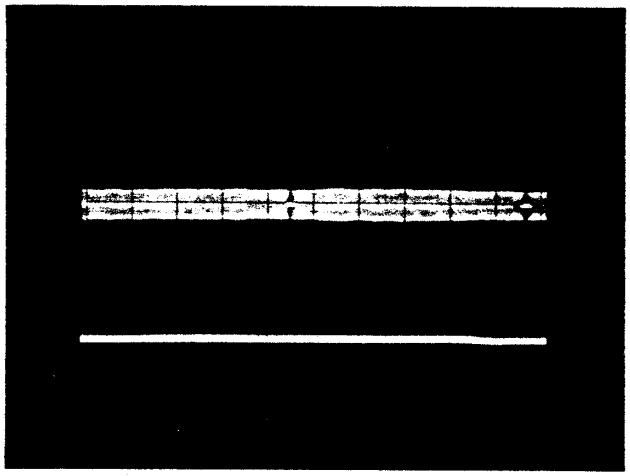


HORZ SWP = 2 MS/DIV

TP1 displays a view of a track of data. The track amplitude is about half the expected value of 2 mv p-p.

TP8 (VCO) is not used here

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: 2 msec/div



TP1 displays a closer view of some of the sectors of the track. Note that the sectors don't have the bell shape near the sector mark.

TP8 (VCO) is not used here.

Scope Settings:
TP1 - 1 volt/div
TP8 - .5 volt/div
Timebase: Variable

Suspect:
1. Analog Card
2. HDA

84

Example of Abnormal AGC

A normal data signal read at Test Points 1 & 2 on the Analog PCB have an average amplitude voltage of about 2 volts peak to peak.

If there is a malfunction in the amplifier circuits between the disc heads and TP1 & 2, the signal amplitude can be much less. For example, the analog to digital conversion circuits may be able to operate with voltages as low as 1.1 volt Peak to Peak.

Below 1.1 volt the analog circuitry just does not get a large enough amplitude signal to function.

The AGC circuitry was added to overcome normal changes in signal as the head seeks back and forth between the inner and outer tracks.

Decreases in the thickness of media coating or head flying height tend to cancel a loss in signal. Inner tracks will usually have the lowest amplitude signals.

In the upper photo at the left, the AGC circuitry was not functioning correctly as the amplitude was about 1 volt peak to peak.

After power up, the interrupter began to scan normally until about track 100, when the scan stopped.

Several hours later the interrupter was still stopped on the same track, unable to read the data present there.

The lower photo shows an expanded view of one entire sector from another track. Note that the normal flared shape present immediately after the sector mark is missing.

This missing "bell" shape is a characteristic of a faulty AGC circuit on the Analog PCB.

The Analog PCB was replaced and the Pro-File functioned normally.

85

SECTION 2
SERVICE PROCEDURES

86

SECTION 2 TABLE OF CONTENTS

87

HOW TO USE THIS SECTION:

This section contains the removal/replacement, check and adjustment, and software operation procedures you will need to repair the Pro-File.

If you are unsure which procedure to perform, go to the troubleshooting procedure in the Troubleshooting Section and follow the directions.

NOTE
THE BACK SIDES FOR PAGES 2.2 TO 2.60 ARE DOUBLE-COPIED AND SOMEWHAT HARD TO READ.

88

PROFILE MODULE REMOVAL AND REPLACEMENT PROCEDURES

INTRODUCTION

The following equipment will be needed in these procedures:

Diagonal cutters ("dikes")	Protective Pad
Tie Wraps	Medium Phillips Screwdriver
Small Flatblade Screwdriver	Needlenose pliers

CAUTION: The ProFile is a mechanical device with motors and moving parts. Rough handling such as dropping the drive, sharply jarring it or allowing heavy objects to fall on it can cause a malfunction. Whenever it is necessary to turn the ProFile over, be sure to rest it on a protective pad.

UNDER NO CIRCUMSTANCES SHOULD THE HDA BE ENTERED!

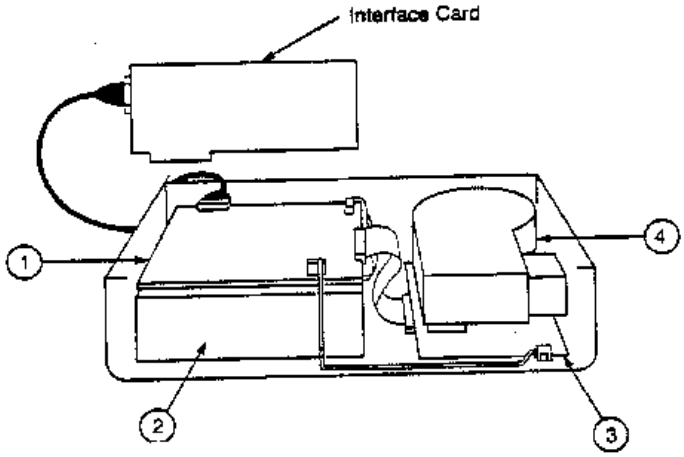


FIGURE 1

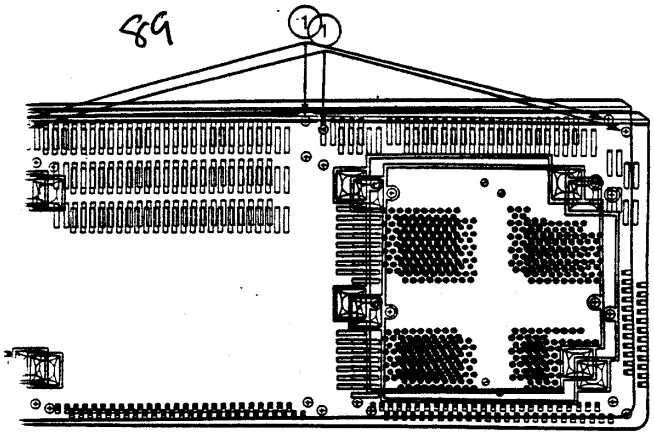


FIGURE 2

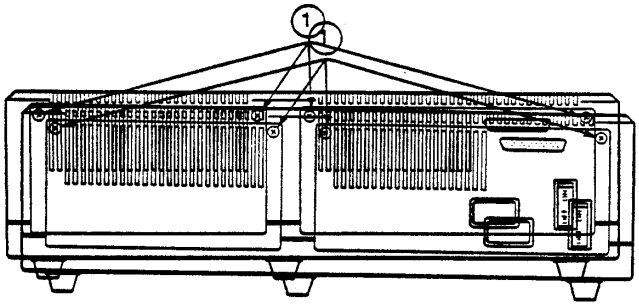


FIGURE 3

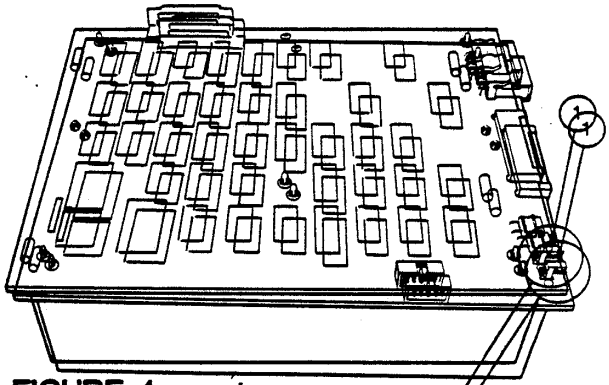


FIGURE 4

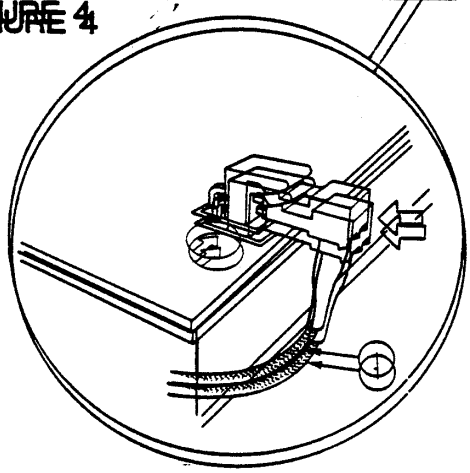


FIGURE 5

90

A. THE PRO-FILE COVER

Removing the Cover:

1. Make sure the ProFile is turned off. Disconnect the power cord and interface cable (ribbon cable) from the back of the ProFile.

2. Turn the ProFile over, lay it on the protective pad, and remove the three Phillips-head screws from beneath the front panel (Figure 2, #1).

3. Turn the ProFile right side up; loosen but do not remove the four screws on the back of the unit (Figure 3, #1).

4. Carefully pull out the lip on the bottom of the cover to disconnect it from the frame.

5. Take care not to pull on the LED cable as you carefully lift the cover off and rest it on the far side of the case, .

6. Unplug the LED cable from its socket on the controller PCB (Figure 4, #1).

Reinstalling the Cover:

1. Attach the LED cable to its connector on the controller PCB (Figure 5). Make sure the LED cable exits down and away from the PCB (Figure 5, #1).

2. Replace the ProFile cover. (Hints: The four slots on the back of the cover fit between the inner and outer rear plates, over the four loosened screws. Line up the back first; then pull the cover gently forward and down. Check around the cover to make sure the LED cable isn't caught between the cover and the base; then tighten the four rear-plate screws.)

3. Turn the ProFile over and replace the three screws on the front edge (Fig. 2).

4. Turn the ProFile right side up. Reinstall the power cord and the interface cable.

91

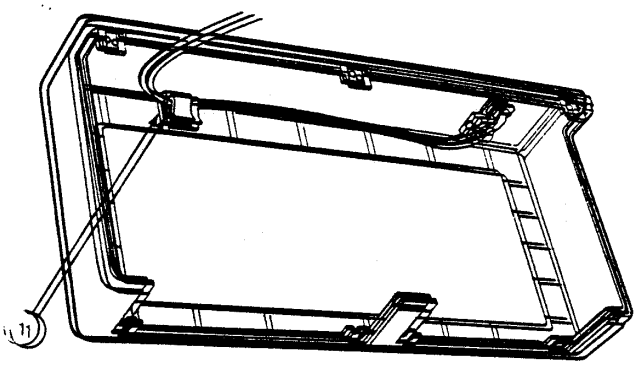


FIGURE 6

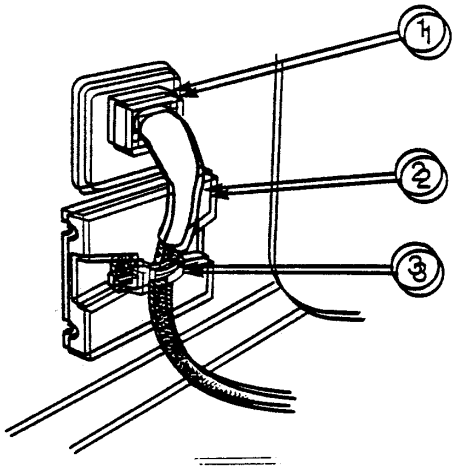


FIGURE 7

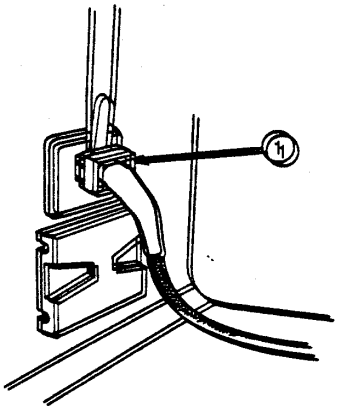


FIGURE 8

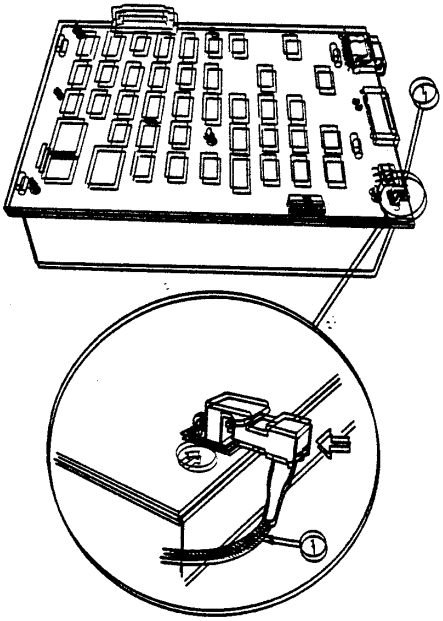


FIGURE 10

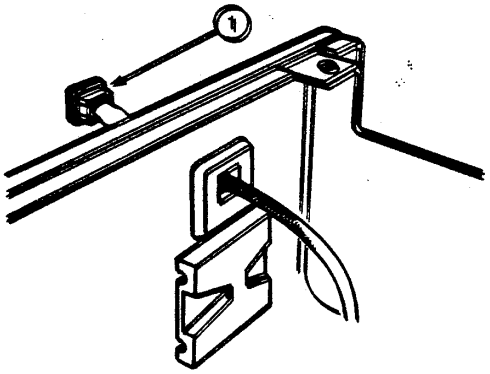


FIGURE 9

92

B. THE READY L.E.D.

Removing Ready LED:

1. Remove the ProFile cover and disconnect the LED cable from the controller PCB. Lay the cover flat, as in Figure 6.

2. Cut off the white plastic tie (Figure 7, #3) that holds the LED cable to the white holder in the cover (Figure 7, #2).

3. Remove the other end of the LED cable from the other holder in the cover. (On some ProFiles, this holder will be a clamp like the one shown in Figure 6, #1; on others, it will be like the one shown in Figure 7.)

4. With a flathead screwdriver, pry the cable clamp off the back of the LED (Figure 8, #1) and slide it down the cable, out of the way.

5. Gently push a few inches of the cable out through the slot in the cover, as shown in Figure 9.

   NOTE: You may have to remove the "ready" label around the LED opening on the cover to free the LED.

6. Around the red LED is a small black plastic mount. Remove the mount (Figure 9, #1) by pushing out its side flaps and sliding it off the LED.

7. Pull the cable back through the hole in the case.

Installing the Ready LED:

8. Thread the LED cable through the opening in the cover, and place the small black plastic mount on the LED (see Figure 9, #1). Then pull the cable back through the opening until the LED fits in its slot. Replace the "ready" label.

9. Push the cable clamp (Figure 7, #1) up to the cover until it holds the cable steady.

10. Place the LED cable against the white plastic holder and fasten it with a tie wrap (Figure 7, #2 and 3).

11. Place the cable in the other holder (Figure 6, #1), using a tie wrap if necessary. Cut off excess tie wrap.

12. Connect the LED cable to the controller PCB (Fig. 10) and reinstall the cover.

93

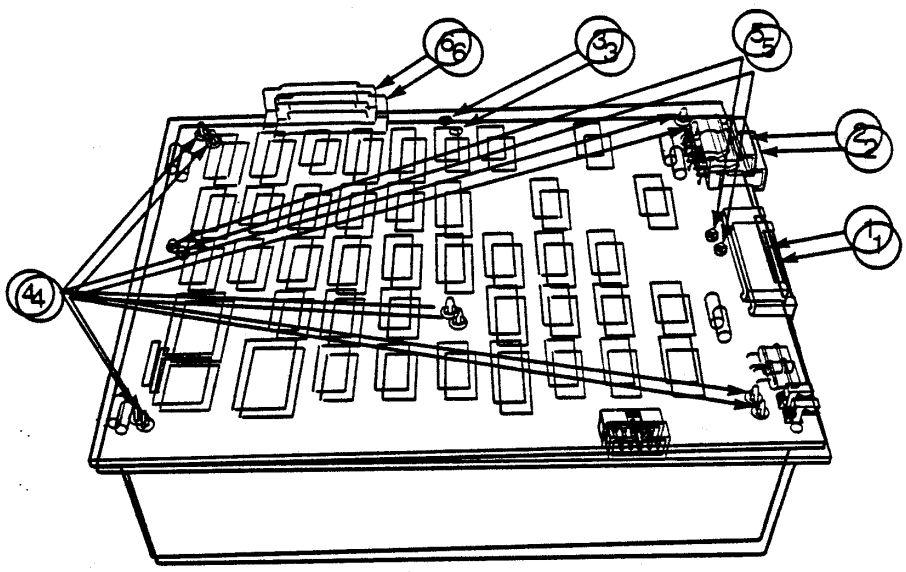


FIGURE 11

FIGURE 11

Service Procedures Page 2.8 rev 9-27-83

94

C. THE CONTROLLER PCB

Replacing the Old PCB:

1. Remove the cover and disconnect the LED from the controller PCB.

2. Disconnect the ribbon cable (Figure 11, #1) and (as far as you can) the orange Mylar motor control cable (Figure 11, #2) from the controller PCB. (You can finish disconnecting the motor control cable more easily after taking the PCB out.) NOTE: Do not pull on the ribbon cable (analog-to-controller cable). Push and wiggle the grey plastic connector until it comes free of the PCB.

3. Remove the long screw at the middle rear of the controller PCB (Figure 11, #3).

4. Use the needlenose pliers to push in the flanges of each of the five plastic stand-offs (Figure 11, #4), to release the PCB. NOTE: Some PCBs will have two additional stand-offs, or two screws to remove, as shown in Figure 11, #5.

5. Lift the controller PCB forward and up off the power supply - you may have to push out on the rear plate to free the interface cable connector (Figure 11, #6), and finish disconnecting the motor control cable. Set the PCB aside. NOTE: When you lift off the controller PCB, you will see a loose metal spacer under the spot where you removed the long screw. Save this spacer for use with the new PCB.

Installing the New PCB:

6. Holding the controller PCB over the power supply, position the interface cable connector (Fig. 10, #6) in its slot in the rear plate.

7. Position the controller PCB over the five plastic stand-offs, but don't push it down yet.

8. Connect a) the orange Mylar motor control cable and
   b) the analog-to-controller ribbon cable.

9. Slide the metal spacer under the PCB so that it is underneath the rear, middle screw hole. Insert the long screw through the hole and the spacer without tightening it.

10. Gently push the PCB down until the stand-offs hold it firmly in place; then tighten the long screw.

11. Connect the LED cable and replace the cover.

95

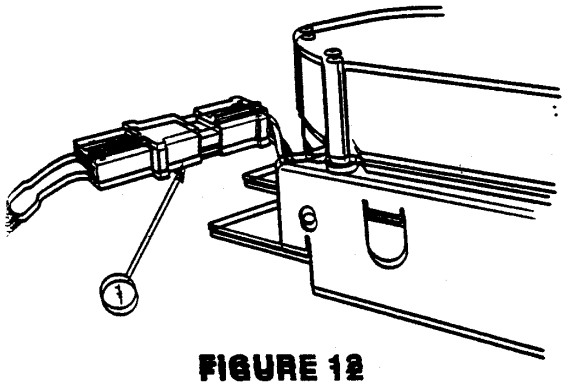


FIGURE 12

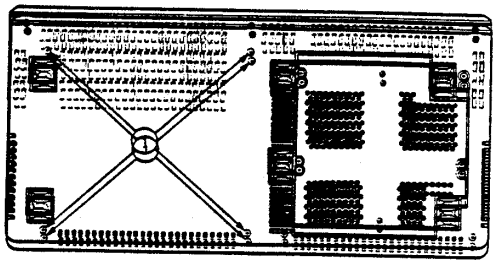


FIGURE 13

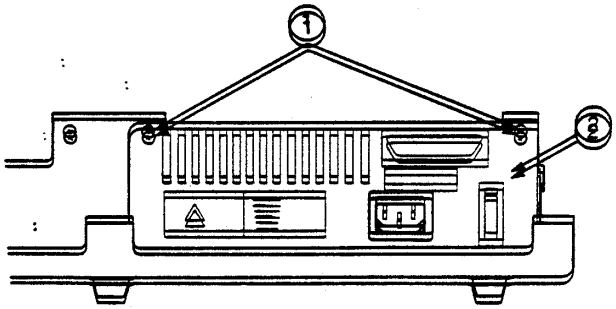


FIGURE 14

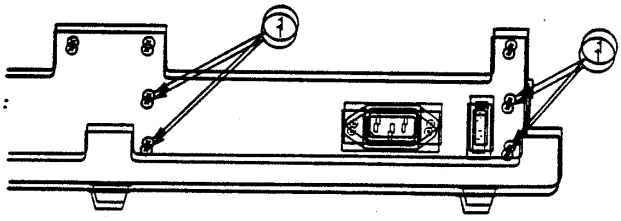


FIGURE 15

96

D. THE POWER SUPPLY

Removing the Old Power Supply

1. Remove the cover, disconnect the LED cable and remove the controller PCB.

2. Disconnect the small white power supply plug (between the power supply and the HDA) from its mate (Figure 12, #1).

3. Turn the ProFile over and rest it on a foam pad. Remove the four screws that hold the power supply onto the bottom of the ProFile case (Figure 13, #1).

4. Hold the power supply in place as you turn the ProFile right side up. Position it with its rear plates facing you. Find the longer of the two rear plates, remove the two screws that hold it on (Figure 14, #1), and remove the plate (Figure 14, #2).

5. Remove the four screws that hold the power supply to the inner rear plate (Figure 15, #1), and lift out the power supply.

Installing the New Power Supply

1. Position the power supply unit in the ProFile frame so that the two cables come out toward the HDA. Line up the four screw holes in the back of the power supply with the four lower holes in the inner rear plate; install the four screws.

2. Replace the outer rear plate. Insert the two mounting screws that hold this plate to the inner rear plate, but don't tighten them all the way.

3. Connect the two-pronged power supply cable to its mate from the HDA (Figure 12, #1); reconnect them **behind** the orange Mylar cable (i.e., closer to the HDA).

4. Holding the power supply in place, turn the ProFile over onto the workpad. Replace the four screws that hold the power supply to the bottom of the case. Turn the ProFile right side up.

5. Reinstall the controller PCB.

6. Connect the LED cable to the controller PCB and put the ProFile cover back on.

97

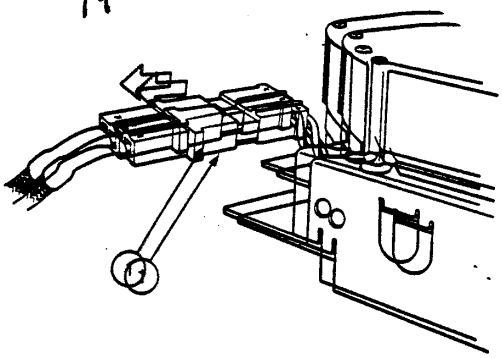


FIGURE 16

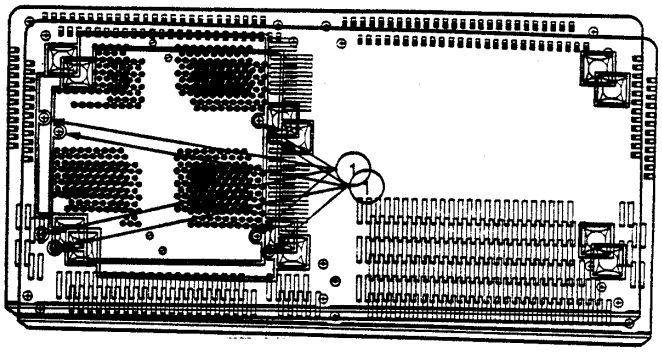


FIGURE 17

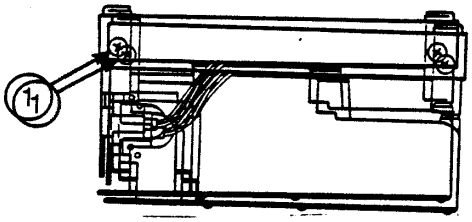


FIGURE 18

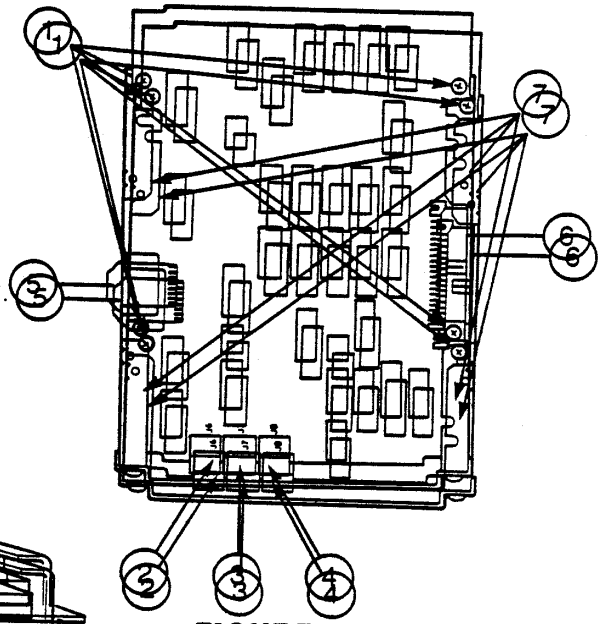


FIGURE 19

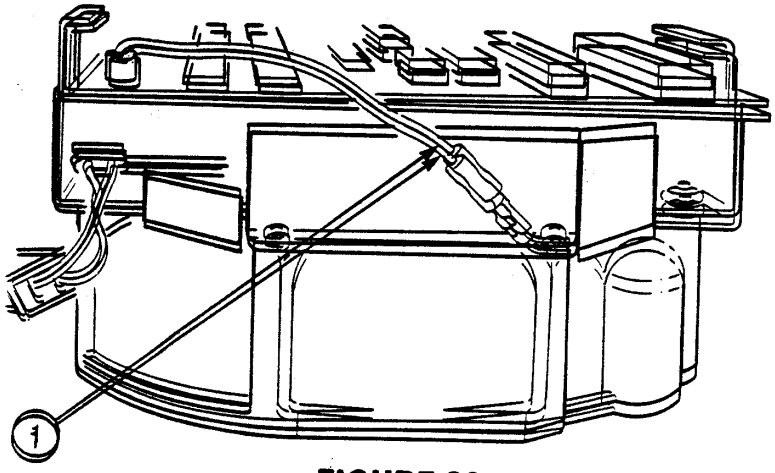


FIGURE 20

98

E. THE HARD DISK ASSEMBLY (HDA), AND ANALOG PCB

Removing the Old HDA and Analog PCB:

1. Remove the cover, disconnect the LED cable, and disconnect the cables that connect the HDA to the power supply and controller PCB.

2. Turn the ProFile over and rest it on the protective pad. Remove the four screws that mount the HDA to the ProFile housing (Figure 15, #1).

3. Carefully lift the housing up and off. Keep the HDA on the protective pad.

Removing Old the Analog PCB:

1. Turn the HDA so that the ribbon cable comes out to your right.

2. If there is a metal bar across the front of the analog PCB as shown in Figure 16, #1, or over the top of the analog PCB, remove the screws that hold the bar in place and set the bar aside. NOTE: This bar does not exist on later models.

3. Three small cables (Figure 17, #2, 3, 4) connect to one side of the PCB. The plug and socket of each cable should bear a corresponding number, so that you can tell which plug goes in which socket. If such numbers are not present, write them in with a marker.

4. Carefully disconnect all five cables from the analog PCB:
   1. Index cable (Figure 17, #2)
   2. Stepper motor cable (Figure 17, #3)
   3. Track 0 cable (Figure 17, #4)
   4. Head cable (Figure 17, #5)
   5. Analog-to-controller cable (Figure 17, #6)

5. Turn the HDA so that the rounded side is facing you, and remove the ground strap (Figure 18, #1) by grasping its plug firmly with the needlenose pliers and pulling.

   CAUTION: This is a short wire and its connection is tight. Use minimum force to prevent ripping it off the analog PCB.

6. Remove the four Phillips-head mounting screws from the analog PCB (Figure 17, #1).

7. Gently squeeze outward on the rails (Figure 17, #7) and lift the analog PCB from the HDA.

99

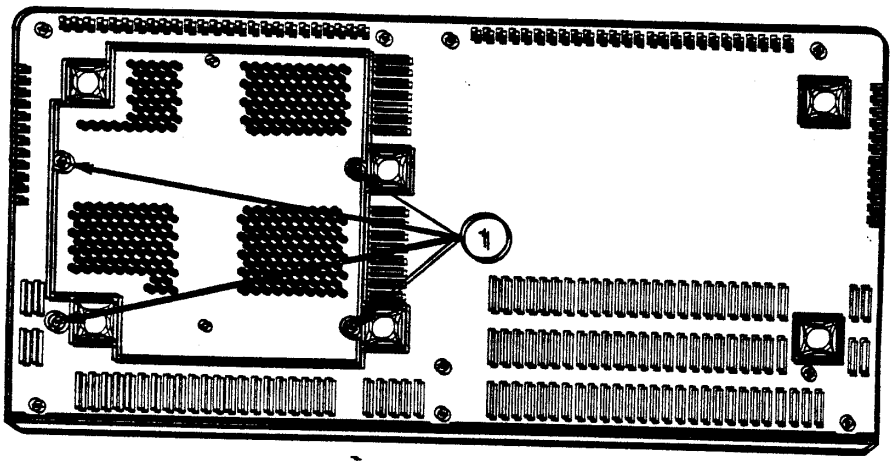


FIGURE 21

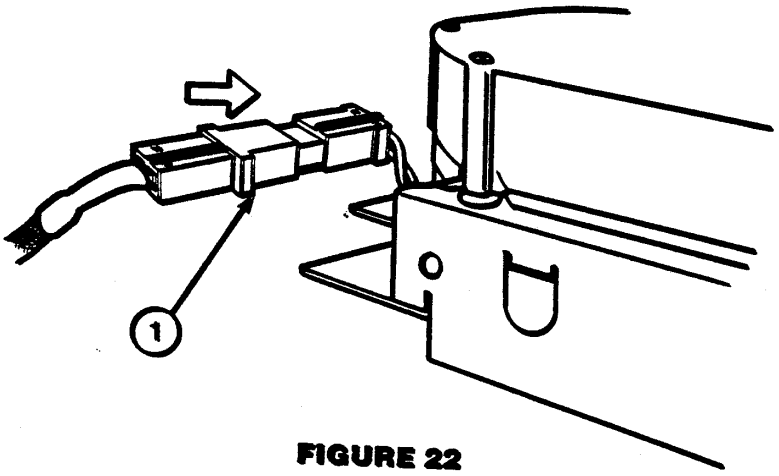


FIGURE 22

100

Installing the New Analog PCB:

8. Slide the analog PCB back under the rails. Line up the holes in the PCB with the screw holes on the rail.

9. Reconnect the analog-to-controller (ribbon) cable. (Do not use the arrow on the ribbon cable connector as a guide to connecting it. The ribbon cable should exit down and away from the analog PCB, and the side of the connector where the ribbon cable curls over a bar should be on top.)

10. Reconnect the other four cables. Check to make sure they are connected to the right plugs.

11. Replace the four Phillips-head mounting screws in the analog PCB.

12. Reconnect the ground strap.

13. If there was a metal bar over the rail, replace it and reinstall its screws.

14. Reinstall the HDA, reconnect all cables, and replace the cover.

Reinstalling the HDA:

1. Turn the HDA so that the ribbon cable comes out to your right.

2. Place the ProFile case over the analog PCB (see Figure 19: the rear plates of the case should be on the far side of the HDA). Line up the screw holes in the case with the screw holes in the railings over the analog PCB (Figure 19, #1).

3. Reinstall the screws.

4. Turn the ProFile right-side up, with the rear plate facing you. (The ribbon cable should now extend out from the bottom of the HDA.)

101

J2 ON THE OLD MOTOR CONTROL PCB, LEADS TO THE BRAKE

OLD MOTOR CONTROL PCB

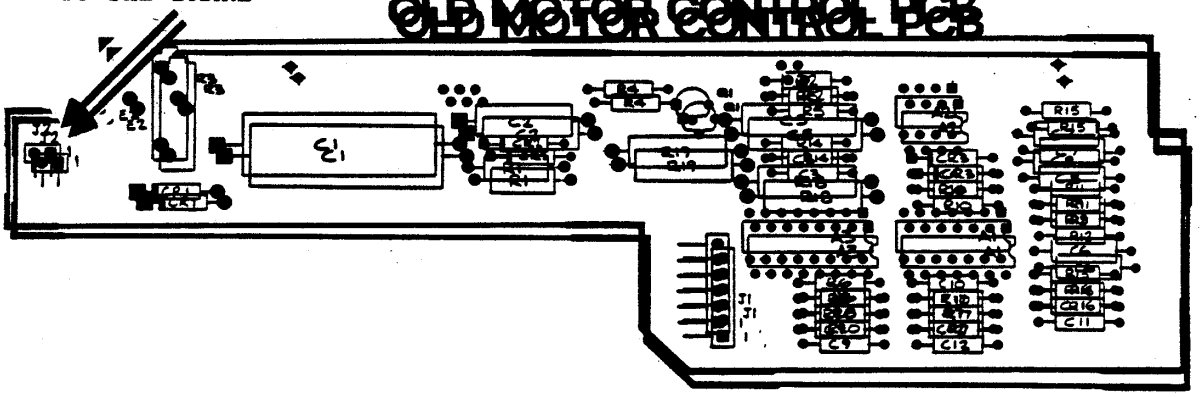
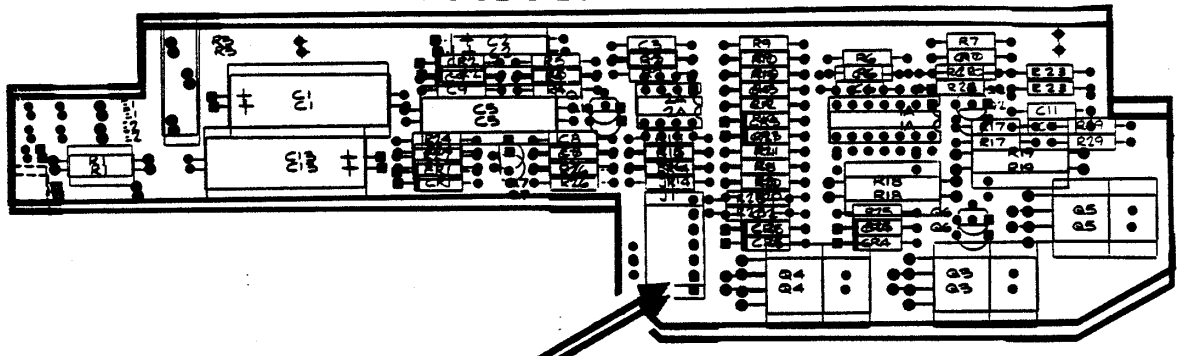


FIGURE 1

NEW MOTOR CONTROL PCB



ON THE NEW MOTOR CONTROL PCB, PIN 1 ON THE MOLEX PLUG IS AN EXTRA PIN

FIGURE 2

102

F. THE MOTOR CONTROL PCB

1. Remove the HDA and the Analog PCB as described in the HDA and Analog PCB Removal/Replacement procedure found in this section.

2. Refer to the illustrations on the opposite page to identify whether you have an old or new Motor control PCB. Your replacement motor control PCB should always be a new revision.

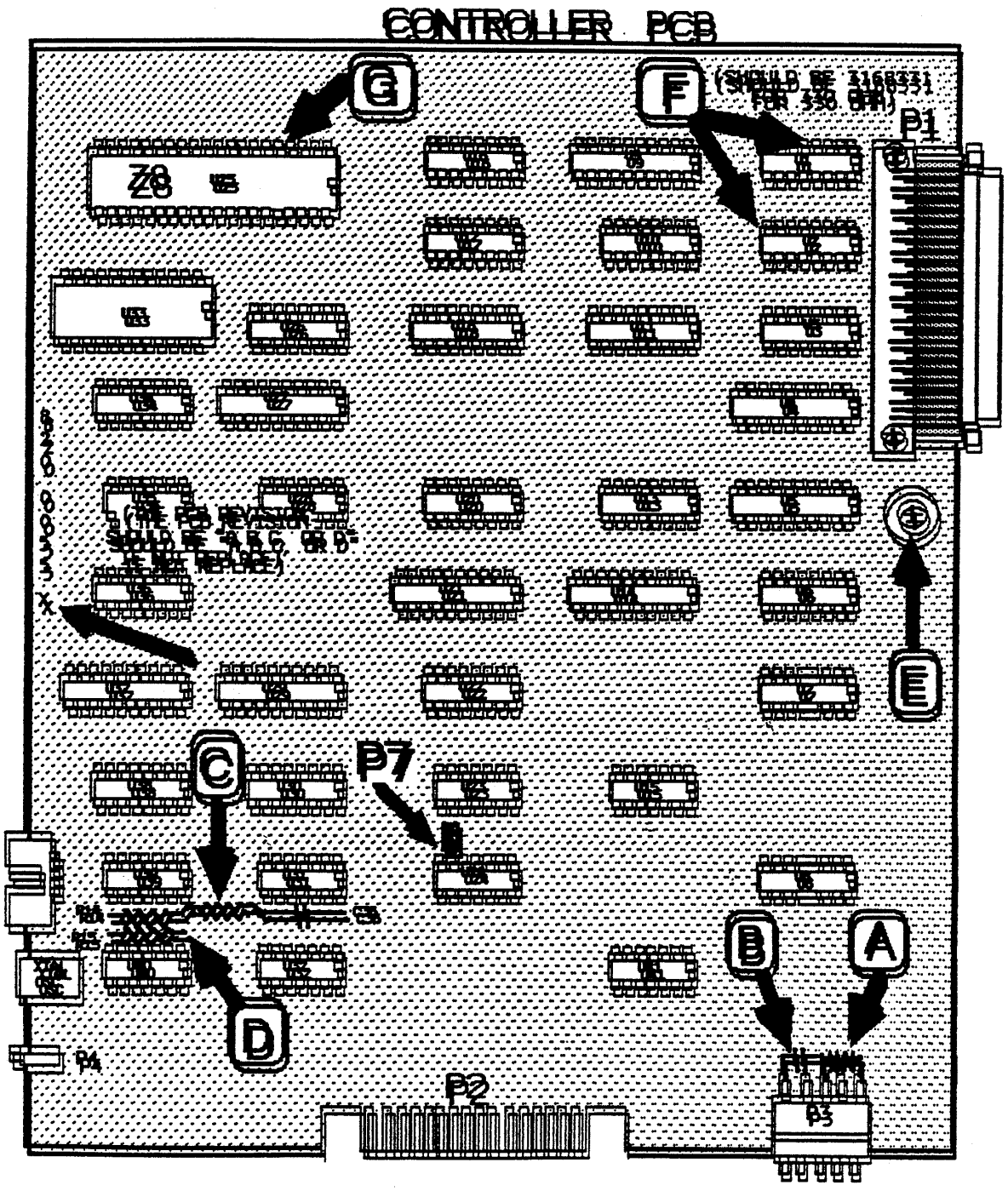
*** If you have an old motor control PCB, then you must remove the brake on the HDA in addition to removing the motor control PCB.

*** If you have a new motor control PCB then P1 may be a 7 pin Molex plug fitting on J1 a 6 pin jack. In this case pin 1 on the plug will be the extra pin with pins 2 through 7 of the plug fitting onto pins 1 through 6 of the jack.

3. Remove the motor control PCB by disconnecting all plugs, and removing the 2 screws fastening it to the HDA main assembly.

4. Reinstall the motor control PCB by reconnecting the plugs, and reinstalling the 2 screws to fasten the PCB to the HDA main assembly.

103



104

CONTROLLER BOARD UPGRADE PROCEDURE

The following equipment may be needed for this procedure:

PART	APPLE PART NUMBER
5.1K ohm Resistor (1)	101-4512
0.1 microfarad Capacitors (2)	130-0007
1K ohm Resistor +5% (1)	101-4102
6.36 X 5/16 Standoff (1)	860-0213
Masked Z8 Microprocessor (1)	341-0171-A
330 ohm DIP Packs (2)	112-0105

Identify the Rev level etched on both sides of the PCB. Refer to the illustration on the opposite page for the location of the items mentioned in the following discussion. If the Rev is 820-0055 Rev A, B, C, or D then perform the following checks and upgrades.

(There are a few controller PCBs in the field that do not have the 820-0055 number etched on the PCB. These PCBs should be replaced with an 820-0055 rev PCB.)

A. Solder a 5.1K ohm resister (PN 101-4512) between pins 1 and 3 of connector P3 on the controller bd.

B. Solder a 0.1 uf capacitor (PN 130-0007) between pins 3 and 5 of connector P3 on the controller.

C. Solder a 0.1 uf capacitor (PN 130-0007) between the upper lead of R14 (U40 pin 2) and the bottom lead of C36 (ground).

D. Check that R15 is present on the PCB. If R15 is missing or one lead is cut then replace it with a 1K ohm 5% resister (PN 101-4102).

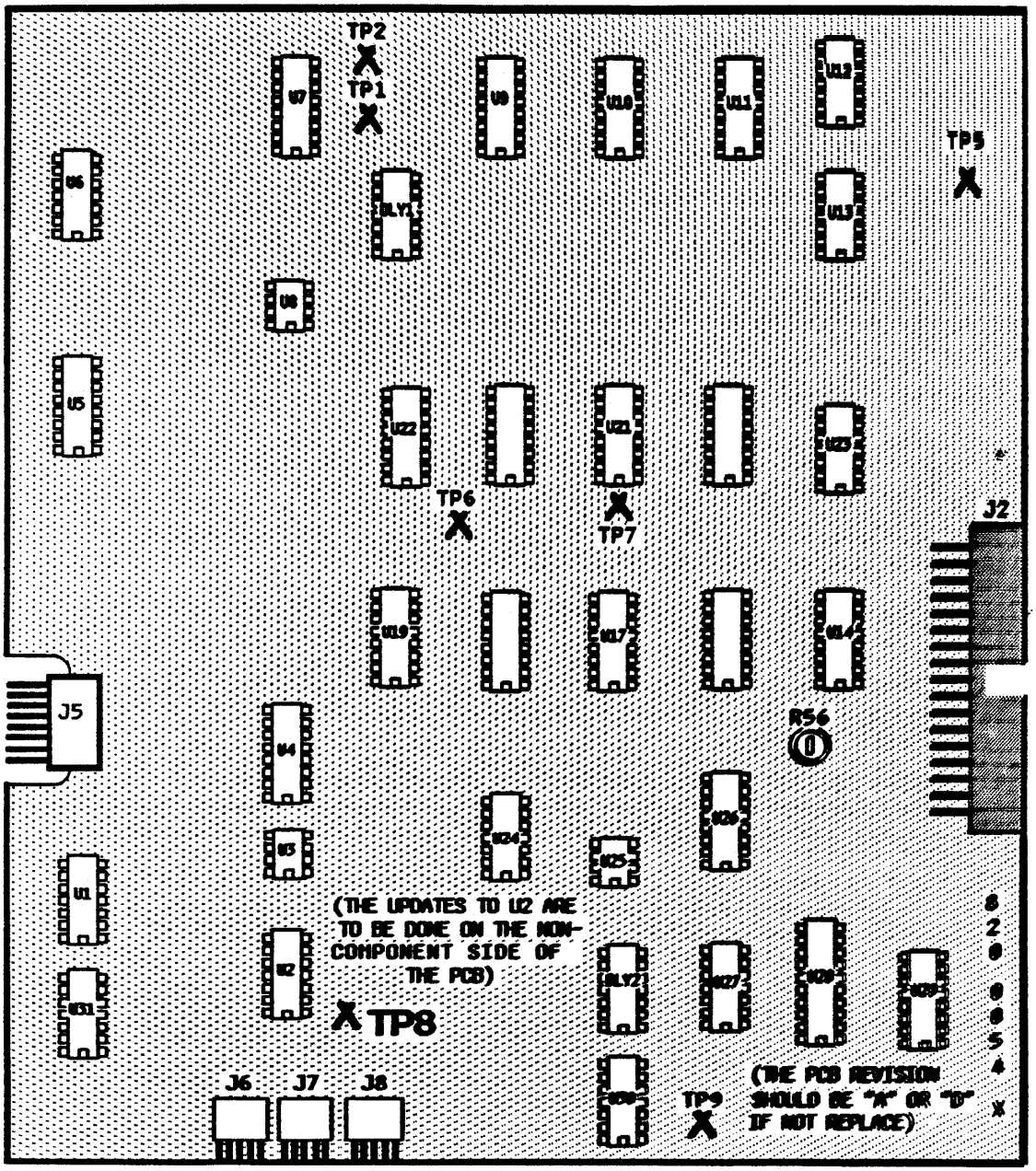
E. Check that the metal standoff between the controller PCB and the power supply assembly is 5/16" in length. If it is not, replace it with a 6-36 X 5/16" standoff (PN 860-0213).

F. Check that U1 and U2 on the controller PCB are between 330 and 332 ohm DIP packages. If they are any other value, then replace them with 330 ohm DIP packages. (You may wish to verify the resistance of the current DIP packages with an ohm meter.)

G. Check the 28. If it is the piggyback version then replace it with the latest mask version (no piggyback) (PN 341-0171-A).

105

ANALOG PCB



106

ANALOG BOARD UPGRADE PROCEDURE

The following equipment may be needed to perform this procedure:

PART	APPLE P/N
20mm section of 26 to 30 gauge insulated wire	N/A
12mm section of 26 to 30 gauge insulated wire	N/A
3.9K ohm Resistors (2)	101-4392

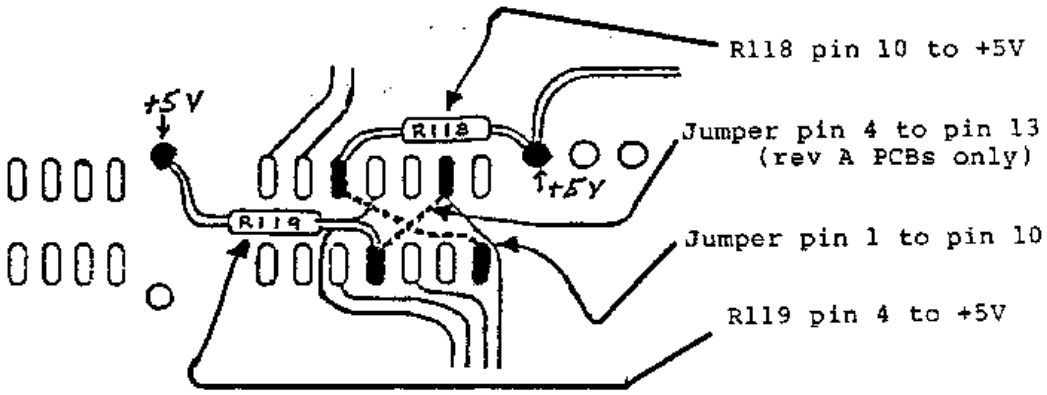
1. Identify the rev level of the PCB. The PCB part number is etched at the bottom left on both sides of the PCB. The number will be either 820-0054-A (rev A) or 820-0054-D (rev D). (No rev "B" or rev "C" PCBs were ever made for production) There are a few older analog PCBs in the field that do not have the 820-0054 number and alphabetic (i.e A, or D) identifier etched on the PCB. These should be replaced with rev "A" or "D" PCBs.

2. Locate device U2 and turn the PCB over.

3. If the PCB is rev "D", this will be the final step (skip step 4). If the PCB is rev "A" perform both this step and step 4.
   A. Solder an insulated wire approximately 3/4 (15mm) long from pin 1 of U2 to pin 10 of U2.
   B. Solder a 3.9K ohm resistor (Apple PN 101-4392) between pin 10 of U2 and +5V. This will be designated R118 on updated schematics.
   C. Solder a second 3.9K ohm resistor from pin 4 of U2 to +5V. Ths will be designated R119 on updated schematics.

4. Solder an insulated wire approximately 1/2" (9.5mm) long from pin 4 of U2 to pin 13 of U2.

Underside of Analog PCB Location U2



107

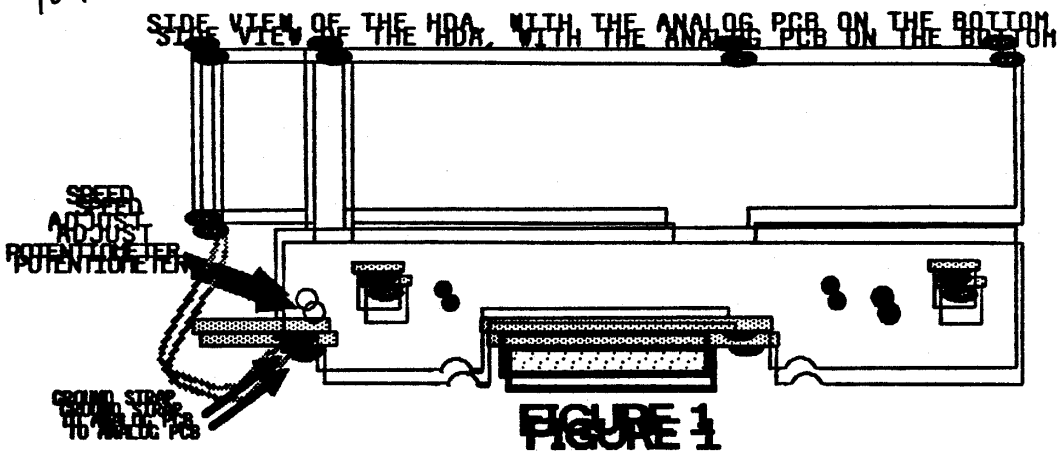


FIGURE 1

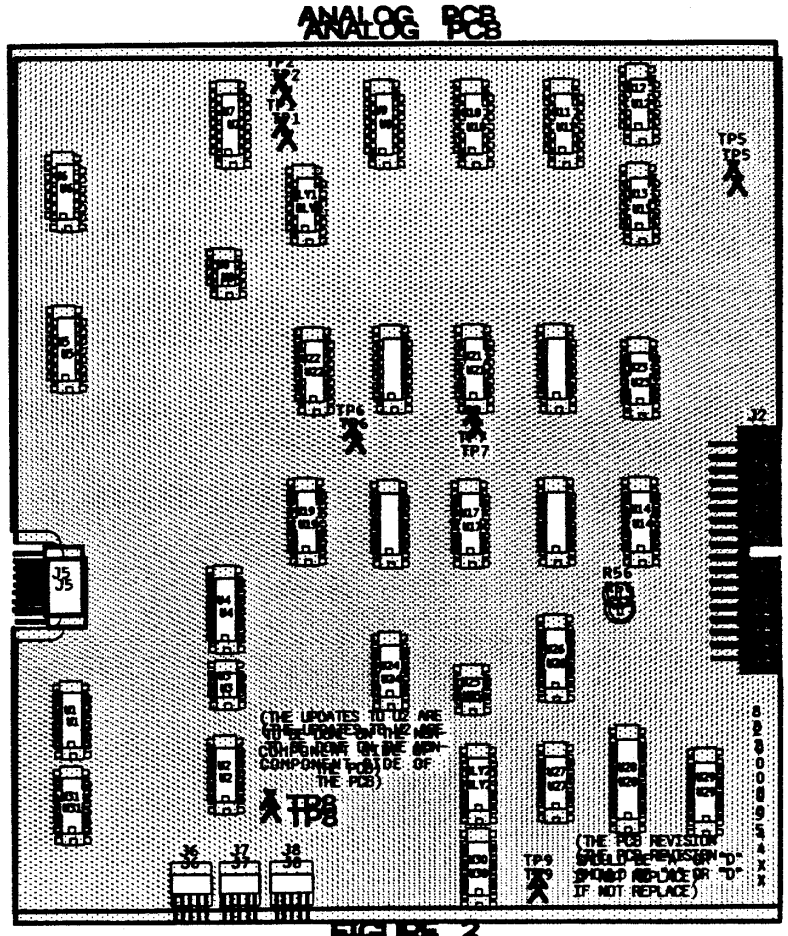


FIGURE 2

108

HDA SPEED CHECK AND ADJUSTMENT PROCEDURE

The following equipment will be needed to perform this procedure:

Frequency Counter and probes
Anti-sabotage Sealant (e.g. glyptol)
Tweaker (1) for adjusting potentiometers

Note: If you need instructions for any removal/replacement, you can find them in the Module Removal/Replacement Procedures part of this section.

To CHECK:

(1) Remove the cover from the Profile.

(2) Connect the black lead from the frequency counter to a ground point on the analog PCB. Connect the red lead to TP-1 on the Controller PCB.

(3) Set the frequency counter to read Per. A @ 10 hz.

(4) Turn on the Profile and allow 30 seconds for the HDA motor to get up to speed. The frequency counter should be showing the motor speed to be within 1% of 16667 ms. (16500 to 16832 ms.) If the motor speed is not within this range then perform the adjustment procedure below. If the motor speed is within the proper range reinstall the HDA assembly and cover on the Profile.

To ADJUST:

(1) Remove the HDA assembly from the Profile. Once removed, reconnect the ribbon cable from the HDA to P2 on the Controller PCB, and power connector P4 from the Power Supply to the HDA.

(2) Locate the trimmer potentiometer positioned near the ground strap on the Analog PCB by P4.

(3) Adjust the trimmer potentiometer until the HDA motor speed is within range (after each turn of the potentiometer allow time for the motor speed to settle).

(4) Once the motor is within the specified limits, lock the adjusting screw on the trimmer pot with anti-sabotage sealant.

(5) Reinstall the HDA assembly and cover.

109

INDEX CHECK AND ADJUSTMENT PROCEDURE

The following equipment will be needed to perform this procedure:

- 5/64 Allen (hex) driver
- .030 inch flat gauge

Note: If you need instructions for any removal/replacement, you can find them in the Module Removal/Replacement Procedures part of this section.

To CHECK:

(1) Remove the Cover, and then the HDA assembly, from the Profile.

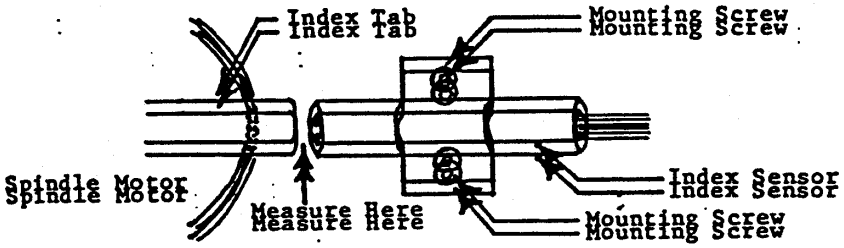
(2) Remove the Analog PCB from the HDA.

(3) Turn the HDA upside down in front of you. Locate the index mechanism as shown in figure 1 below.

(4) Rotate the spindle motor until the silver index tab on the spindle motor is aligned with the index sensor.

(5) Slide the gauge into the gap between the silver index tab on the spindle motor and the index sensor. If the gap is too small for the gauge, or the gauge is not snugly held in the gap, perform the adjustment below. If the gap is OK, reinstall the Analog PCB on the HDA assembly and the HDA assembly in the Profile. Then reinstall the Cover.

INDEX MECHANISM



(Fig: 1)

(This procedure is continued on the next page:)

110

INDEX CHECK AND ADJUSTMENT PROCEDURE (continued)

To ADJUST:

(1) Loosen the two mounting screws so that the index sensor moves freely.

(2) Align the silver index tab and the index sensor.

(3) Place the 30 mil. flat gauge between the index tab and the index sensor.

(4) Push the index sensor forward (towards spindle motor) until it makes light contact with flat gauge.

(5) Tighten the mounting screws.

(6) The flat gauge should be held firmly but should not be so snug that it cannot be easily removed.

(7) Remove the flat gauge and check the operation of the index sensor.

(8) Reinstall the Analog PCB into the HDA assembly, and the HDA assembly into the Profile.

BRAKE CHECK AND ADJUSTMENT PROCEDURE

The following equipment will be needed to perform this procedure:

- 5/64 Allen (hex) driver
- .010 inch flat gauge
- Locktite thread sealer

There are 2 main revisions of motor control PCBs for the Pro=File HDA. The older rev uses a brake to slow the HDA disk motor after power is removed. It is for this rev of motor contol PCB that this procedure was developed. The newer rev reverse biases the disk motor to slow the disk after power is removed. You can identify which PCB you have by referring to the illustration on page 2.16.

Note: If you need instructions for any removal/replacement, you can find them in the Module Removal/Replacement Procedures part of this section.

To CHECK:

(1) Remove the Cover and then the HDA assembly from the Profile.

(2) Remove the Analog PCB from the HDA.

(3) Turn the HDA upside down in front of you. Locate the brake mechanism as shown in figure 1. Insert the flat gauge between the Soleneid Body and the Plunger Plate as shown in figure 1.

The gauge should fit snugly but slide easily.

If the clearance is too tight or too loose, perform the procedure below.

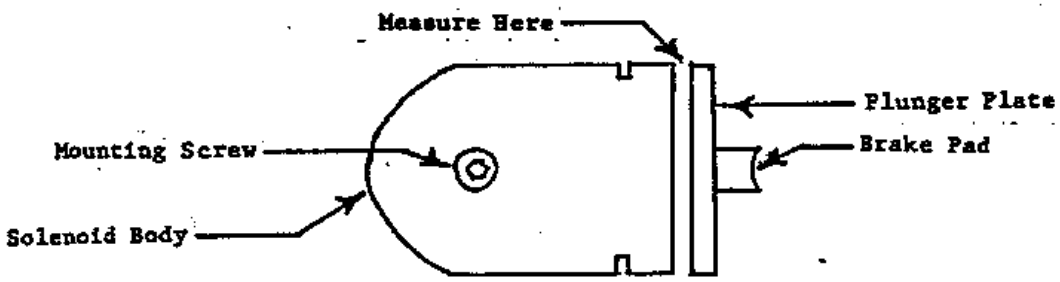
If the clearance is OK reinstall the HDA assembly, and then reinstall the Cover.

(The adjustment procedure is continued on the next page.)

112

BRAKE CHECK AND ADJUSTMENT PROCEDURE (continued)

BRAKE MECHANISM



(Fig.1)

To ADJUST:

(1) Remove the mounting screw and coat the end threads with locktite.

(2) Carefully replace the mounting screw; avoid getting any locktite on the sides of the screw hole in the solenoid body.

(3) Run in the mounting screw until snug, then back out 1/4 turn.

(4) Place the flat gauge between solenoid body and plunger plate.

(5) Push the solenoid body forward (toward spindle motor) until the brake pad makes light contact with the braking surface of spindle motor.

(Hint: an Allen driver or similar object placed through the screw hole in the side bracket behind the solenoid body provides a convenient method of positioning the solenoid).

(The adjustment procedure is continued on the next page.)

113

BRAKE CHECK AND ADJUSTMENT PROCEDURE (continued)

(6) Tighten the mounting screw.

(7) The flat gauge should be held firmly by the solenoid but should not be so snug that it can not be removed easily.

(8) Remove flat gauge and check operation of the brake by performing the following steps:

a. Carefully position the HDA on a level surface, with the brake on top.

b. Reconnect P4 (no other HDA connections are necessary) from the HDA to the Power Supply.

c. Plug the Power cord into the Profile and turn it on.

Observe: That when the spindle motor begins to spin, the brake solenoid energizes and pulls the brake pad away from the braking surface of the spindle motor.

d. Turn the Profile off.

Observe: That the brake solenoid deenergizes and the brake pad is released to contact the braking surface of the spindle motor.

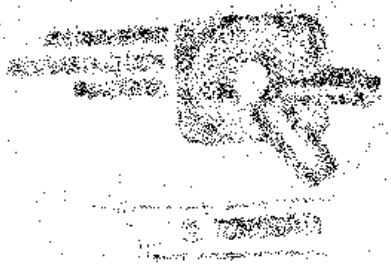
WARNING: Disconnect Power from the Profile before going any further.

(9) If the brake "chatters", a possible remedy is canting or tilting the solenoid body left or right.

(10) Reinstall the Analog PCB on the HDA assembly, and the HDA assembly in the Profile.

114

THIS PAGE LEFT BLANK INTENTIONALLY



115

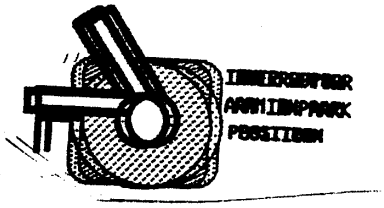


FIGURE 11

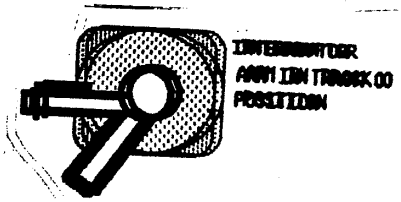


FIGURE 2

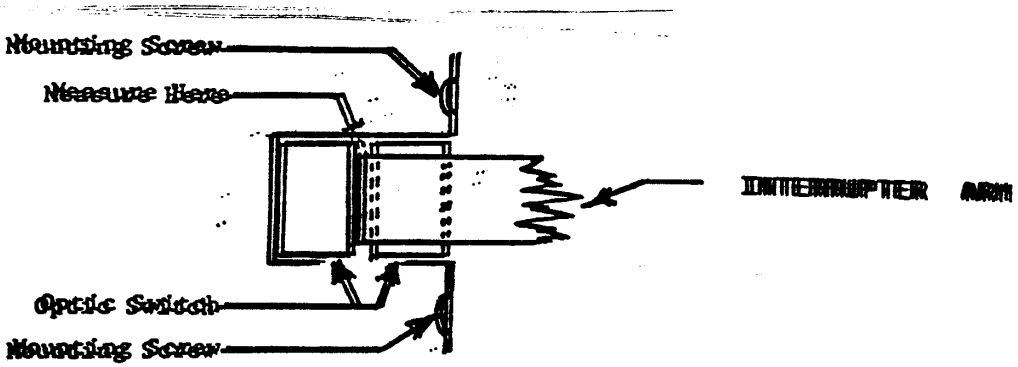


FIGURE 3

116

TRACK 0 CHECK AND ADJUSTMENT PROCEDURE

The following equipment will be needed to perform this procedure:

Profile Debug Z8
.010 inch wire gauge
.070 inch wire gauge
A/// with internal disk drive
.050 inch Allen (hex) driver

Note: If you need instructions for any removal/replacement you can find them in the Module Removal Replacement Procedures in this section.

To CHECK:

(1) Remove the cover from the Profile.

(2) Position the Profile so that the interruptor arm on the HDA is in front of you.

(3) On the Controller PCB remove the system Z8 with masked ROM, and install a piggyback Debug Z-8 into its socket.

(4) Turn on the Profile

Observe: That after approximately 20 seconds the HDA interrupter arm will move from Park position to track 0 position as shown in figures 1 and 2.

(5) Use the .010 inch and the .070 inch wire gauges to insure that the clearance between the top of interrupter arm and optic mounting bracket (See Fig. 3) is between .010 and .070 inches. If it is not then perform the adjustment procedure on the next page.

NOTE: If the following adjustment does not work, try adjusting the position of the interrupter arm on its shaft to obtain the proper clearance.

(The adjustment procedure is located on the next page.)

117

TRACK 0 CHECK AND ADJUSTMENT PROCEDURE (continued)

To ADJUST:

(1) Loosen the mounting screws on the optic bracket.

(2) Slide the bracket/optic assembly towards the top of HDA to open the gap between the interrupter arm and the optic mounting bracket.

(3) Slide the 50 mil. wire gauge into the gap, being careful not to disturb interrupter arm.

(4) Slide the bracket assembly down until the top of the bracket and the interrupter arm close lightly on the gauge.

(5) Carefully tighten mounting screws so as not to disturb the adjustment.

(6) Turn off the Profile and replace the Debugger Z8 with the System Z8.

(7) Turn on the Profile and wait approximately 2 minutes as it performs its Scan sequence. At the end of the Scan sequence the interrupter arm should be returned to the Park position.

(8) Turn off the Profile.

(9) On the Controller PCB remove the System Z-8, and install the piggyback Debug Z-8 in its socket.

(10) Turn on the Profile and recheck the clearance between the optic bracket and the interrupter arm as described above. Reperform the adjustment until the clearance is OK.

(11) Turn off the Profile and replace the Debug Z8 with the System Z8.

(12) Reinstall the Cover on the Profile.

118

HDA FORMATTING PROCEDURE

The following equipment will be needed for this procedure

Apple ///
Profile Interface PCB for the Apple ///
Profile Interface Cable
Jumper (a 2 inch length of 22 ga. wire with small alligator clips at both ends will do)
Debugger Z8
Format software (Profile format diskette #889-0013)

1. Install the Profile Interface PCB in slot 1 of the Apple /// and connect the Profile Interface Cable from it to the Profile.

2. Remove the cover from the Profile. If instructions are needed refer to the Module Removal and Replacement Procedures in this section.

3. Remove the system program Z8 from the Controller PCB and insert a Debugger Z8. If at all possible do not remove the prom from the piggyback socket as this can cause intermittent failures because of damage to the piggyback socket.

4. Turn on the Profile and wait 30 seconds for the HDA motor to come up to speed.

   Note: When the Profile is turned on, its firmware will flash the Ready LED as the power comes up. After that function is performed the firmware program in the Debugger Z8 will not turn the Ready LED on again until the program in the Apple /// initiates communications with the Profile.

5. Boot the Format program.

   Observe: the Ready LED comes on steadily. If it does not, check all connections between the Apple /// and the Profile.

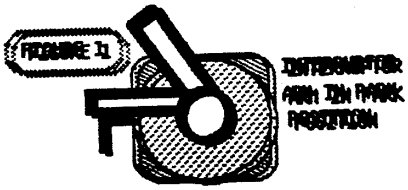
   Note: the program may prompt you that the Profile has incorrect FD.ROM version 03.11. This is normal the Format program will work OK.

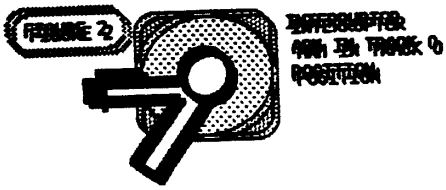
119

6. Press <RETURN>, then as prompted by the display, install the jumper to short the two pins of P7 on the Controller PCB together (refer to the illustration of the Controller PCB on page 2.18 to help you find P7).

Note: the statement "press any key", as now displayed, is incorrect, it should read "press RETURN" THIS IS TRUE FOR THE REST OF THE FORMAT PROCEDURE

7. Press <RETURN> and watch the interrupter arm on the Hard Disk Assembly (HDA), it should step from track 0 to the park position as shown below in approximately 180 seconds.





Note: If the arm did not complete the scan swap the HDA to a known good Profile and retry. If Format still does not occur the HDA is probably faulty, replace it.

8. After completion of the scan the screen display will tell you if the formatting was successful or not. (if there was an error in formatting the screen will display an error code, recheck all connections on the Pro-File and between the Pro-File and the host computer.)

Observe: If the formatting was successful the Ready LED will come on steadily and the screen will display.

"Formatting completed. Pass
Remove jumper. Press any key to continue"

120

Caution: At this time remove the jumper from P7 to prevent damage to the unit.

9. Press <**RETURN**> as prompted . The following should occur within the next 2 and 1/2 minutes:

Observe: The profile will now automatically:
* Scan all sectors, heads and tracks
* Compare buffers on track 77
* Certify all sectors on track 77
* Initialize spare tables on track 77

At the end of each process listed above the screen display should indicate "pass".

Note: the last line on the screen display "press <**RETURN**> when profile is ready" only indicates that if you have more than one profile to format, the diskette does not have to be rebooted each time.

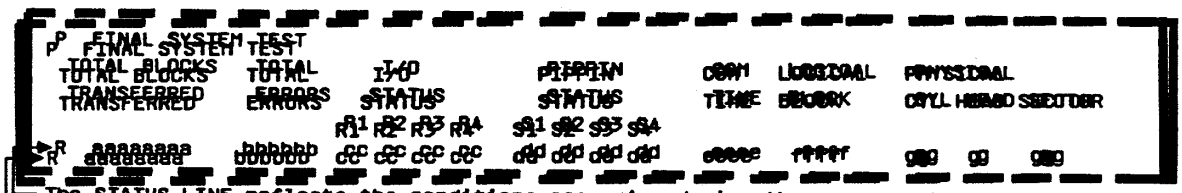
121

FINAL SYSTEM TEST (FST) PROCEDURE

The following equipment will be needed for this procedure

Apple ///	Apple /// ProFile Interface PCB
FST Software (ARN #8890029)	ProFile Interface Cable

1. Install the ProFile Interface PCB in slot 1 of the Apple /// and connect the ProFile Interface Cable from it to the ProFile.
2. Ensure that there is a system ROM Z8 installed on the Controller PCB.
3. Turn on the ProFile and wait 2 minutes for the ProFile to complete its Scan sequence.
4. Boot the FST program. Observe that for 10 minutes the ProFile will write and verify each track in sequence from track 0 to track 152 (the Ready LED will blink as each operation occurs). You should see a screen like the one below.
5. Run FST for the specified period (15 minutes, or 48 hours). If an error line occurs (described below) determine if it is fatal by checking the MAX ERRORS ALLOWED column for the error code on the Criteria Sheet on the opposite page.
6. To terminate the test press <ESCAPE>.



a = Total number of blocks transferred during the test. This number will be continuously incrementing as long as the test is being run.

b = Total errors that have occurred throughout the test. This number should equal the number of error lines at the bottom of the FST screen.

c = These bytes relate to the I/O status bytes the ProFile sends to the host operating system during communications.

d = These bytes relate to the particular status codes that were sent from the ProFile to the host (FST test program).

e = Indicates the time elapsed for the current block transfer.

f = The Logical block sent by the host (FST test program).

g = Indicates the Cylinder (Track), Head, and Sector being read from or written to.

IF AN ERROR OCCURS, THE STATUS LINE FOR THAT ERROR WILL BE PERMANENTLY DISPLAYED AT THE BOTTOM OF THE FST SCREEN. THE ERROR CODE IS TAKEN FROM THE "d" FIELD AS DESCRIBED ABOVE. YOU MAY INTERPRET THIS ERROR BY COMPARING IT TO THE CRITERIA SHEET ON THE OPPOSITE PAGE. A MORE DETAILED EXPLANATION MAY BE FOUND ON PAGE 2.58 AND 2.59. THE LOCATION OF THE SECTOR THE ERROR OCCURRED ON MAY BE FOUND IN THE "g" FIELD OF THE ERROR LINE.

122

FINAL SYSTEM TEST (FST) CRITERIA SHEET

10/20/83

FINAL SYSTEM TEST (FST) ERROR STATUS CODES	MAX ERRORS ALLOWED
00 [illegible] 00 00 — SEEK TO WRONG TRACK	6
00 [illegible] 00 00 — CAN'T READ 3 SECTORS AFTER SEEK	0
00 [illegible] 00 00 — SEEK TO WRONG TRACK; CAN'T READ 3 SECTORS AFTER SEEK	0
04 00 00 00 — CAN'T FIND TARGET HEADER IN 9 REVS	0
04 04 00 00 — SPARING OCCURRED; CAN'T FIND TARGET HEADER IN 9 REVS	2
04 04 00 00 — UNABLE TO READ STATUS SECTORS; CAN'T FIND TARGET HEADER IN 9 REVS	0
05 00 00 00 — CAN'T FIND TARGET HEADER IN 9 REVS, REQUESTED OPERATION FAILED	0
[illegible] 00 00 00 — CAN'T READ 3 SECTORS AFTER SEEK; CAN'T FIND TARGET HEADER IN 9 REVS	0
00 00 00 (01 TO 09) (SEE NOTE 3) — SOFT READ ERROR	99
00 00 00 (10 OR HIGHER) (SEE NOTE 3) — HARD READ ERROR	8
00 04 00 00 — SPARING ROUTINE; READ ERROR	10
[illegible] 00 00 00 — READ ERROR OR REQUESTED OPERATION FAILED	10
[illegible] 00 00 — CAN'T READ 3 SECTORS AFTER SEEK; CAN'T READ 3 SECTORS AFTER 2 RESEEKS	0
15 [illegible] 00 [illegible] — CAN'T READ 3 SECTORS AFTER SEEK; CAN'T FIND TARGET IN 9 REVS OR REQUESTED OPERATION FAILED; CAN'T READ 3 SECTORS AFTER 2 RESEEKS	0
15 [illegible] 00 00 — SEEK TO WRONG TRACK; CAN'T READ 3 SECTORS AFTER SEEK; CAN'T FIND TARGET IN 9 REVS OR REQUESTED OPERATION FAILED; CAN'T READ 3 SECTORS AFTER 2 RESEEKS	0
[illegible] 00 00 00 — 3 BYTE SEPARATOR BETWEEN WRITE BUFFER AND STATUS TABLES OR $2-3 SET. SPARE TABLE UPDATE OCCURRED, BUFFER DATA IS CHANGED.	0

NOTES:

1. A "BLOCK ID MISMATCH" OR A "BUFFER COMPARE FAILED" IS NOT ALLOWED
2. ANY COMBINATION OF ERRORS EXCEEDING 99 TOTAL IS NOT ALLOWED
3. BRACKETS INDICATE A RANGE OF NUMBERS THAT CAN OCCUR

123

QUICK DEBUGGER TEST PROCEDURE

The following equipment will be needed for this procedure

- Apple ///
- Profile Interface PCB for the Apple ///
- Profile Interface Cable
- Quick Debugger Software Version E00.07 (APN# 889-0004)
- Silentype Printer

The Quick Debugger program reads the spares table from the Pro-File and prints the original sector location of every spared sector listed in that table.

1. Install the Profile Interface PCB in slot 1 of the Apple /// and connect the Profile Interface Cable from it to the Profile.

2. Ensure that there is a Z8 with standard system ROM installed on the Controller PCB.

3. Turn on the Profile and wait 2 minutes for the Pro-File to complete its Scan sequence.

4. Connect the Silentype to Port A on the back of the Apple ///.

5. Boot the Quick Debugger program.

Observe: The Quick Debugger command prompt line is displayed as shown below.

```
COMMAND (B,F,P,Q,R,S,T,V,W,?)=>
```

6. Type <S> to select the Get Status command.

Observe: a. The line shown below is displayed next to the command prompt line:

```
STATUS=GET STATUS BLOCK 00 00 00 00  00 00 00
00
```

b. The Ready LED goes out as the interrupter arm on the HDA goes to track 77 to read the spares table.

c. A new command prompt line is displayed:

(continued on the next page)

124

7. Type <P> to select the Print Status command.

Observe: The line shown below is displayed on the CRT.

PRINT STATUS TO SCREEN OR PRINTER (S/P)

8. Type <P> to select the printer.

Observe: The Silentype prints out as shown below.

```
CONTROLLER VERSION - 03.98


SPARED SECTOR
TOTAL   00

  LIST       CYL HD SECT


BAD BLOCKS
TOTAL   00

  LIST       CYL HD SECT
```

(End of this procedure.)

125

DSKDBG - BLOCK DEVICE DEBUGGER VERSION E00.17

TABLE OF CONTENTS

126

PART 1 INTRODUCTION

The Formatter/Debugger program is provided as a service aid for troubleshooting block I/O devices such as the ProFile disk drive. It is a program that allows you to directly communicate with a block I/O device and exercise it through a series of unique commands or build simple test sequences that will assist in debugging the device.

It is a particularly helpful tool for servicing devices that have been returned because of a malfunction, etc.

For example, the program can be used to detect an error and then by using the loop on error function, you can use an oscilloscope to analyse the circuit that caused the error.

You can use the program to service a drive containing either the Format/Debugger ROM Z8 or the Standard System ROM Z8. (The Standard System ROM Z8 is the operational ROM in the Z8 that is shipped in the disk drive.) The command summary, part 3, specifies which type of ROM must be in the drive to use a given command.

PART 1.1 HOW TO USE THIS PROGRAM

To use this debugger properly, you should be aware of some of the design concepts. In designing the user interface (command structure) it was decided to use single character commands (explained later). This permits you to type in the command very rapidly and use options (which are not always required) to modify certain test variables.

The most used test variables are the three byte logical block that is treated as either a 24-bit number or as three 8-bit numbers. As a 24-bit number, the variable represents a standard logical block with a decimal range of 0-16 million blocks (Hex 000000- FFFFFF). As three 8-bit numbers, each variable has a range of 0-255 (Hex 00-FF).

The program makes no assumptions about the use of the test variable; it just gives you two ways to talk to it.

You can consider it as a large (24 bit) number or as three smaller (8 bit) numbers. This variable is sent to the disk exactly as a 24-bit number which the firmware decides how to interpret.

You, the user, must know how the firmware will react to understand what the firmware is doing with the block number.

127

PART 2: TYPICAL COMMAND EXAMPLES

NOTE: Data shown in <X> is meant to be typed.
Data shown in [] or ( ) is optional.

To help make the concept of this program more clear, here are some examples that show what the more common commands do.

<R> (Read)

This command requests data to be read from the unit being tested. It passes the command and the BLOCK variable to the unit and transfers the data from the unit to the input buffer (see the display command, part 2.2.3, for information on displaying the input buffer).

R by itself does not change the BLOCK variable; it uses its current value. However, you can change the block variable by adding a modifier to the command in the form of a number.

This number is normally treated such that leading zeros are assumed, thus 0, 00, 000, 0000, etc. all produce the 24-bit value 000000. For example:

R0 or R00 or R000000, etc. (read block 000000)
R13 or R013 or R00013, etc. (read block 000013)

The above example is applicable only when using the Standard System ROM Z8. Notice that the BLOCK variable sent is either 000000 or 000013.

The number 13 is the same as 0013 or 00000013 and produces the value 000013 in the BLOCK variable.

NOTE: The number 13 in the example is a HEX number, not a decimal number.

13 Hex = 19 Decimal

Remember that the firmware in the disk drive being tested determines how the BLOCK variable will be treated. Normally it is treated as a logical block.

In the example on the following page, let's change the command to show how it would be used with the Format/Debugger ROM Z8:

128

For example: <RT1H2S3> or <RT01H02S03>, etc. (read block 010203)

Notice that the command now has 3 modifiers in the form of T and a number, H and a number, and S and a number.

The modifier T (Track) references the first 8-bits of the BLOCK variable, the H (Head) references the second 8-bits, and the S (Sector) references the third 8-bits.

Note that they are separate modifiers and can be used independently.

For example:

If BLOCK = 000000, then the command **<RT5><RETURN>**
will set BLOCK = 050000

If BLOCK = 050000, then the command **<RH3><RETURN>**
will set BLOCK = 050300

If BLOCK = 050300, then the command **<RT16S9><RETURN>**
will set BLOCK = 160309

Note that the last command in the example happened to affect the first and third bytes. Had the last command been RS9 then the BLOCK would have been 050309.

When all three modifiers are used, they must be in the T, H, S sequence.

REMEMBER! The numbers used are always Hexidecimal numbers.

<W> (write)

This command requests data to to be written to the unit being tested.

It passes the command and the BLOCK variable to the unit and transfers the data to the unit from the output buffer (see the buffer fill command, part 2.2.1, and the display command, part 2.2.3).

W by itself does not change the BLOCK variable; it uses its current value. However, as was the case with a read command, the BLOCK variable can be changed by adding a modifier.

129

For example, the commands W0, W03H2561, W#9, etc. have the same effect on the BLOCK variable that they do in the Read command.

<+> (plus)

This command increments the BLOCK variable. + by itself increments the variable by 1. For example, if the BLOCK variable was 00001F, then after + it will be 00000220 (Note the hex numbers). +3 will increment the BLOCK by 3 each time. +3T will increment the first 8-bit variable by 3 each time. For example, if the BLOCK was 050311, then +3T will change it to 080311. H and S work the same way to modify the second and third groups of 8-bit variables.

To provide wraparound and carry, the + command will wrap a 24-bit number at FFFFFF. It will wrap T at 97 (97 increments to 0) and will set H an S to 0 to provide full wrap. H will wrap at 3 (3 goes to 0) creating a carry to T and S will wrap at F (F goes to 0) creating a carry to H. Future enhancements will be to find out what the maximum block count for a device is and then wrap the 24-bit number at that count.

<D> (Display)

This command displays the contents of the input or output buffers.

D by itself will display the input buffer.
DI is the same as D.
DO will display the output buffer.
DS will decode the contents of the input buffer as extended status information.

The form of the display is the same as that for an APPLE monitor memory dump. It will display the first 256 bytes of the buffer and pause so that you can study the values.

At the bottom of the screen it asks for <ESCAPE> to terminate, <RETURN> to quit, ANY other key to continue.

<ESCAPE> stops further display and cancels the rest of the command line effectively stopping processing and returning control to the user.

<RETURN> quits the display command and causes the program to go on to the next command.

130

Pressing any of the other keys, except the space bar, will allow the program to display the next 256 bytes of the buffer.

Pressing the space bar will allow you to display the buffer line by line.

CAUTION: If you just keep hitting a key to see the next 256 bytes you will see first the buffer you requested and then memory above it. There are areas of ram above the buffers that will, when displayed cause your screen to get VERY sick and you will have to either re-boot the program or be very familiar with APPLE III hardware and monitor.

Just in case you are not familiar with the monitor memory display, here is an example. (The monitor doesn't display the descriptive information above the dashed line.)

```
     Hex
Address  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
-------------------------------------------------------
8000:  00 1A 23 92 FF FE A5 2F 33 34 00 00 00 00 00 00
8010:  12 11 34 87 FE 44 00 00 3E 4C 00 20 A9 0F C5 00
etc.
```

131

PART 2. COMMAND STRUCTURE

Each command consists of one or more characters and modifiers in command groups.

Each group is separated by one or more spaces.

The command line is one or more command groups.

There is enough space in the command buffer for 255 characters.

Multiple command lines are possible by use of the Macro Add command.

PART 2.1 COMMAND PROMPT

The program prompt consists of the following two lines:

```
S1 D1
COMMAND (B,C,D,F,G,H,I,L,M,N,O,P,Q,R,S,T,V,W,X,+,-,//,   ?)) =>
```

The first line contains two flags, S1 D1, that tell which slot and drive the program is set to test.

The second line of the prompt consists of the first letter of each valid command. A list of the commands is always available by typing H or ? and pressing <RETURN>.

NOTE: After typing a command and pressing <RETURN> to execute it, you can press <ESCAPE> to terminate the command.

However, if you wish to review the command that you just executed for the purpose of changing it, etc., you can simultaneously type <CONTROL> <A> instead of <ESCAPE>.

132

PART 2.2 DETAILED COMMAND DESCRIPTIONS

Following are descriptions of some of the commands that require more detailed explanations than others. In depth examples of some of the more commonly used commands such as Read, Write, Increment, and Display were provided in part 1.2. For an overall summary of all commands, refer to part 3.

PART 2.2.1 BUFFER FILL

<B>[num]<**RETURN**>
- where num is an optional 16 bit hex number
(eg.,B0123)

The buffer fill command is used to fill the output buffer with a specific data pattern. It can be used with either the Format/Debugger ROM Z8 or the Standard System ROM Z8 installed in the drive. To use the command, type B followed by a number which will be treated as a 16 bit (2 byte) pattern.

For example, the command

<**R0**><sp><**B1234**><sp><**W0**><sp><**B45**<sp><**W1**><**RETURN**>

would read logical block-0, fill the output buffer with 12341234..., write block-0, fill the output buffer with 00450045..., and finally write this data to block-1.

PART 2.2.2 CREATE SPARE

<**C**><**RETURN**> ---- the C-command must be the only command on the line.

The create spare command is used to force a peripheral device to transfer a logical block to a new (spare) physical location on the device. The command will then request the block number to be spared and will then request confirmation from the user. If confirmed, the logical block will be spared by the device controller. This command should only be used when the Standard System ROM Z8 is installed in the drive.

Using this command will allow you to fix flaky blocks found during testing. The user will be asked for confirmation of this command. Respond with Y (Yes) or N (no).

NOTE: Do not attempt to use the create spare command when the Format/Debugger ROM Z8, version D3.11 or earlier, is installed in your drive.

133

PART 2.2.3 DISPLAY

<D>[I,O,S]<RETURN> - where I is input buffer, O is output buffer, and S is the extended status information in the input buffer.

The display command can be used with either the Format/Debugger ROM Z8 or the Standard System ROM Z8 installed in the drive. It is used to display the contents of the input or output buffers.

These buffers occupy the following Hex locations in the computer's memory:

input buffer ----8000H - 8213H
output buffer ---8300H - 8523H

The display will show 256 bytes at a time, scrolling to the next 'page' after each key press.

To end the display and continue executing the command line, press <RETURN>.

To abort the command line press <ESCAPE>.

With the display on the screen, you can press the spacebar to cause a scroll to the first line of the next 'page'. Then, each time you press the space bar, the display will scroll one line forward.

The Display Status command works only slightly differently in that it decodes the input buffer into Extended status info about the spares and bad blocks and it will not stop until it is finished or until the space bar, <RETURN> key, or the <ESCAPE> key are pressed.

REMEMBER! The Display Status command will decode anything found in the input buffer so be sure to use the Get Status command first to make sure that the information is valid.

Before using the Display Status command with the Format/Debugger ROM Z8 installed, you must initialize the spare tables (see part 2.2.6) and then use the Get Status command, otherwise the display will contain garbage.

With the Standard System ROM Z8 installed, you only have to issue the Get Status command before displaying the extended status of the input buffer.

134

Here are some additional tips about using the Display Status command.

- After typing D[S] and pressing <**RETURN**>, press the space bar to stop the listing. Press the space bar again each time you wish to step through another line of the listing. Pressing any other key will cause a fast scan of the listing.

- Pressing <**ESCAPE**> will cancel the rest of the display function.

- Pressing <**RETURN**> will terminate the spare sector listing and start the bad block listing.

PART 2.2.4 FORMAT

<**F**><**RETURN**> ---- this command must be the only command in the command line.

The Format command is used with the Format/Debugger ROM Z8 installed in the drive. It is used to erase all old data from memory and lay down a new pattern of address and data.

After formatting a drive with the Format/Debugger ROM Z8 installed, you can type D[I] to get a list of the defective blocks. The list will end with FF FF FF FF. Refer to the documentation provided for the F/D ROM for further details. **WARNING:** This command is very dangerous and should only be used if damage to the address headers has occurred and only after every reasonable attempt has been made to recover other data from the device. The user will be asked to confirm this command. Respond with <**Y**> (Yes) or <**N**> (No).

Remember! This command will erase all previously recorded data.

Following is an example of the error message that will be displayed if
you attempt to format a drive with a Standard System ROM Z8 installed.

			I/O			ProFile				
	Block	R1	R2	R3	R4	S1	S2	S3	S4	S5	S6
FORMAT S1 D1	00 03 5F	00	00	00	00	55	55	55	55	00	00

The 55's in this example are the error indicators.

135

PART 2.2.5 GET STATUS

<G><RETURN>

The Get Status command works only with the Standard System ROM 28. It is used to obtain extended status information from the drive and place it in the input buffer.

(You will need to execute the Display command to view the contents.)

The Get Status command makes sure that the status you are going to view is valid.

When using a ProFile you can also get the status by reading block FFFFFF.

PART 2.2.6 INITIALIZE SPARE TABLE

<I><RETURN> ==== this command must be the first command in the command line.

The Initialize command is used by the ProFile Format/Debugger ROM 28 to setup the spare tables in the ProFile drive.

After the Debugger ROM has formatted the disk, the entire disk is available to read from or write on so that certification of the entire disk is possible.

After the spare table sectors have been certified, the tables need to be initialized to allow the controller ROM to work properly.

The Initialize command is not used when the Standard System ROM 28 is installed in the drive.

WARNING: This command is potentially dangerous as it effectively erases any old spared table data and if used incorrectly will cause the loss of valuable data on the ProFile.

The user will be asked to confirm this command. In response, type <Y> (Yes) or <N> (No):

136

PART 3 COMMAND SUMMARY

Following is a summary of all the commands that can be used with this program:

BUFFER FILL **B[<RETURN>] or B[num]<RETURN>**

Fill the output buffer.
where num is a 16 bit hex fill number
used with both Formatter/Debugger and Standard System ROM Z8s

CREATE SPARE **C<RETURN>**

Force the drive to spare the specified block.
must be the only command on the command line
used only with Standard System ROM Z8

DISPLAY **<D><RETURN> or <D>[I]<RETURN> or <D>[O]<RETURN> or<D>[S]<RETURN>**

Display the I/O buffers.
used with both Formatter/Debugger and Standard System ROM Z8s

FORMAT **<F><RETURN>**

Format the device.
This is a dangerous command
used only with Format/Debugger ROM Z8

GET STATUS **<G><RETURN>**

Get extended status information.
Remember! When using the ProFile, RFFFFFF (read block FFFFFF) also returns status information
used only with the Standard System ROM Z8

137

HELP <H><RETURN> or <H>[E]<RETURN> or
<H>[E2]<RETURN> or <H>[char]<RETURN>

Print a list of commands, or errors, or a detailed description of each command.
Where E is to display errors when the Standard System ROM Z8 is installed.
Where E2 is to display errors when the Format/Debugger ROM Z8 is installed.
Where char is any legal command shown in this command summary.

INITIALIZE SPARE TABLE <I><RETURN>

Clear the spare block table.
This is a dangerous command
must be the first command on the command line
used only with Format/Debugger ROM Z8
requires confirmation (Y or N)

LOOPON <LPF>[S,H,D]<RETURN>

Send Loop on Format commands to the firmware.
where S = sector marks
H = address headers
D = data field
LPF can be followed by any combination of S, H, and D
executed following a Read Track command to loop on format of the current track (eg., RTT reads the T byte of the block variable to determine the current track)
used only with Format/Debugger ROM Z8

MACRO <M>[A(0-9),C,L,0-9]<RETURN>

Use macro functions (alternate command lines).
where A = add current line to macro table
C = clear all macros
L = list macros
used with either Formatter/Debugger or Standard System ROM Z8
not effectively implemented at this time

138

N,O Not implemented yet

PAUSE <**P**>[<**RETURN**>] or
<**P**>[A(num),C,(num),E(num),N(num)]<**RETURN**>

Pause and wait for user.
Where A = any error/nonerror
C = clear any error/nonerror
E = on error
N = on no error
(num) is a 16-bit mask of S1 S2

Note: All four digits of the 16-bit mask must be turned on

used with either Formatter/Debugger or Standard System ROM Z8

QUIT <**Q**>[<**RETURN**>]

Return to calling routine (This module is a subroutine).

READ <**R**>[(num),T(num),H(num),S(num)]<**RETURN**>

Read a block
where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

Remember! The firmware of the device tested determines how the number is treated. See the following example:

T H S - Formatter/ Debugger ROM
ProFile 00 00 00
Hi Mid Lo - Standard System ROM Z8

>O means read to the output buffer (Eg., R23>O)
used with either Formatter/Debugger or Standard System ROM Z8

139

SCAN <S><RETURN>

Order the firmware to scan the entire disk (read only).
Requires confirmation (Y or N)
after scanning, use D[I] to get list of any bad blocks
found during scan
used only with Format/Debugger ROM Z8

TURN OFF STEPPER <T><RETURN>

Turn off the power to the stepper motor,
used with Format/Debugger ROM Z8 but not normally used
with Standard System ROM Z8

V Not implemented yet

WRITE <W>[(num),T(num),H(num),S(num)]<RETURN>

Write to a block (same format as read).
where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

Remember! The firmware of the device tested determines how the number is treated. See the following example:

T H S = Formatter/ Debugger ROM
ProFile 00 00 00
Hi Mid Lo = Standard System ROM Z8

<I means write from the input buffer (eg., W23<I)
used with either Formatter/Debugger or
Standard System ROM Z8

140

XECUTE <X><RETURN> or X[E(num)]<RETURN>

Execute the current command line again.
where E = execute the line to this point on error
num = 16 bit error mask of S1 S2
X[E] will execute a function over and over again, if an error has occurred, up to the point where the error occurred

+ <+>[(num),T(num),H(num),S(num)]<RETURN>

Increment the block number.
where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

- <->[(num),T(num),H(num),S(num)]<RETURN>

Decrement the block number.
where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block
the - command does not decrement properly for H and S modifiers

/ </>[S(num),T]<RETURN>

Choose options (/H for help).
where S = slot set
T = translate current block to cylinder/head/sector

? Help.

141

PART 4 NEW FEATURES

The following new features have been developed since the release of DSKDBG V-E00.16 and are included in DSKDBG V-E00.17.

* The Help errors command has been slightly modified. <HE> or <HE1> provides help for the Standard System ROM Z8 and <HE2> returns help for the Format/Debugger ROM Z8.

* The Help command lists the form of the extended help commands.

* Display now has a command to allow the decoding of the status table into the version, spare list, and bad block list.

DS = display status info.

* Use <ESCAPE> or <RETURN> to terminate a long list of spares or bad blocks.

* An additional modifier is available for read/write commands that allows you to specify which buffer you are reading to or writing from. For example, a read command followed by >0 means read to the output buffer. A write command followed by <I means write from the input buffer.

PART 5 TYPICAL PROBLEMS AND HOW TO AVOID THEM.

* The + (Increment) command doesn't work.

This is to date the most common problem. Usually the command line will look something like

R0 PE + X

What you were trying to do is sequentially read thru the disk starting at block 0. What you said was

Read block 0 (block = 0), Pause on error,
Increment block (block = 1), Repeat line,
Read block 0 (block again = 0) .....

To fix this problem just do one line to set block 0 and then go on to the next line to do the Read, Pause on error etc.

142

For example:

<R0><RETURN>
<R><sp><PE><+><X><RETURN>

* Display Status Won'T Work Or Won't Stop

The display status command allows you to display the input buffer and make assumptions about what the data means.

For it to work, the Get status command must successfully read the status informtion from the drive.

If the input buffer has garbage in it, the Display Status command may go round and round trying to display all the spared sectors and bad blocks.

To stop the command when it is caught in a loop or is listing a bunch of bad blocks, simply press the <RETURN> key or the <ESCAPE> key. The functions of the other keys are the same as for all other display commands.

PART 6 KNOWN BUGS IN VERSIONS E00.17 AND E00.16

The Macro command has 3 known bugs.

The first occurs when a macro is added to the list and there is already one there of that number. The effect is to make both of them disappear.

The secondbug occurs when the macro is invoked on a line with the X command (eg., M1 X). This disables the <ESCAPE> key so the program can only be halted by re-booting.

The third bug also involves the X command. When a macro is added (eg., R MA1 X), it will keep asking to delete the old macro.

The - command doesn't decrement properly for H and S modifiers

143

PART 7 STATUS BYTE DESCRIPTIONS

This part provides a bit-by-bit description of the four status bytes available with the Standard System ROM Z8 or the Format/Debugger ROM Z8.

PART 7.1 STATUS BYTES WITH STANDARD ROM V-3.98 INSTALLED

STATUS 1

7 = 1 iff ProFile did not receive 55 to its last response

6 = 1 iff write or write/verify was aborted because
>5332 bytes
iff data were sent
iff ProFile couldn't read its spare table

5 = 1 iff host's data is no longer in RAM because ProFile updated its spare table

4 = 1 iff SEEK ERROR — unable in 3 tries to read 3 consecutive headers on a track

3 = 1 iff CRC error (only set during actual read or verify of write/verify, not while trying to read headers after seeking)

2 = 1 iff TIMEOUT ERROR (couldn't find header in 9 revolutions — not set while trying to read headers after seeking)

1 = N/A

0 = 1 if operation unsuccessful

(continued on the next page.)

144

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/A

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/A

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/A

3 = N/A

* 2 = 1 if ProFile was reset

* 1 = 1 if ProFile gave a bad response

* 0 = 1 if parity error

* These bits are set by the SOS ProFile driver and are not used by the Dskdbg ProFile.IO.

STATUS 4

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

145

PART 7.2 STATUS BYTES WITH FORMATTER/DEBUGGER ROM FD3.98 REV 11 INSTALLED

STATUS 1

7 = 1 if Profile did not receive 55 to its last response

6 = 1 if no index found during formatting

5 = 1 if no sector mark found during formatting

4 = 1 if SEEK ERROR = unable to read 3 consecutive headers on a track (one try only)

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/A

0 = 1 compare error on a write compare

STATUS 2

7 = 1 if ProFile has been reset

6 = 1 if track number invalid while reading or writing a sector

5 = N/A

4 = N/A

3 = N/A

2 = N/A

1 = 1 if seek to wrong track occurred

0 = N/A

(continued on the next page)

146

STATUS 3

D7, D6, and D5, along with D4 and D3 from STATUS 1, tell why a write compare operation failed.

	D7	D6	D5
write timeout	0	0	0
read timeout	1	0	0
read CRC	1	1	0
data compare	1	1	1

STATUS 4

7 - 0 = the number of errors encountered when formatting data fields or scanning the disk.

147

SECTION 3

APPENDICES

148

SECTION 3 TABLE OF CONTENTS

149

HOW TO USE THIS SECTION

This section contains general information on the functional operation of the 4 modules in the Pro-File, and its firmware operation.

It is designed to be referred to from the other sections in this manual, but you may wish to use it to help you better understand Pro-File operation.

If you are unfamiliar with Winchester disk drives, then a good place to start would be with the Pro-File HDA description in this section.

Then if you would like general information on each of the 4 Pro-File modules read the "Overview of the Pro-File".

150

OVERVIEW OF THE PRO-FILE

The following description will give you an overview of the technical details of the ProFile. It is intended to be a summary only, and is not a detailed explanation of the engineering aspects of the ProFile.

General Information on the Pro-File

The intelligent Controller (Controller PCB), which is built into ProFile is continually checking the operation of the disk. ProFile performs many operations to assure that the user will never see a problem.

These operations start the moment ProFile is turned on. After power-up, ProFile does a Scan operation of the entire disk surface, and checks for any errors.

Upon any data read error, an extensive analysis of the error is performed to determine whether a media error exists. If that is the case, the data will be moved to a spare sector of the disk. That part of the disk with a media error will no longer be used, it is "spared".

In most cases, the error recovery routines in ProFile will be able to extract the data even from a bad sector. The recovery operation includes more than 300 retries under various conditions. Maps of the bad sectors are redundantly recorded on ProFile (these maps are called spares tables), so that an error in the map will not cause a problem in operation.

The ProFile moves the heads to a "parking position" off the data zone after three seconds of no activity. This prevents the loss of data if, for instance, the ProFile is dropped or jarred. ProFile constantly checks for errors during operation.

After any change in tracks, ProFile verifies that the operation has been performed correctly. ProFile also checks that the heads are positioned accurately on a track before any read or write operation is performed.

Unless the system requests that the ProFile do otherwise, data is always verified after a write operation. In all these cases, ProFile will correct the problem so that no errors occur.

151

Data can be transferred from the ProFile to the system at up to one Mbyte per second DMA rate. Data is interleaved at a 5:1 ratio, which allows three 512 byte sectors to be transferred on each rotation. MFM encoding allows the maximum data storage capacity with low formatting overhead.

Note: 1. For more detailed information on Scan operation, and error recovery routines refer to the firmware routine's functional description in this section.

2. For more information on the format (including spares tables), used for the Pro-File HDA refer to the Pro-File HDA description in this section.)

What is ProFile?

ProFile is a Winchester technology hard disk drive designed to operate with a host computer. It has a formatted storage capacity of 5 Megabytes, which is essentially the same as 35 floppy disks (the Apple III disk drive uses floppies that have a capacity of 140K Bytes).

Functionally, the ProFile consists of four major modules:

1. The Controller PCB
2. The Analog PCB
3. The Power Supply
4. The Hard Disk Assembly (HDA) with motor control PCB

The discussions on the following pages describe the general function of these modules.

152

1. The Controller PCB

Functionally, the Controller provides communications with the host computer, provides signals to read and write serial data on the disk, moves the heads to the proper track, and monitors error conditions.

The Controller consists of a Z8 microprocessor, 2K bytes of RAM, error detection logic, and read/write control logic.

The Z8 provides an intelligent interface to the host computer. High level commands, such as read, write, and status, are generated by the host computer to through the Pro-File interface cable to the Z8. The Z8 uses the buffer area in RAM to temporarily store any data being read from or written to the disk.

The Controller also interfaces to the Analog card to pass head control information to it. In this way the Controller determines when read/write operations will take place.

The Z8 controls disk operations in the HDA. Basically it sends the stepper motor in the HDA step pulses to move the heads from track to track (such moves are called "seeks"), selects one of four read/write heads, and enables the different functions (i.e. sync, clock data, etc.) on the Analog PCB to perform the read or write operation called for by the host.

Read/Write functions are performed by the read/write logic on command from the Z8. This logic in turn controls the parallel to serial data conversion when writing, and the serial to parallel data conversion when reading.

To ensure the proper transfer of data, the Controller does a CRC (cyclic redundancy check) of the serial data. If an error occurs, the Z8 will automatically perform an error recovery routine and try to relocate the data onto a different section of the disk. The sector causing problems will be removed from use to prevent future errors.

After 3 seconds of no communication with the host, the Z8 on the Controller PCB will move the head to a Park position on the disk (track 155 a non data track), to prevent accidental damage to the disk surface in the event of a hardware failure (such as power loss). Once in Park position for 0.75 seconds, the Z8 removes power from the stepper motor to prevent heat build-up in the drive.

The READY light on the front of the ProFile is lit whenever the Controller is idle (not busy).

153

2. Analog PCB

The Analog PCB serves as the interface between the Controller and the Head Disk Assembly (HDA).

The Analog PCB consists of a data encoder, a data decoder, write driver, head select logic, automatic gain control (AGC) preamplifier, read detector, phase lock loop (PLL), and sector detector.

The head select matrix selects one of the four heads for a read or write operation in response to control signals from the Z8 on the Controller PCB.

The ProFile has two fixed disks in its HDA, and there are two heads for each disk (one for each side, since the disks are double sided). Thus you have a choice of four heads, depending on which section of the disk you are trying to access.

It is not necessary for the host computer to know which section of the disk it is trying to access; the Controller and the Analog PCB take care of that.

During a write operation, the serial data is encoded using a technique known as Modified Frequency Modulation (MFM).

During a read operation, the AGC circuit amplifies the low level head signal (.6 to 2.0 mv) to a fixed signal level (1.0 volt). The read detector simply shapes the signal so that it appears in a standard fashion.

The PLL and data decoder then convert this signal back into serial data, which is passed to the Controller, which in turn converts it to parallel data and passes it to the host computer.

High speed, low noise ECL (emitter coupled logic) provides wide margins for the Analog electronics PCB of the ProFile.

When the disk is initially formatted, the sector boundaries are written to the disk (this is done by removing all read signals from certain sections of the media). During the read operation, the sector detector looks for these areas of no read signal, and signals the Controller that a sector boundary has been found.

154

3. Switcher Power Supply

The power supply provides the +5VDC, and ±12VDC needed by the ProFile for operation.

The supply also contains monitoring circuitry to detect a power failure. This monitoring circuitry senses a power failure before the internal DC voltages drop.

If a power failure should occur the monitoring circuitry uses a signal called Power OK (POK) to reset the Z8 on the Controller PCB.

This allows the intelligent Controller in ProFile to prevent any data loss if the power fails or is turned off accidentally. The system will not begin any operation until the power is on for at least one second.

Once a failure is detected, the head current is shut off to prevent the accidental writing of false data that would otherwise occur if a write operation were in process when the power failed. The power supply is completely shielded to eliminate the effects of electro magnetic radiation.

155

4. Head Disk Assembly (HDA)

The ProFile HDA is a random access storage device with two non-removable 5 1/4 inch discs as storage media. Each disk surface employs one movable head to service 153 data tracks. The total formatted capacity of the four heads and surfaces is approximately 5 Megabytes (16 sectors per track, 512 bytes <A/// format> or 532 bytes <Lisa format> per sector, and 612 tracks).

High reliability is achieved through the use of a band actuator and open loop stepper head positioning mechanism. The read/write heads are mounted on a ball bearing supported carriage which is positioned by the band actuator connected to the stepper motor shaft.

Mechanical and contamination protection for the heads, actuator, and discs are provided by an impact resistant aluminum enclosure. A self contained recirculating system suplies clean air through a 0.3 micron filter.

A special spindle pump assures adequate air flow and uniform temperature distribution throughout the head and disk area.

Thermal isolation of the stepper and spindle motor assemblies from the disc enclosure provides significantly greater "off track" margin (temperature changes are less likely to cause read errors).

A brushless DC drive motor rotates the spindle at 3600 RPM. The spindle is driven directly with no belt or pulley. The motor and spindle are dynamically balanced to insure a low vibration level.

Depending on the revision of Motor Control PCB on the HDA a physical brake mechanism, or a dynamic brake (involving reverse biasing the coils on the disk motor) may be used to provide a fast stop to the spindle motor when power is removed. (continued on the next page)

156

4. Head Disk Assembly (HDA) (continued)

The recording media consists of a lubricated thin magnetic oxide coating on a 130 mm diameter aluminum substrate. This coating formulation, together with the low load force, low mass Winchester type "flying heads", permits reliable contact start/stop operation.

The HDA (hard disk assembly) can be operated from 10˜C to 60˜C/. This wide operating range, combined with a 75% efficient switching power supply allows ProFile to operate from 10 to 40˜C in still air without a fan.

(For more information on the Pro-File HDA format (including spares tables), used for the Pro-File HDA refer to the Pro-File HDA description in this section.)

157

FIGURE 1

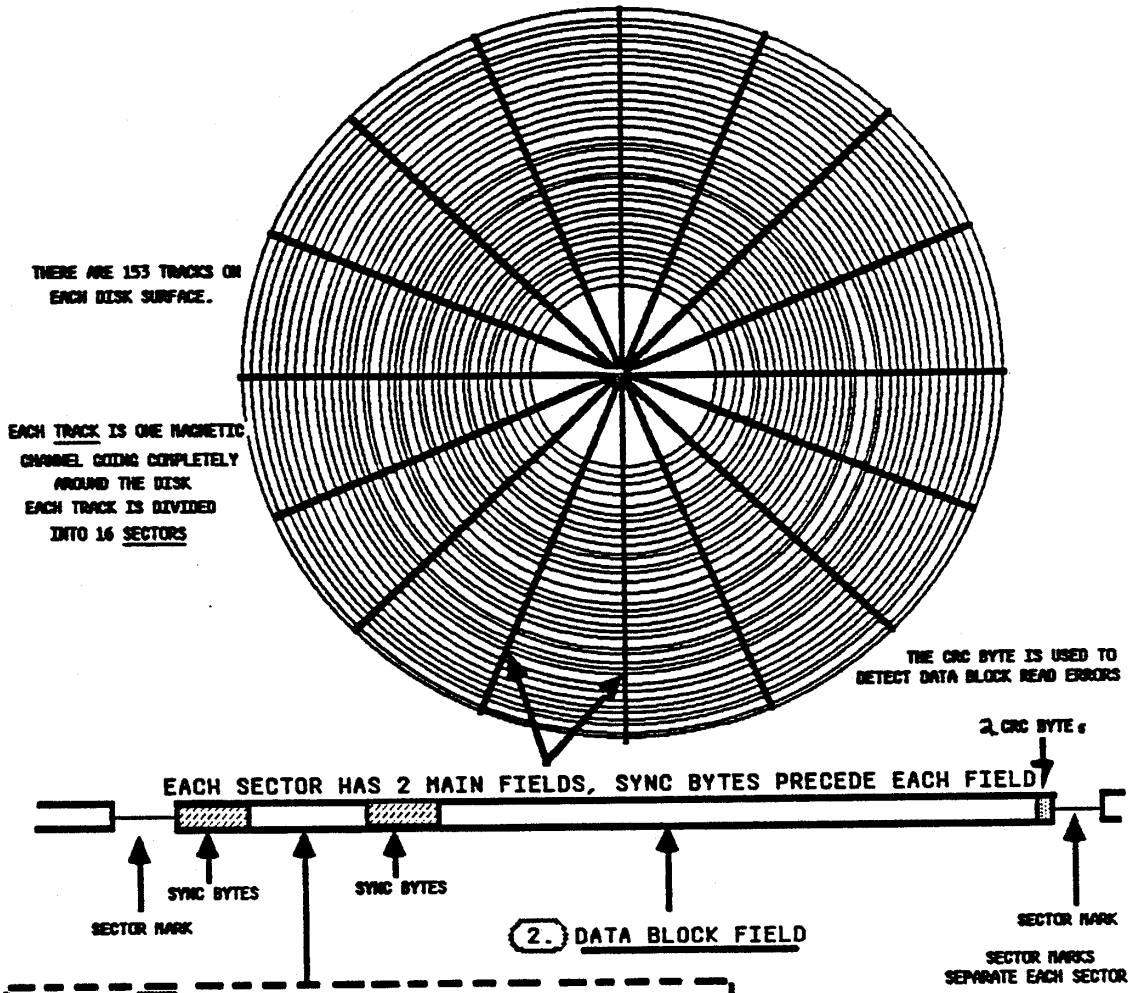


FIGURE 2

158

PRO-FILE HDA DESCRIPTION (THE HARD DISK ASSEMBLY)

PHYSICAL DESCRIPTION OF THE HDA

The HDA contains 2 rigid disks, which are constantly rotating at 3600 RPM. Each disk has 2 surfaces, (top and bottom), making a total of 4 surfaces numbered 0, 1, 2, and 3 (see figure 1).

The disk surfaces are covered with a metallic film which is very easily magnetized. In the beginning, when the disk is first made, there is no magnetic information on it at all.

No information will be put on the disk until its magnetic areas have been specially arranged, or **formatted** with a Format program run from the Host computer system. A discussion of the features common to formats used for the Pro-File HDA is explained later in this discussion.

Each surface has a device called a head, which converts electrical impulses into magnetic impulses and vice versa. The heads are numbered after the surfaces they cover, that is, 0, 1, 2, and 3.

During a write operation, binary data in the form of a bit stream is written on the disk surface by the head. As the disk surface rotates under the head, each pulse in the bit stream will cause a small area on the medium (disk surface) to be polarized.

Extreme closeness of the head to the medium (in the Pro-File's case 20 microinches) is a distinguishing characteristic of Winchester type disk drives.

The head rides (flies) on the thin cushion of air between it and the media (something like an airplane wing). If particles as small as a piece of dust are allowed into the head chamber, they can cause the head to lose its flight stability. The head may then become erratic causing it to crash into the media, and destroy some areas on the disk.

For this reason, the HDA is kept in an air-tight enclosure with a small filtered aperture. This allows atmospheric pressure equalization while preventing potentially dangerous particles from gaining access.

During a read operation, the head senses the magnetic areas on the disk surface as they pass under the head, and produces an electrical impulse for each change in their polarity.

159

FIGURE 1

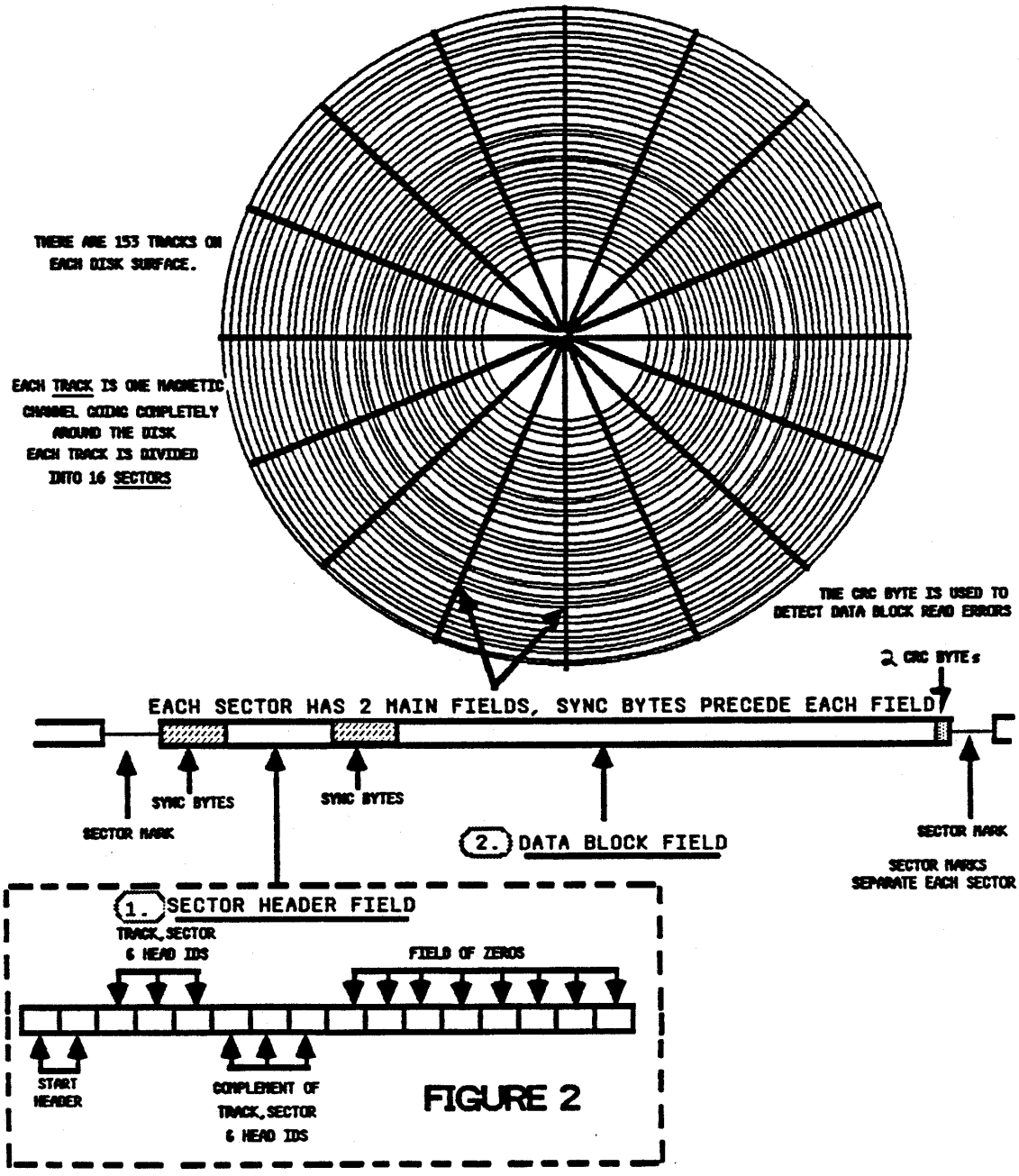


FIGURE 2

160

DESCRIPTION OF THE PROFILE HDA FORMAT

Now you know how binary data is written onto and read from the disk. But how does the drive store different sets of data (files) and keep them separate?

To do this, the drive needs a way of writing different files to different areas of the disk, and then finding them when it needs to read them.

To understand how the drive does this, you need to understand **steps, tracks, blocks,** and **sectors.**

Each head is mounted in such a way that it may move in to the center and out to the perimeter of the disk surface as the disk rotates beneath it.

A write or read operation will never take place while the head is moving, but only while the head is stationary and the disk is moving under it.

The heads move in very exact increments called **steps.** On the Pro-File the head must take 153 steps to cover the entire area on the disk where data may be stored. The head may stop after any step to read or write data. The 153 possible positions of the head determine the 153 magnetic channels, or **tracks,** on which data can be stored (refer to figure 1).

The selection of which track the head is positioned over is controlled by the firmware program in the ROM on the Z8 microprocessor on the Controller PCB (the particular firmware routine used for this purpose is called the **Seek Routine** and is discussed in the Firmware Routine descriptions in this section).

The Z8 performs track selection by sending step pulses to the motor controlling the head. The number of step pulses the Z8 sends to the stepper motor determines how far the head will move and so which track it will be over.

161

DESCRIPTION OF THE PROFILE HDA FORMAT (continued)

The firmware program in the Z8 also controls when a read or write operation occurs. When writing, the bit stream to the head travels from the Host (Apple ///, or Lisa), to the Pro-File's electronics, to the head; and when the bit stream is read from the disk it goes back to the Pro-File's electronics, and then to the Host.

There are 612 tracks on the HDA (153 tracks on each disk surface, and 4 disk surfaces), and approximately 8,200 bytes on each track.

If track selection were the only method the Z8 firmware had to control the location of stored data, it would have a maximum of 612 very large storage places (8,200 bytes apiece). Since most files are considerably smaller than 8,200 bytes, this would be very inefficient.

So to make data storage more efficient, the host computer breaks all data up into segments called **data blocks.** If the Host is an Apple /// the data blocks are 512 bytes long, and if the Host is a Lisa they are 532 bytes long.

The Pro-File stores these **blocks** into physically addressable areas on the track called **sectors.** There are sixteen sectors on each track (refer to figure 1), so each track on the Pro-File can hold 16 data blocks.

The sectors are separated from each other by small blank areas of no writing called **sector marks.** The Pro-File uses these sector marks to detect when one sector ends and a new one begins.

Each sector can be divided into 2 fields: the first field is called the **sector header,** the second field follows the sector header and is called the **data block** (see figure 2).

(Continued on the next page.)

162

DESCRIPTION OF THE PROFILE HDA FORMAT (continued)

Each field in the sector is preceded by sync bytes to allow the Pro-File electronics to synchronize to the data in the field before attempting to read one.

The sector header field contains 16 bytes: 2 are "start header" bytes; 3 bytes contain track, sector, and head ID's; 3 bytes contain a complement of the track, sector, and head ID information; and the remaining 8 bytes are all 0's.

The data block field consists of the actual data from the Host computer with 2 CRC bytes at the end (a CRC (Cyclic Redundancy Check) byte is used by the Pro-File electronics to detect if an error has occurred during a read or write operation).

The data block field is transparent to the Pro-File (which can only locate sectors), so the operating system from the host computer must keep track of which data block is in which sector.

The operating system references sectors by block numbers. It knows that it has 9728 (decimal), or 0 to 25FF (hexidecimal) sectors available for data on the Pro-File.

So it assigns each block of data to a number from 0 to 25FF called its **logical block number** and writes/reads the data blocks to/from the Pro-File using this number. The Pro-File converts the **logical block number** into sector locations to actually store or read the data block.

During formatting, the Format program writes the sector marks and headers into their areas within each sector. At this time, all sectors are referenced to a physical location on the disk called the Index (see figure).

The Index position is sensed when a small magnetic tab fastened on the disk motor passes by a magnetic sensor during motor rotation.

Note: The Index signal is only used during formatting, although it may be sampled by the technician with a frequency counter to determine the HDA's motor speed.

(Continued on the next page.)

163

DESCRIPTION OF THE PROFILE HDA FORMAT (continued)

In a properly operating Pro-File, sector marks and headers should never change after formatting, since the locations of the sectors will never change. A data block, on the other hand, will change every time its file is written to, by the Host computer system.

When looking at figure you probably noticed that the sectors are not sequentially numbered. This is because the sectors are **interleaved**. When the Pro-File goes after a sector, it reads sector headers to determine where the head is in relation to the target sector.

Without **sector interleaving** if it read the sector header of the sector immediately preceding the target sector, then the electronics in the Pro-File could not react fast enough to begin reading the data block of the target sector, so it would have to wait another complete revolution of the disk to do so.

With **sector interleaving** the program can read a sector header and know that the next sector in the numerical sequence will always occur a specific number of sectors following it, so it has time to prepare within that same revolution.

Sector interleaving is a result of the formatting for a given system. The format in the figure shown is for an Apple /// system. Each sector in this format is offset from the next in its numerical sequence by 5 sectors. The interleaving scheme for a Lisa system offsets the numerical sequence of each sector by 7.

164

SPECIAL FUNCTION TRACKS

Directories: Now another question: How does the host computer's operating system keep track of where all the blocks belonging to a file are located?

The answer, it uses the Pro-File's HDA to store **directory** information on each file.

Directories are created and maintained by the host computer's operating system. As far as the Pro-File is concerned, they are just data block fields. But without them the operating system would not be able to send the Pro-File the proper block numbers to retrieve the data blocks for a file.

NOTE: The following discussion which very briefly describes the Apple /// Sophisticated Operating System's (SOS) directory usage, is for example purposes only, and subject to change as the Sophisticated Operating System is revised.

On an Apple /// system, track 0 on the Pro-File is reserved for the **Root directory** which contains descriptive information about every main file that SOS (the Apple /// operating system) has put on the Pro-File. This information includes:

a. File name - The name of the file in ASCII.

b. The block number for each file's key block - Each files key block contains many items of information about the file. Among them will be a list of every block number in the file.

When an Apple /// host computer wants a file it reads the Root directory which refers it to the files key block (this may be through several subdirectories). The host then reads the key block and sequentially takes each block number from the file's block listing in the key block and sends it to the Pro-File to read the file.

The firmware program in the Z8 takes care of converting the logical block number from the host to a sector address so it can actually acquire the data block. (For more information on the Apple /// directory structure refer to the SOS Reference Manual)

165

SPECIAL FUNCTION TRACKS (continued)

A Lisa uses a different directory structure, but Pro-File operation is still the same.

Because all communication between the Pro-File and the HOST is done block by block, the Pro-File is known as a **block device** (as opposed to **character devices**, which communicate one byte at a time (such as a printer)).

The next block in the file is not necessarily located in the next sequential sector on that track. In fact very often it is located on an entirely different track.

That is why the Pro-File sometimes seems to go through a lot of gyrations as it reads a file. The sectors containing the blocks in the file are probably located on several different disk surfaces and tracks.

Track 77 on all disk surfaces is dedicated for the exclusive use of the Pro-File. The host cannot write to these tracks unless the Pro-File is being formatted. The following discussion describes their usage.

The Spares Storage Track (Track 77, Disk Surfaces 0, and 1):
What happens if there is an error?

When the Pro-File looks for a block of data, it performs many tests to confirm that the correct sector is being read accurately (for a more detailed description of these tests, refer to the Firmware Routines description in this section).

Sometimes the Pro-File determines that the media where a sector is located is not useable, and so won't use it any more. A sector identified as being unuseable is called a **spared** sector.

If the data block from the spared sector can be salvaged, the Pro-File will store the data block from the faulty sector on a special track reserved for that purpose called the **Spares Storage Track** (located on track 77 on disk surfaces 0 and 1).

166

SPECIAL FUNCTION TRACKS (continued)

The **Spares Storage Track** provides storage for a maximum of 32 spared sectors (2 surfaces, 1 track on each surface, 16 sectors per track).

The Spares Table Track (Track 77, on Disk Surfaces 2, and 3):
Once the data block from the spared sector has been stored, the Pro-File will record the old sector location of the salvaged data block along with its new spares storage sector location in the **Spares Table Track** (located on track 77 on disk surfaces 2, and 3).

Spares table information is stored redundantly on the **Spares Table Track** so any addition of a newly spared sector address will require an update to all 32 sectors on both surfaces of this track.

When the host sends the Pro-File a block number to be read, the Z8 will search the **Spares Table** (an image of this table is maintained in RAM), to determine if that block is on the **Spares Storage Track.** If it is, the Z8 will then go to the spares sector address to acquire the data block.

167

FIRMWARE ROUTINES

The following descriptions discuss the operation of some of the firmware routines contained in the ROM of the system Z8 on the Controller PCB, and how they affect the operation of the Pro-File.

1. SCAN OPERATION

After power up there are several things that take place before the firmware program in the system Z8 microprocessor on the Controller PCB will allow communication with the host computer system.

Note: The position of the heads over the disks may be visually monitored by observing the position of the Interruptor arm on the HDA (as shown in Sheet 1 of the troubleshooting flowchart).

a. **First** the power must come up to the proper levels and be stabilized. **Note:** To indicate this process, upon power up, the Ready Lamp will light for between 0.5 and 3 seconds.

b. **Second** the Z8 waits for about 20 seconds to allow the HDA motor to come up to 3600 RPM (its operating speed). **Note:** During this process, the Ready Lamp will remain off.

c. **Third** the Z8 does a **Seek** routine to the **Spares Table Track** on track 77, on disk surfaces 2, and 3. **Seek** routine operation is standard for any read, write, or write/verify operation requiring head movement to a new track and is performed as follows.

   During the **Seek** routine the Z8 issues the proper number of steps to the stepper motor to move the head over the target track, and performs a verification process to confirm that the head is over the proper track.

   During the **Seek** verification process the Z8 reads 3 alternate sectors from the track the heads are over.

(Continued on the next page.)

168

1. SCAN OPERATION (continued)

The sector header fields from these sectors tell the Z8 which head and track have actually been selected. A CRC check is also performed on the data block fields from these 3 sectors.

(a) If the head and track ID's read from the 3 sector header fields prove that the proper head is positioned over the proper track, and the 3 data block field's CRC status is OK, then the Z8 will exit the **Seek** routine and enter the **Read** routine.

The **Read** routine would normally read every sector on the now confirmed track until the target sector is read. However since each sector on the spares table track is redundant (each sector contains the duplicate copies of the spares table), there is no target sector.

Any sector that has a good CRC check during read is acceptable. If the first sector has a CRC error, then the Z8 will just read the next and so on until it finds a good one.

(b)* If during the attempt to confirm a good **Seek** to the **Spares Table Track** on track 77, a sector header is unable to be read, or a CRC error in a sectors data block field occurs, then the **Seek** routine will continue reading the 3 alternate sectors for up to 64 errors.

* If over 64 errors occur, then the Z8 moves the head to track 0 (the outermost track) and trys the **Seek** routine to the target track again (in this case track 77). If that fails the Z8 will move the head in to track 155 (the innermost track) and again reseek to the target track.

* The head movement to track 0 and track 155 and reseeking process during **Seek** verification is performed twice.

If the proper track is still not found, then the **Seek** routine will be exited, **Scan** operation is abandoned, and the heads are moved back to park position.

(Continued on the next page.)

169

1. SCAN OPERATION (continued)

d. **Fourth** the firmware program in the Z8 performs a sector header read, and data block CRC check of every sector on the HDA beginning with track 0 as follows: **Note:** The Ready Lamp will flash on and off during this process.

(1) Each movement of the heads to a new track requires a **Seek** routine as described above.

(a) If the head and track ID's read from the 3 sectors during the **Seek** routine prove that the proper head is positioned over the proper track, and their CRC status is OK, then the Z8 will exit the **Seek** routine and enter the **Read** routine.

The **Read** routine will read each sector header on the track until it finds one that matches the target sector whose sector address is currently in RAM.

* If during the **Read** routine the correct header is found, then the data block is read and its CRC status is checked. If the data blocks CRC status checks good, then the Scan operation goes on to the next sector to be read.

* If during the **Read** routine there is a timeout error (the target header cannot be found within 150 ms), or a CRC error occurs in the target sectors data block field, then the **Retry** routine (described in section 2) is called to attempt to get a good data block read from the sector.

(b) At this point in **Scan** operation, if an error occurs the **Seek** error recovery procedure described above will be followed, with the following exception.

If after the procedure is accomplished the proper track is still not found, then the Z8 will use the last track it could confirm a good seek on, as a reference to approximate the position of the target track, and move the head to that position. Then regardless of whether it could confirm a good seek to the target track or not, it will exit the **Seek** routine and enter the **Read** routine to read the target sector.

170

2. RETRY ROUTINE

The **Retry** routine is called for the purpose of attempting to read a good data block field after an unsuccessful read operation has occurred.

There are 2 variables in the **Retry** routine, the retry number, and the sparing threshold. These values are given by the host computer's operating system when communications between it and the Pro-File are initiated.

During the Scan operation since no communication from a host can occur, default values of 105 for the retry variable, and 31 for the sparing threshold are used. When called the **Retry** routine will:

a. Read the target sector the number of times specified by the retry variable (in the Scans case the default value is 105)

(1) If after the total number of retrys have occurred, there are no successful reads, the **Retry** routine will try to reread the sector 90 more times, or until it reads a data block without a CRC error.

(a) If all of the 90 reads are unsuccessful, then the sector address is written to the bad block table in RAM, and the spares table track (track 77 of disk surfaces 2 and 3).

Note: If **Retry** is called during Scan operation, the spares table track (track 77 on disk surfaces 2 or 3) will be updated when the host computer's operating system initiates communication with the Pro-File (this will be after the completion of Scan).

(b) If any of the 90 reads are successful a good data block from one of the successful reads is held in RAM while the **Diagnostic** routine (described later) is called to check out the media for the sector in question.
If the **Diagnostic** routine determines that the sector has bad media, it will spare the sector.
If the **Diagnostic** determines the sector to be good, then the **Diagnostic** and **Retry** routines are exitted.

171

2. RETRY ROUTINE (continued)

(2) If after the total number of retrys have occurred, there were successful reads, then the Z8 checks to see if there were more unsuccessful reads than the spares threshold variable allows (in Scans case the default is 31).

(a) If the sparing threshold has not been exceeded then the **Retry** routine is exitted, and control is given back to the **Scan** operation.

(b) If the sparing threshold is exceeded, but at least 1 good data block read occurred, then the data block field is rewritten to the questionable sector and the **Diagnostic** routine (described later) is called to check out the target sectors media.

If the **Diagnostic** routine determines that the sector has bad media, it will spare the sector. If the **Diagnostic** determines the sector to be good, then the **Diagnostic** routine is exitted and control is given back to the **Scan** operation.

172

3. DIAGNOSTIC ROUTINE

The **Diagnostic** routine is called for the purpose of checking media quality on a given sector location, and sparing the sector if the media quality is determined to be bad. When the **Diagnostic** routine is called it will:

a. Read the target sector (specified by the sector address currently in RAM), 100 times.

(1) If over 30 CRC and/or timeout errors occur, then the **Diagnostic** routine will write the data block into a sector on the spares storage track (track 77 of heads 0 or 1).

The **Diagnostic** routine will then write the old sector address, and its new spares track sector address in the spares table in RAM, and after the next handshake with the host system will update the spares table track (track 77 on heads 2 or 3).

The **Diagnostic** routine is then performed on the new spare sector location of the data block to make sure that it is a good media location.

Note: During Scan operation the spares table track (track 77 on disk surfaces 2 or 3) will be updated when the host computer's operating system initiates communication with the Pro-File (this will be after the completion of Scan).

(2) If less than 30 CRC or timeout errors occur, then the **Diagnostic** routine is exitted, leaving the data block in its present sector (the media must be OK).

173

4. During Normal Operation - the **Seek** and **Read** routines will be performed just as they were in the Scan operation.

If an error occurs, the sequence of operations is the same as that of scan with the exception that the retry and spares threshold variables will be different depending upon the operating system.

The **Diagnostic** routine will always be called if the verify fails during a Write/Verify operation.

If a sector address is in the Bad Block table, then that means that a Retry routine must have been called for that sector, and during the routine there was never a good read of the data block.

This means that the Pro-File does not have a good data block for that sector and can't perform a **Diagnostic** on the questionable sector location until it has one.

If the host does a write to that sector address listed in the Bad Block table, then at that time the Pro-File has a good data block for that sector so then the Diagnostic routine is automatically called to verify the media at that location.

Of course if it doesn't check out then the sector is spared. In any event, whether the sector is spared, or is determined to be good and left as is, the sector's address will be deleted from the Bad Block table because the Pro-File now knows whether its media is good or not.

174

CONTROLLER PCB CIRCUIT DESCRIPTION

What's in This Description

On the next six pages you will find tables listing the pins and their functions for the:

a. Host interface plug P1.

b. Analog PCB interface plug P2.

c. Z8 Main Processing Unit on the Controller PCB.

Block diagram and circuit descriptions for the main functions on the Controller PCB follow these tables.

For specific circuit diagram information refer to the Controller PCB schematic at the end of this section.

175

Controller PCB to Host Interface (P1)
The Controller PCB and the Host communicate via a 25 pin plug, **P1**. Signals at P1 on the Controller PCB are as follows:

PIN	SIGNAL	DESCRIPTION
5, 6, 8, 11 12, 13 22, 23	XD0-XD7	(bidirectional) /Data lines between the Host and the Pro-File.
3	RRW	(input) /Controls the direction of data on lines XD0-XD7.
17	CMD	(input) /Initiates communication with the Pro-File.
16	BSY/INT	(output) /Notifies the Host of Pro-File status.
15	PSTRB	(input) /When writing, this signal is generated by the Host to clock a byte into Pro-File's RAM; when reading, the pulse causes the Z8 to increment the Address Counters and so select the next sequential address to be read. During Read, the RAM is always selected to output the contents of the selected address; so after the Host generates PSTRB, it waits a short time and then gates in the byte from the I/O cable.
18	TPARITY	(output) /This signal is high or low, depending on the parity of the byte currently being transferred on data lines XD0-XD7.
21	CRES	(input) /When generated by the Host, this signal resets the Z8 to its initial state. This usually occurs at the beginning of communications, but whenever it happens depends on the Host.
25	CDET	This signal is used by Host interface to detect a disconnected or broken cable.
1	PH0	(input) /PH0 is a timing signal from the Host.

Controller PCB Circuit Descriptions

176

Controller PCB to Analog PCB Interface (P2)
The Controller PCB and the Analog PCB communicate via **P2**. The Analog PCB converts the Controller PCB's digital data into analog signals for the disk media. Signals at P2 on the Controller PCB are as follows:

PIN	SIGNAL	DESCRIPTION
12	Write Data	(output) /NRZ serial data to be written on the disk.
32	Read Data	(input) /Digitized NRZ serial data recovered from the disk.
10	Sys Clock	(output) /Provides the 10 MHz system clock to the Analog PCB for write operations.
34	Read Clock	(input) / Provides the Controller PCB with a clock that is in sync with the Read data.
30	Read Gate	(output) /Enables the Read circuitry on the Analog PCB.
22	Write Gate	(output) /Enables the write circuitry on the Analog PCB.
4	Index	(input) /Index is a pulse that occurs once per disk revolution and is only used by the Z8 during formatting.
8	Write Sector Mark	(output) /Enables the Analog PCB to write DC sector boundaries during formatting.
15	Sector Mark	(input) /Signals the Controller PCB when the heads have encountered a sector mark on the disk.
26, 28	Head Select 1,0	(output) /Causes selection of one of four Read/Write heads.
2	Track 0	(input) /Indicates that the heads are positioned over the outermost tracks.
24	Precompensation	(output) /Enables the Precompensation and low current circuitry on the Analog PCB when the inner tracks (128 to 152) are being written to.
14, 16, 20, 18	Stepper Phases	(output) /Activates stepper motor coils to position heads. Generated by the Z8.

177

Z8 Main Processor Pin Descriptions

The following are signal descriptions produced from the Z8 on the Controller PCB. The Z8 has five sets of pins:

1. General purpose signals.
2. Port 0.
3. Port 1.
4. Port 2.
5. Port 3.

1. General Purpose Signals

PIN	SIGNAL	DESCRIPTION
2, 3	Clock	5 MHZ
6	Reset	'Power OK'; system reset line or $\overline{\text{CRES}}$.
9	$\overline{\text{AS}}$	Address strobe, used to latch data from Port 1 and Port 0 [0:3] into LS161 counters (RAM address counters).
8	$\overline{\text{DS}}$	Data strobe, defines data valid time during Writes or Reads to RAM and writes to 8253 counter.
7	ZR$\overline{\text{W}}$	Z8 Read/Write line. Defines external memory for I/O operation as a Read or Write.

NOTE: $\overline{\text{AS}}$, $\overline{\text{DS}}$, and $\overline{\text{Z}}$RW are tri-stated when Port 1 is placed in the high impedance state.

2. PORT 0

PIN	SIGNAL	DESCRIPTION
13,14	P00-P02	High-order address bits, A8-A10, for RAM
15	A8-A10	access. They are latched into the LS161 address counter during AS.
16	P03 PCOMP	Precomp/Reduce Write Current command to disk Write electronics. (State is 'don't care' if not writing.)
17,18	P04-P07	0B2, 0B1, 0A2, and 0A1 commands, respectively,
19,20	01-04	to the stepper motor drive circuitry.

178

3. PORT 1

System data bus (8 bits). Its different functions are listed below.

PIN	SIGNAL	DESCRIPTION
21thru 28	P10-P17 ZR7-ZR0	a. Low order address bits and data, multiplexed by AS and DS, for Z8 RAM accesses.
		b. Data bits for loading the 8253 counters. (LDCTR, derived from DS when MSEL1 is low, is the 'Write' input to the 8253.)
		c. Data to 1 from the Host. The Z8 program must set Port 1 to the high impedance state.
		d. Data to 1 from the disk. The Z8 program sets Port 1 to hi-Z.

4. PORT 2

These pins are programmed as inputs or outputs using a Z8 mode register. All pins are set to hi-Z (input mode) after a reset.

PIN	SIGNAL	DESCRIPTION
31	P20 WRTSM	Write Sector Mark (WRTSM) output. This signal, used in conjunction with Format Enable (FMTEN), which causes WTGT to be true, writes sector marks on the disk. This should happen only during the format operation.
32	P21 TRK0	Track 0 (TRK0) input. Indicates that R/W heads are positioned over outermost track.
33	P22 CMD	Command (CMD) input. This line is set true by the Host via the interface card, and it indicates that the Host is requesting access to the controller RAM. The corresponding handshake line, BSY/INT, is a Z8 output described later.
34	P23 BSY/INT	Busy/Interrupt (BSY/INT). This signal acknowledges the CMD input via the CMD-BSY protocol described elsewhere. BSY can also be used to interrupt the Host if enabled on the Interface card.

(Continued on the next page.)

179

(Port 2 pin description is continued from the previous page.)

35	**P24 MSEL0**	(MSEL0) The low order select bit to control RAM access.
36	**P25 MSEL1**	(MSEL1/PSEL) High order RAM access select bit. Also controls data input to RAM from the Host or the disk; hi = disk, low = Host.
37	**P26 START/ RSTERR**	Start/reset error (START/RSTERR). Enables Read/Write control hardware to begin a Read or Write operation at the next sector pulse. Also resets the CRC error flip-flop and the sector timing register.
38	**P27 DRW**	Disk Read/Write (DRW). Controls the Read/ Write inputs to the RAM for Host or disk accesses. Command input to Read/Write hardware. High when Read, low when Write

5. PORT 3

P33-P30 are inputs and/or interrupts; P37-P34 are outputs.

PIN	SIGNAL	DESCRIPTION
5	**P30 SER IN**	Not currently used.
39	**P31 SECTOR**	This signal is generated on the Analog PCB whenever it detects a Sector Mark on the disk.
12	**P32 SECTDN**	Sector Done (SECTDN). (Input) Resets the 8253 byte counters and the lower four bits of the RAM address counter when the Read/Write hardware is searching for the target sector. Goes low when the Read/Write operation has been completed. Stays low until START.

(Continued on the next page.)

1980

(Port 3 pin description is continued from the previous page.)

30	**P33 CRCERR**	CRC Error (CRCERR). (Input) Goes low if a CRC error is detected during a disk Read.
29,10	**P34,P35 HS0,HS1**	Head Select 0, 1 (HS0, HS1). Binary coded bits to select head 0, 1, 2, or 3.
40	**P36 WRTSM**	Write Sector Mark (WRTSM). (Output) Causes sector marks to be written on the disk if FMTEN is high and the Format jumper is installed.
4	**P37 RDHDR/TX**	Read Header/Transmit (RDHDR/TX). Command input to the Read/Write control circuits. If DRW (P27) is high and START/RSTERR (P26) is low, RHDHR hi will cause the leader field on the next sector to be stored in RAM.

Controller PCB Circuit Descriptions

KSI

Principles of Operation
Upon power up, the Z8 issues phase steps to the Head Stepper motor on the HDA to move the heads to track 77. The Z8 then performs a Read operation to read the Spares table. If this is successful, the firmware in the Z8 will perform the Scan sequence.

(For a more detailed explanation of what the firmware in the Z8 is doing (i.e., Scan, Seek, etc.), or information on the format used for the Pro-File HDA (i.e., Spares Table, sector header composition, etc.), refer to the Appendices section of this manual.)

The power up reading of the Spares Table and the performance of the Scan sequence are basically composed of Check Header and Read operations. These operations are discussed later.

Once the Scan is completed and the initial (operating system specific) handshake has taken place, the Pro-File remains idle until the Host requests an operation. All Host-requested operations in the Pro-File are performed in three steps: the Command Handshake, the Check Header function, and the Operation.

1. **The Command Handshake** - The Host sends command bytes to the Pro-File to tell it whether to perform a Read, Write, or Write/Verify operation, and on what logical block number to perform it.

   The Z8 on the Controller PCB converts the logical block to location bytes (i.e., physical location on the disk; target head, track, and sector) and stores them in RAM.

   The Command Handshake is the same for any of the three operations. (Read, Write, or Write/Verify).

(Continued on the following page.)

182

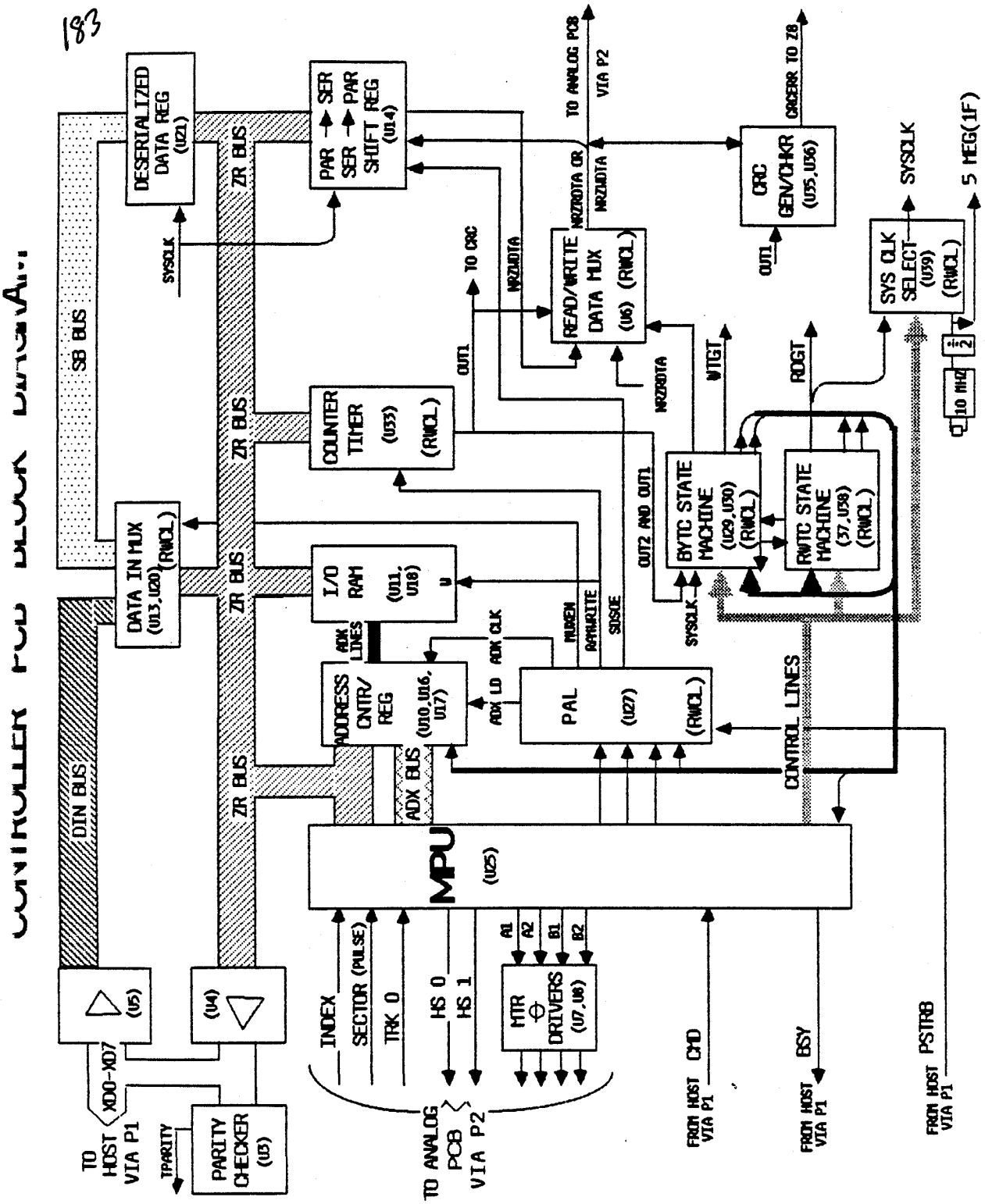
2. **The Check Header Function** - If the target sector is on a different head or track from the one already selected, the Z8 will perform the Seek routine to move all 4 heads to the target track (the track containing the target sector) and will enable the target head (the head over the disk surface containing the target track and sector).

   The Seek routine then uses the Check Header function to read three consecutive sector headers from the track it has moved the head to, and compares them with the target head, track, and sector bytes in RAM.

   The process used in the Check Header function to read the sector headers (to confirm a Seek to the right track) is almost identical to the first part of the Read, Write, and Write/Verify operations, which also read sector headers to find their target sectors.

3. **The Operation** - Once a successful Seek is confirmed, the Z8 coordinates the circuitry in the Controller PCB and Analog PCB to perform it. Each operation follows a different sequence of events.

183



184

CONTROLLER PCB DETAILED BLOCK DIAGRAMS

The following is a brief description of each of the 17 functional elements on the Controller PCB. Understanding the functions of these elements will help you to understand the signal flow discussions later.

For more specific circuit diagram information refer to the Controller PCB schematic at the end of this section.

Z8 Main Processing Unit U25
The Controller PCB's MPU is a Z8 microprocessor, which has a self-contained ROM program. It provides an intelligent interface to the Host computer. It indirectly controls the Pro-File's electronics by setting modes and directly controls the stepper motor and head selection.

PAL U27
This single chip is a logic array specifically programmed for this application. It performs complex and/or combinational logic functions. Primarily, it controls the function of the address counter, the direction of data to/from RAM, loading the counter/timer, and the serial/deserial register.

RAM Address Counter/Register U10, U16, U17
The method of use depends on what operation is currently going on. This element can be preset to a certain point and counted up through a sequential range of addresses for RAM access. It is also used as an Address register where it is loaded with a value for a specific single access.

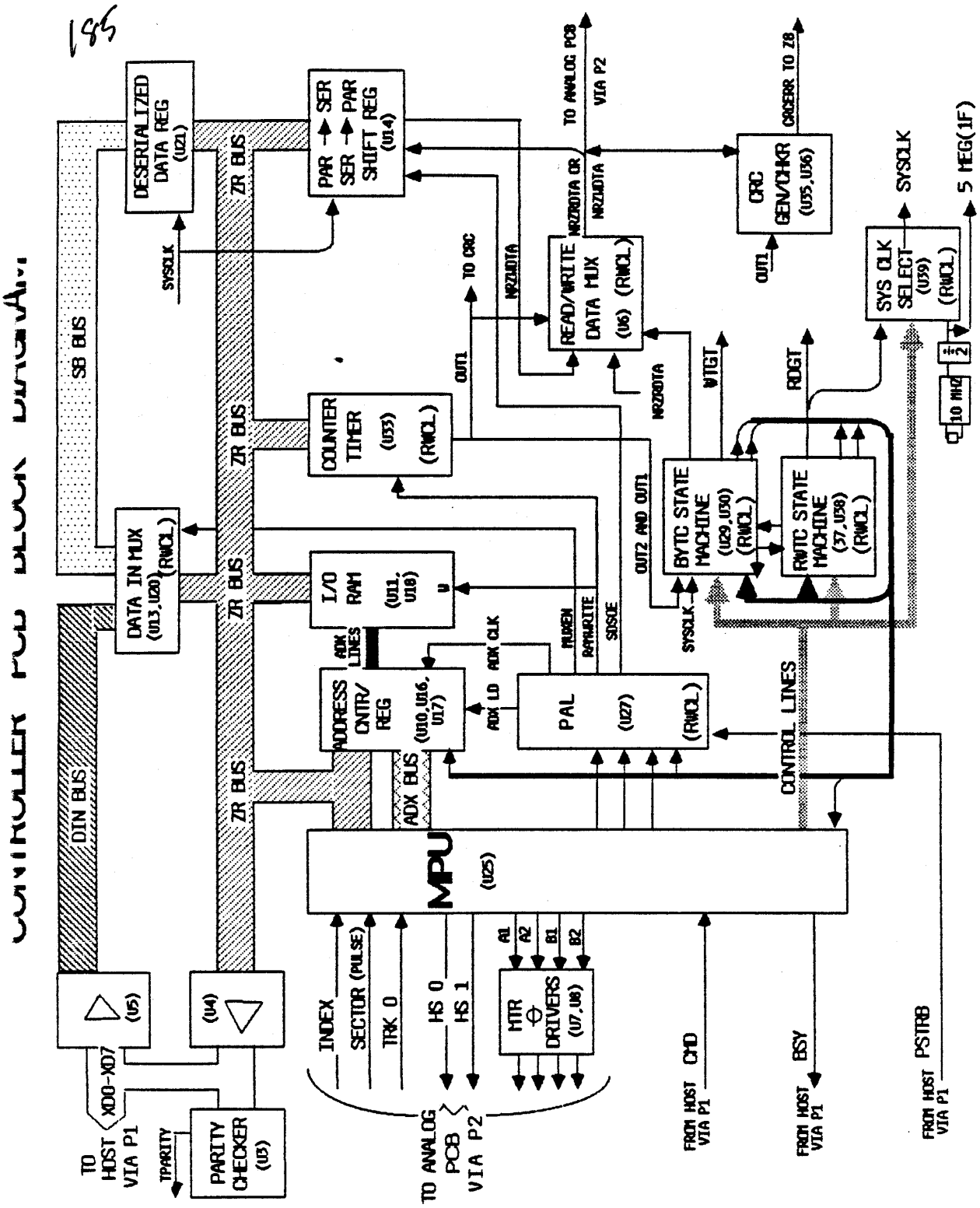
RAM U11, U18
The RAM is a two kilobyte RAM array used to hold data to and from the Host and disk. Various locations are used to hold status information and the current spares tables.

Data Line Drivers U4, U5
These simple Line Drivers drive data to and from the Controller and Host interface PCBs.

Data In MUX U13, U20
This MUX is used to direct data coming in from the Host interface PCB or from the disk. The usual destination of its outputs is the RAM.

Controller PCB Circuit Descriptions

185



186

Bus Parity Checker U3
The BPC forms the other half of the bus parity checking circuit on the Host interface PCB. It constantly monitors the bus, checks the parity at the Controller end, and sends its sum to the Host interface to be compared with the sum on the interface end. A parity error should not occur unless a cable fault exists.

BYTC State Machine U29, U30
The BYTC (Byte Control) element of the State Machine, one of the two major elements of the Read/Write Control circuitry, steps through the control states for each part of an operation at a bit-time rate.

RWTC State Machine U37, U38
The RWTC (Read Write Timing Control) element of the State Machine, the second major element of the Read/Write Control, steps through the control states necessary to control the timing for all the operations associated with block/sector reading and writing.

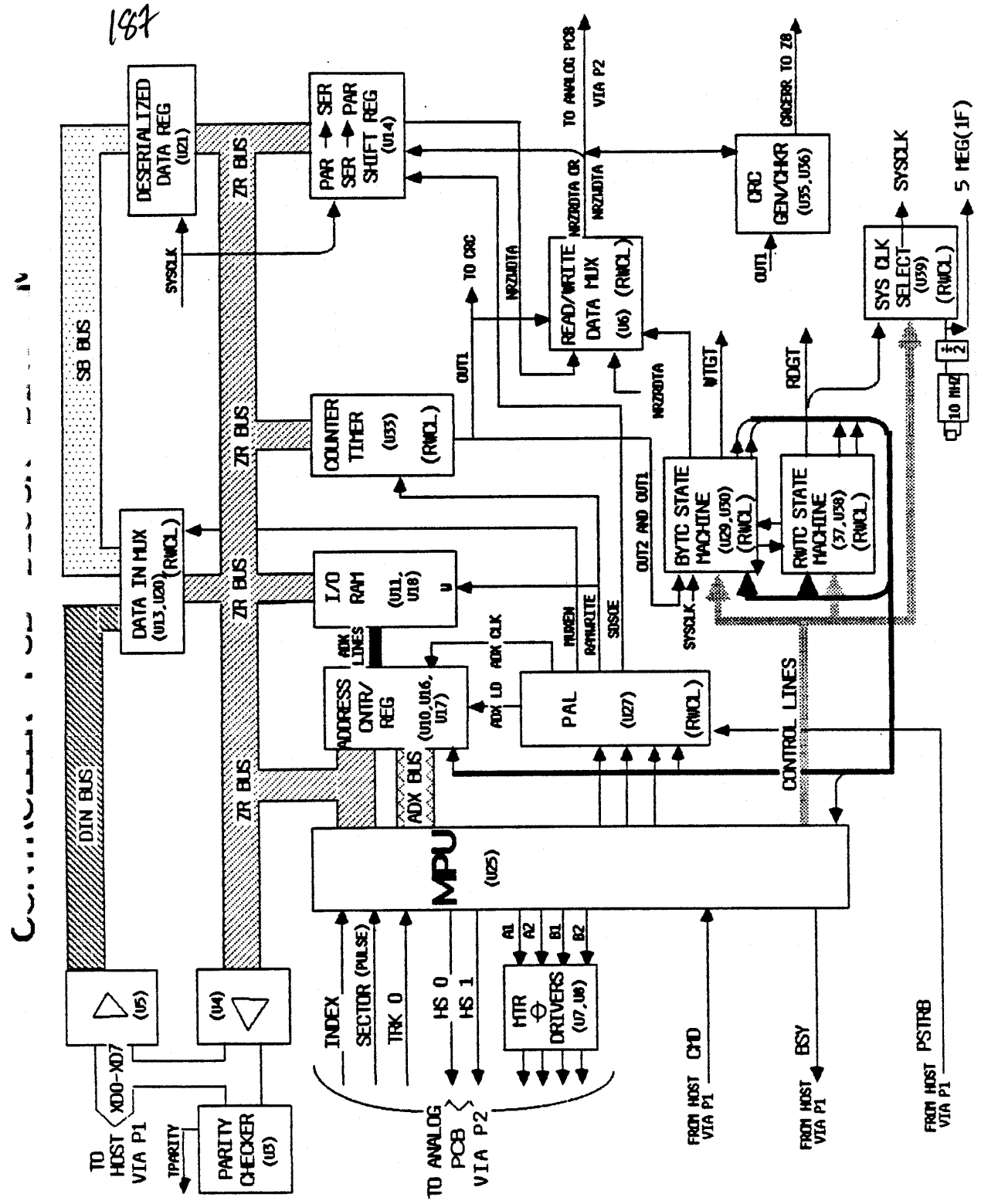
System Clock Selector U31, U32, U39
The SSC switches from the crystal oscillator, used during idle and Write operations, to the Read clock generated by the Analog PCB during the Read operation. This keeps the logic in sync with the data.

Divide-by-8 Counter
Counts the system bit-clocks that the System Clock Selector has selected (either the clocks from the Controller PCB's crystal oscillator or RDCLKs from the Analog PCB) and produces a positive transition from its QC output for every eight bit-clocks received (one per byte). Its QB and QA outputs enable the State Machine to discriminate phase (bit times) within the processing of a byte.

Programmable Counter/Timer U33
This chip receives the QC output of the Divide-by-8 Counter as its input clock. It contains three programmable counters, which yield byte-time information to the Read Write Control logic, and it basically keeps track of what part of the particular sector is being processed by the Controller PCB.

Controller PCB Circuit Descriptions

187



188

Read/Write Data MUX U6
The Read Data MUX is used to select one of two sources of data, either Serialized Data from the SER/DESER register, or NRZ Read Data from the Analog PCB. When reading, it gates NRZ Read Data through to the Serial/Deserial Shift Register U14. During a Write, it gates serial data from the Serial/Deserial Shift Register U14, to the CRC generator U35.

CRC Generator/Checker U35, U36
The CRC (Cyclic Redundancy Check) circuit is used to compute CRC check characters that are written at the end of each data block on disk during Write operations, to compute CRC for Read data, and to compare the result with the CRC characters that were read at the end of each data block

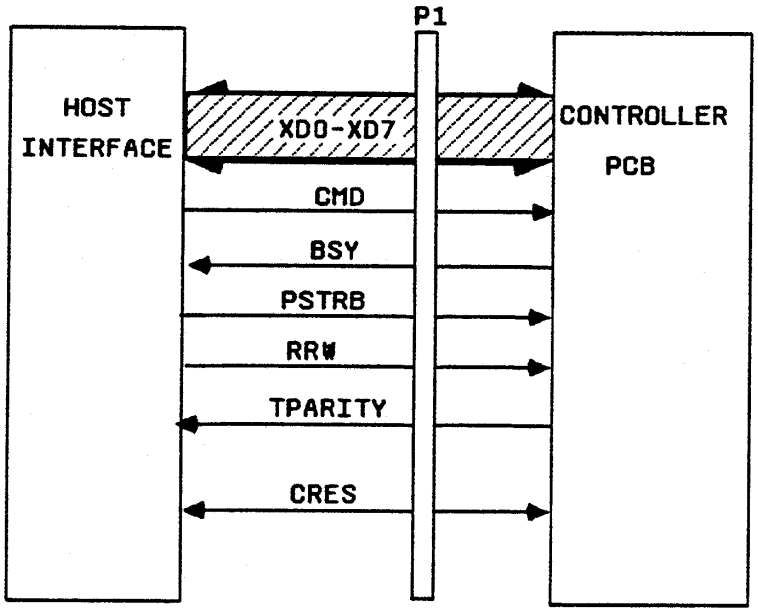
Deserialized Data Register U21
This 8-bit register temporarily holds the deserialized data from the disk so that the shift register can receive the next byte. When the logic is ready, it directs the register's contents to RAM though the Data In MUX.

Serial/Deserial Shift Register U14
This register is used to take the parallel data from RAM and shift it out serially to the Analog PCB and to take the serial data from disk and shift it into a parallel format for transfer back into RAM.

Stepper Motor Drivers U7, U8
The 4 phases required for Stepper motor movement are generated by the Z8. The current for these signals is then boosted by the Stepper Motor Drivers U7, and U8.

189

PRO-FILE HOST INTERFACE



190

CONTROLLER PCB SIGNAL FLOW DESCRIPTION

To understand Controller PCB operation, you should first become familiar with data flow in the three stages of an operation; the Command Handshake, the Check Header function, and the operation itself. The following discussions describe each stage, first in general and then in detail.

The Z8 is used to condition the logic, but it is not actively involved with data transfers to/from the disk or Host; that is done by the Read/Write Control logic (RWCL which is just about everything else on the Controller PCB except the Z8, RAM, and Host interface circuitry).

(For information on the Pro-File format on composition of the sectors in the format, refer to the Pro-File HDA Description in the Appendices section.)

A. Command Handshake

Command Handshake General Explanation
Assume that the Pro-File is initially sitting idle with the BSY line high (not active), the disks spinning, and the heads over track 155 (Park position), waiting for the Host to tell it to do something. The Host communicates this message during the Command Handshake.

The Host asserts CMD (active low) to initiate communications with the Z8. Upon seeing CMD going low, the Z8 lowers its BSY line and waits for the Host to raise CMD.

When the Pro-File sees CMD go high it places a 01 response byte on the interface bus and raises BSY.

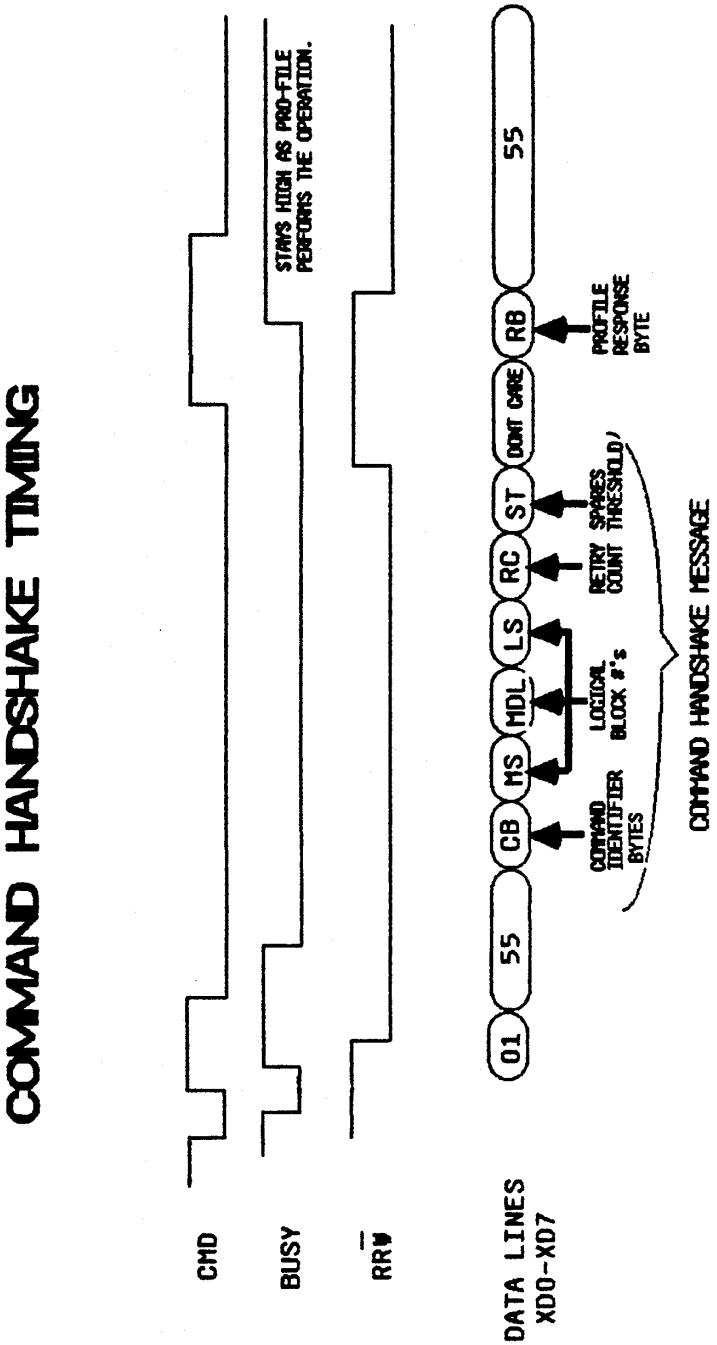
The Host sees the BSY line go high and interprets the 01 as an ACK, so it lowers the RRW signal. The low RRW signal enables the Pro-File to write the command bytes the Host will send into its RAM.

The Host must acknowledge the Pro-File's response with a 55, or the Z8 will abort the operation and go back to idle. The Host puts its 55 response byte on the bus and lowers the CMD line.

The low RRW and the response of 55 cause the Z8 to condition the bus to receive the command bytes, which are not immediately read by the Z8 but are stored in RAM for future reference.

After the Host puts each byte on the I/O bus, it generates PSTRB. The positive transition of the negative pulse PSTRB is used to clock the byte into the RAM.

191



192

When CMD goes low again, the Z8 interprets the command bytes and responds with the result of its command interpretation. For example, if the Host has said to read a block, the Z8 would respond with "02", which means "I'm going to read a block".

If it wants the operation to continue, the Host must confirm the response with a 55 on the bus again. If it disagrees or has changed its mind, the Host will send a byte other than a 55 causing the Pro-File to abort the operations.

Two handshakes are required to complete a Read operation. The first one is the Command Handshake, and the second is when the Pro-File sends the Read data, and the completion status of the operation back to the Host.

Three are required for both a Write and a Write/Verify operation. The first one is the Command Handshake, and the second is when the Host sends the block of write data to the Pro-File, and the third is when the Pro-File sends the completion status of the operation back to the Host.

The command identifier bytes for each of the three commands are as follows, 00 fro Read, 01 for Write, and 02 for Write/Verify. A Command Handshake message is composed of the following elements.

Command Identifier	Logical Block #	Retry Count	Sparing Threshold
XX	Most, Middle, and Least Significant	Host Specific	Host Specific

The Pro-File interprets CMD high as a request from the Host to send a byte telling the Host what the Pro-File expects to do next. When the Pro-File is waiting for a command, it sends an '01' in response to CMD high. The Pro-File's other responses are shown in the table below.

Host's command to Pro-File	Pro-File's Response
Initiate handshake (lowers CMD)	01
Read a block	02
Receive Write data	03
Receive Write/Verify data	04
Do the Write or W/V on disk	06

Controller PCB Circuit Descriptions

193

Following a Read or a Write, the Z8 provides the Host with four status bytes, which are placed in the buffer immediately preceding the data just read or written. The significance of the individual bits is listed below:

STATUS 1

7 = 1 if Pro-File received 55 to its last response

6 = 1 if Write or Write/Verify was aborted because the number of data bytes sent exceeded the data block limits or because the Pro-File couldn't read its spares table

5 = 1 if the Host's data is no longer in RAM because the Pro-File updated its spares table.

4 = 1 if SEEK ERROR - caused by Pro-File being unable in three tries to read three consecutive headers on a track

3 = 1 if CRC error, may occur only during an actual Read or verify of Write/Verify, not while trying to read headers after seeking

2 = 1 if TIMEOUT ERROR (couldn't find target sector's header in nine revolutions - Not set while trying to read headers after seeking)

1 = N.C.

0 = 1 if operation is unsuccessful

STATUS 2

7 = 1 if SEEK ERROR - occurs if Pro-File is unable in one try to read three consecutive headers on a track.

6 = 1 if spares table overflow (More 32 sectors spared)

5 = N.C.

4 = 1 if bad block table overflow occurs (Less than 100 bad blocks in table)

3 = 1 if the Pro-File is unable to read its status sector.

2 = 1 if sparing occurs.

1 = 1 if Seek to wrong track occurs.

0 = Not used.

194

STATUS 3

7 = 1 if the Pro-File has been reset

6 = 1 if block number is invalid

* 5 = 1 if block I.D. at end of sector is mismatched

4 = N.C.

3 = N.C.

* 2 = 1 if the Pro-File has been reset.

* 1 = 1 if the Pro-File gave a bad response

* 0 = 1 if CRC error occurs.

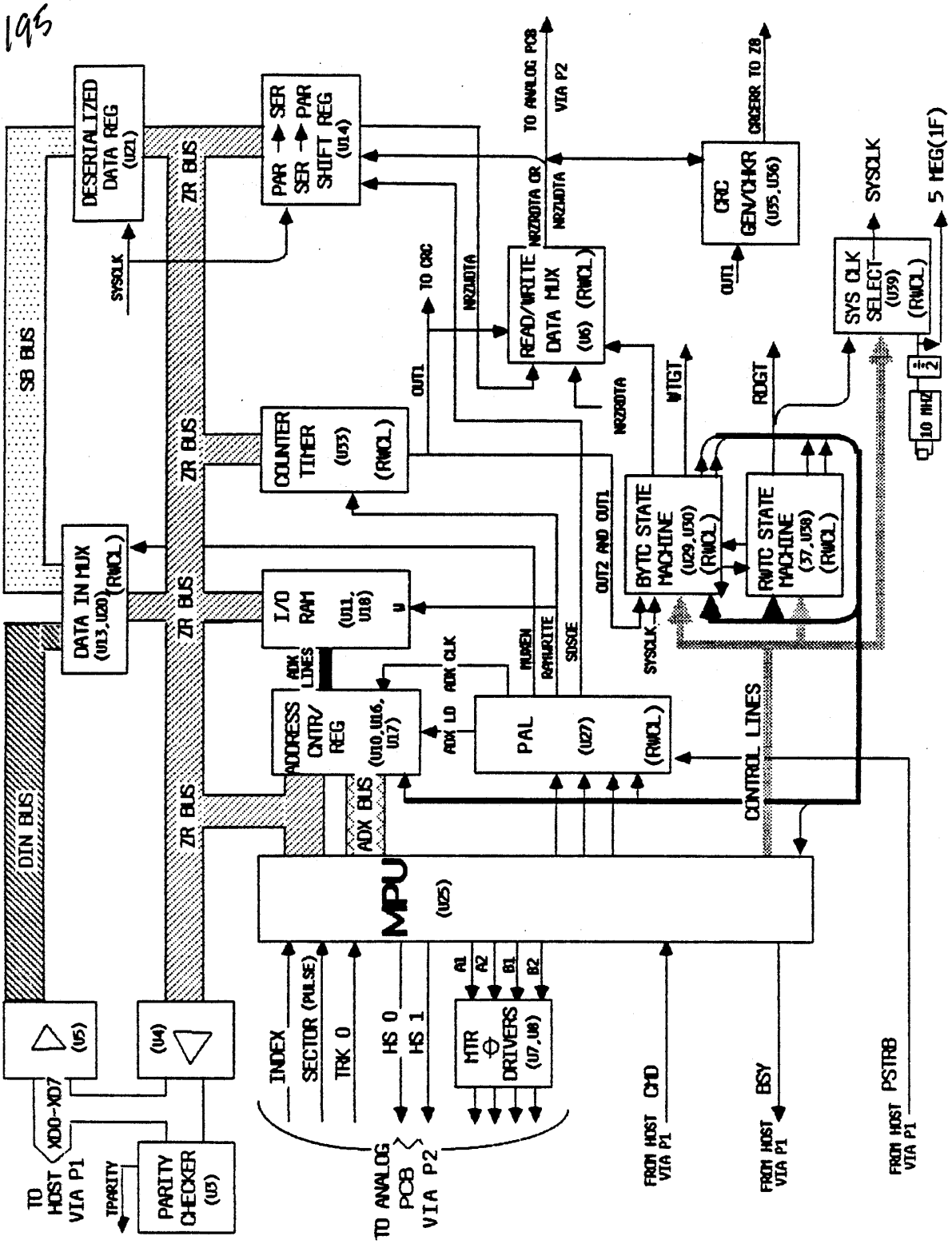
* These bits are sent by the Host driver.

STATUS 4

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

(Continued on the next page.)

195



Controller PCB Circuit Descriptions

196

When the Host interface's CMD signal goes high, the Z8 responds by sending a "01" response byte up to the ZR bus, and off to the left through the buffer to the Host interface. (See figure on the opposite page).

The Host lowers RRW and CMD. When the Host interface lowers CMD, the Z8 should see the Host's "55" acknowledgement coming in through the buffer to the Data In MUX directly to the Z8. This acknowledgement and RRW's being low triggers the Z8 to get the RAM ready to accept the command bytes.

After the first handshake of CMD and BSY, the Host transfers the actual command and delimiting bytes. They come, as do all data, from the Host through the Data In MUX, and are then stored in RAM.

The Z8 then evaluates the command bytes by bringing them in from RAM. After interpreting the command bytes, the Z8 gives its response byte, which is ack'd or nack'd by the Host.

Controller PCB Circuit Descriptions

197

Refer to the Controller PCB schematic at the end of this section for the following discussion.

Command Handshake Component Explanation

1. When the Host lowers CMD to pin 33 of the Z8, the program in the Z8 will lower BSY on pin 34. When the Host raises CMD, the Z8 puts a 01 on the bus and signals the Host by raising BSY.

   Because RRW is already high to pin 4 of U9, the direction of the data on the bus is from the Pro-File to the Host.

2. The Host interprets the 01 as an ACK and lowers RRW, which changes direction of the bus. Then it places 55 on the bus and signals the Pro-File by lowering CMD.

3. The first phase of every operation is always the same, so the program in the Z8 knows that after sending the 01, the Host should respond with a 55. The Z8 must enable the contents of the DI bus (should be a 55) to be gated onto the ZR bus so that it can verify that the Host has sent a 55.

4. To do this, the Z8 generates the low MSEL1 and MSEL0 signals, to enable the Write Multiplexers (U12 and U20) to select the inputs from the DI bus.

   Once the Z8 senses that CMD has gone low (an indication from the Host that it has put the data, in this case 55, on the data lines), it sends to PAL (U27) the combination of inputs it needs to generate the MUXEN signal, which gates the DI data onto the ZR bus.

   (For a table showing the conditions necessary to generate PAL signals, refer to the PAL Function Table at the end of this section.)

5. With the signals described above, a data path is established to pass the 55 from the DI (Data In) bus, through the Data In Multiplexers (U13, and U20), and onto the ZR bus. The Z8 then reads the 55 from the ZR bus.

198

6. The Z8 now knows that it will receive some command bytes telling it what operation the Host is requesting, so it puts the first address it wants the command bytes stored in on the ZR bus and causes the PAL to generate the LD (Load) pulse to pin 9 and the clock to pin 2 of each Address Counter.

   This presets the Address Counters to the address in which it wants to store the first command byte. **Note:** A10 in the address must be low to select RAM.

7. The Data In Multiplexers (U12 and U20) are still enabled to select the DI bus inputs and pass them to the ZR bus.

   (Recall that after the Z8 read the 55 (Step #3), it sent to PAL (U27) the combination of inputs it needed to generate the SEL signal to RAM).

8. A data path is now established to pass the command bytes that will shortly follow from the Host to the DI (Data In) bus, through the Write Multiplexers (U13 and U20), onto the ZR bus.

9. The Host waits a short period after each byte is put on the lines for the data to settle, and then generates the PSTRB signal.

10. Because of the PAL's current inputs, the PSTRB signal causes the PAL to generate the RAMWRITE signal from pin 16, which stores what is on the ZR bus into the RAM address specified by the Address Counters.

11. PSTRB also causes the PAL to generate an Address Counter Clock from pin 19, to increment the Address Counters for the next byte to come in.

12. When the Host has finished sending bytes, it raises RRW to pin 4 of U9, to change the direction of the Data bus.

    The Host then raises CMD again to tell the Z8 that no more command bytes are coming. Now the Z8 reads the command bytes from RAM to interpret them.

199

13. To read the command bytes, the Z8 sequentially sets the Address Counters to the addresses of the command bytes.

    RAMSEL and an address on the lines to the 2114-type RAM fully enable the RAM to put the contents of the address on the ZR bus. As each command byte is read onto the ZR bus, the Z8 reads and interprets it.

14. If the command is for a Read, the Z8 will respond by putting 02 on the ZR bus.

    If the command is for a Write, the response byte will be 03. If a Write/Verify, the response byte will be 04.

    After the Z8 puts the response byte on the ZR bus to the Host, it asserts BSY, signifying it is now performing the operation. The response byte is driven by U4 to the interface of the Host.

15. The Host checks the response. Because it knows which command it requested, it knows which response byte to expect.

    Since the response byte is 02, as expected, the Host acknowledges with 55 and raises CMD (it is true when low, so high means CMD is deactivated).

16. The Command Handshake is now complete. Once the Host raises CMD, the Z8 performs (if necessary) a Seek routine (explained later) to select the proper head and move it over the target track.

17. If the command is a Write or Write/Verify, then there will be another handshake to transfer the Write data during the Seek routine.

18. If the command is a Read, then once the proper head is selected and positioned over the target track, and the Seek has been verified, the Z8 will perform the Read.

200

B. Check Header Function

Check Header Function General Operation
For each Read or Write operation, the specific sector must be located and checked. This is accomplished by the the Z8, and the Read/Write Control Logic, during the Check Header function.

Before the Z8 conditions the logic to start a Read, Write, or Write/Verify operation, it translates the logical block number requested by the Host into target sector information bytes (i.e., head, track, and sector). It then checks to see if the desired block is in the spares table and sets the track, head, and sector accordingly.

The Z8 then sets a complete replica of the target header into a specific area of RAM. (For more information on the Seek routine and other firmware routines used in the Pro-File Z8, refer to the Firmware Routines Description.)

Seek Routine
The Seek routine is a firmware routine in the Z8. It controls selecting the proper head, positioning the heads, and putting the Read/Write Control Logic in the proper modes for a Check Header function after it has moved the selects

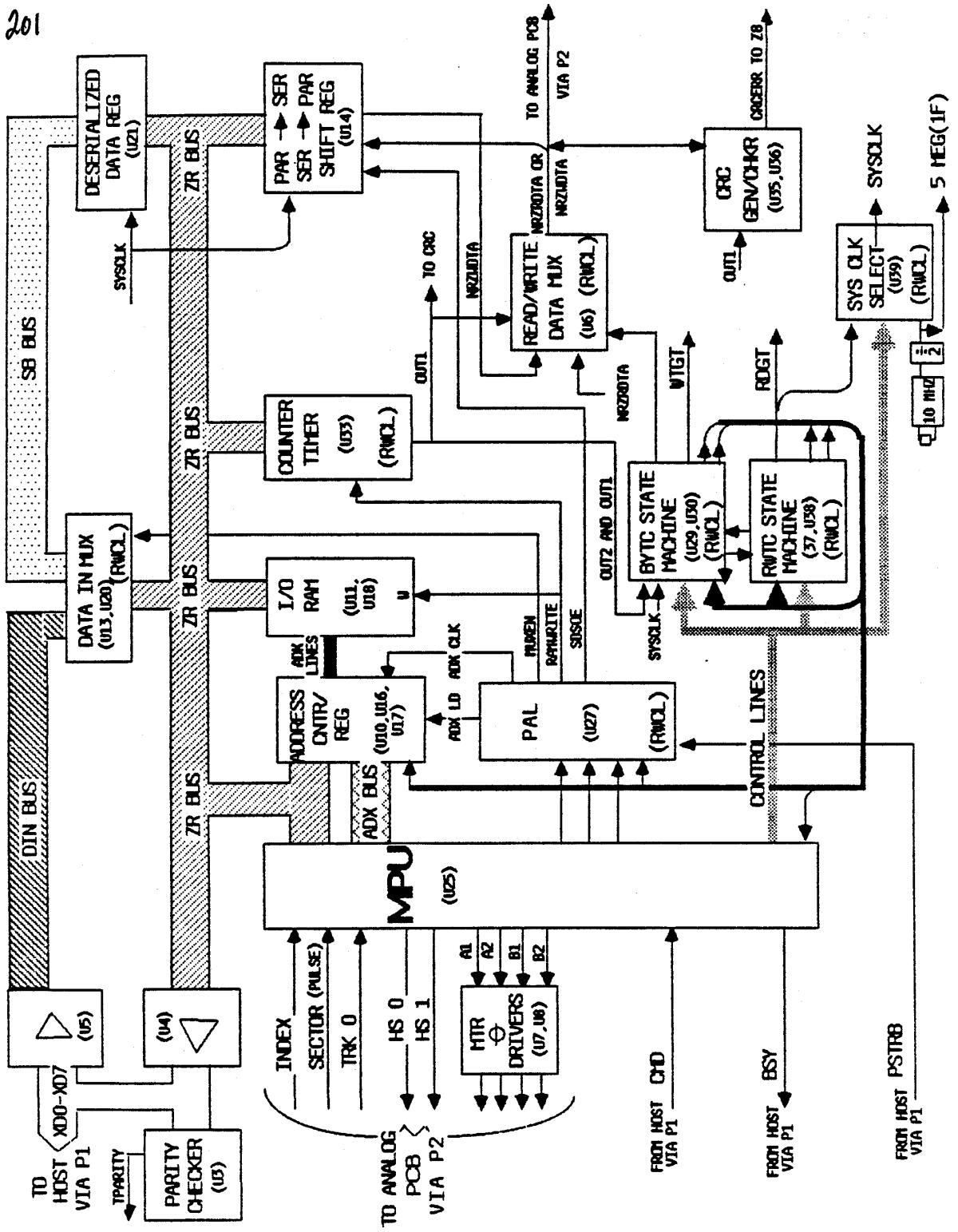
If the current block and the last block read or written have the same track and head, the Z8 will exit the Seek routine because the target head is assumed to be positioned over the target track.

If the track is the same but the head is different, the Z8 will send out enable signals to the target head, wait 750us, and then exit the Seek routine to perform the Host-requested operation.

If the track is different, the Z8 will generate the proper number of stepper pulses on the stepper phase control lines to position the head where it thinks the target track should be. Then it waits for the sector pulse, generated by the Analog PCB when it detects a sector mark.

201

CONTROLLER PCB BLOCK DIAGRAM



Controller PCB Circuit Descriptions

202

When the Z8 sees the sector pulse it starts the State Machine.

(For more information on sector headers and Pro-File disk format refer to the Pro-File HDA description in this section.)

Block Diagram For Check Header Operation
The State Machine, in combination with the PAL, moves each successive byte of the header replica into the SERIALIZER, where they are serialized and shifted out in sync with the incoming NRZRDTA.

The target header information and the NRZDTA are compared, and if any difference exists, the State Machine aborts the attempt and resets to wait for the next incoming sector pulse.

The process is repeated until either the header matches the image in RAM or a timeout error occurs (the Z8 program is waiting for the "sector done" from the State Machine). If it doesn't see SECTDN within two revolutions, the Z8 will take over and go through an error recovery routine.)

If the desired operation is to read a block, the logic will accept the data in from disk and will move it into RAM. If the operation is to write a block, the logic will be conditioned to move the data from RAM to the disk.

The circuit description for Check Header function and the Read, Write, and Write/Verify operations are described in the following pages.

203

Refer to the Controller PCB schematic at the end of this section for the following discussion.

Check Header Function Component Explanation

1. After the Command Handshake, if the Z8 firmware has interpreted the command, the first thing it will do is put the location of the target sector (i.e., head, track, and sector) in RAM. The Z8 selects the RAM addresses the same way it did when reading the command bytes in the Command Handshake (with the Address Selectors).

2. If the target sector is on a different head or track than the ones already selected, the Z8 will perform the Seek routine to the target track (the one containing the target sector).

   During the Seek routine, the Z8 issues phases from pins 17-20 through U7 and U8 to the Head Stepper motor on the HDA. It issues the exact number necessary to put the heads where it thinks the the target track is (this is a program function of the Z8).

3. When the head movement is completed, the Z8 enables (loads) the target head.

4. At this time, the Z8 lowers ZRW and AS to cause the PAL to produce the TIMSEL signal out of pin 12 to the Programmable Counter Timer U33. The Z8 then writes commands to the Programmable Counter Timer U33 to program the following registers:

   a. OUT1 register for 555.
   b. OUT2 register for 557.
   c. OUT3 register for 521.

   The counts are shown in the diagram on the opposite page with the events they are timing.

(Continued on the next page.)

204

5. The Z8 then causes the SEL signal out of pin 17 of the PAL to be generated. This enables the RAM to gate the contents of address 0 (since that is the address in the Address Counters at this time) onto the ZR bus to the inputs of U14, the Deserializer Shift register.

   Note: Addresses 0, and 1 both contain all 0's. The sector location information begins with the track at address 2.

6. The Z8 then generates the START signal to:

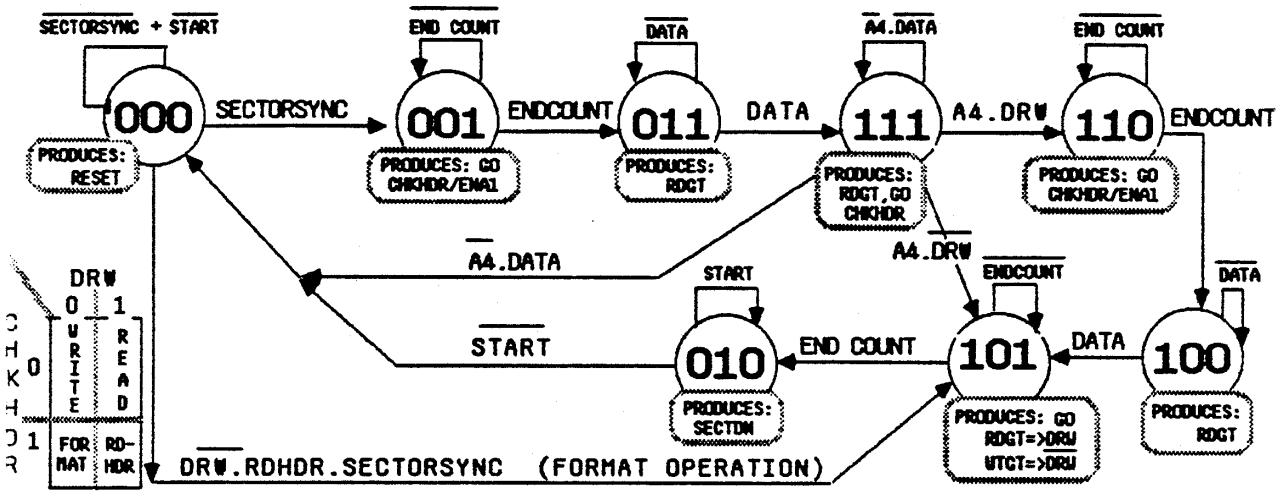
   a. Remove the reset to enable the CRC Error FF.
   b. Remove the reset to enable the storage FFs for the Programmable Counter/Timer in U34.
   c. Remove the reset to enable pin 13 of U39 the System Clock Selector.
   d. Remove the reset to enable the latches in the BYTC (U30) and RWTC (U38) State Machine.

7. When the Analog PCB detects a sector mark (gap) on the target track, it generates the SECTOR signal to pin 30 of the Z8 and pin 15 of U30, the latch for the BYTC State Machine, causing the State Machine to advance from sequence 0 to sequence 1.

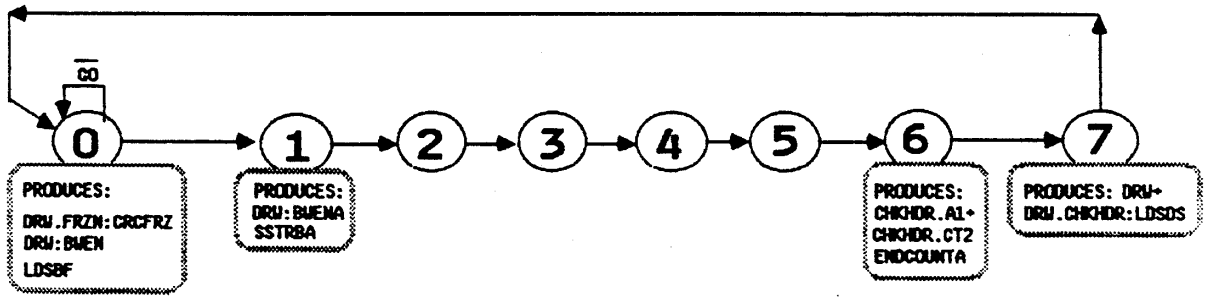
(Continued on the next page.)

205

STATE MACHINE SEQUENCE DIAGRAM



STATE MACHINE PHASE (BIT TIME) DIAGRAM



206

State Machine Operation

8. Each State Machine is composed of a ROM device and a latch. The inputs to the ROM select the ROM address to be read out. The bits in the contents of this address are used as control signals or select inputs to the latch. The outputs of the latch feed back around to the inputs of the ROM to select the next address in the sequence.

9. Various events (i.e., SECTOR, ENDCOUNT, DATA, etc.) determine the sequence of the ROM addresses to be read out and, therefore, which control signals will be enabled. These control signals regulate the sequence of functions that must occur for each operation. The sequences for the Check Header function are discussed under that header.

10. Refer to the diagram at the top of the opposite page for the following discussion. It shows the sequence of State Machine paths and the conditions that determine the next path to be selected. For example, the diagram shows that during sequence 0, the critical signal that the Controller PCB is waiting for is SECTORSYNC (sector mark).

11. Once that signal occurs, depending on whether RDHDR is true or not (RDHDR is only true while formatting the HDA), a path is selected to sequence 1 or 5. During normal operation, RDHDR is low, so the State Machine advances to sequence 1.

    At sequence 1 the critical signal is ENDCOUNT. When ENDCOUNT occurs in sequence 1, the only path is to sequence 3, etc.

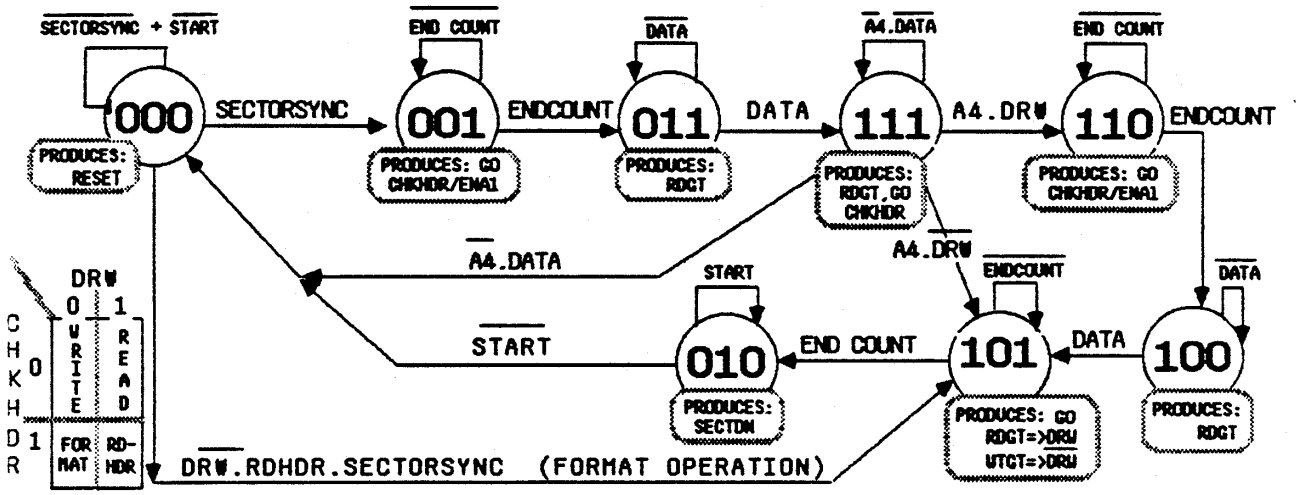
    This diagram is true for all operations. During the Command Handshake, the Z8 interprets the command bytes from the Host to determine the operation requested.

(Continued on the next page.)

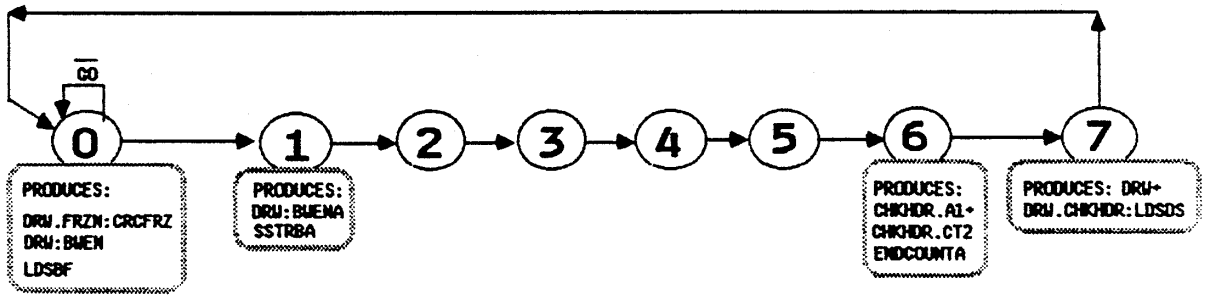
Controller PCB Circuit Descriptions

207

STATE MACHINE SEQUENCE DIAGRAM



STATE MACHINE PHASE (BIT TIME) DIAGRAM



Controller PCB Circuit Descriptions

208

12. As an operation is implemented, the Z8 and PAL control the selection of sequence paths with their inputs to the State Machine (i.e., A4, RDHDR, CHKHDR/ENA1, DRW).

13. The program in the Z8 selects these signals depending on which operation it is trying to implement.

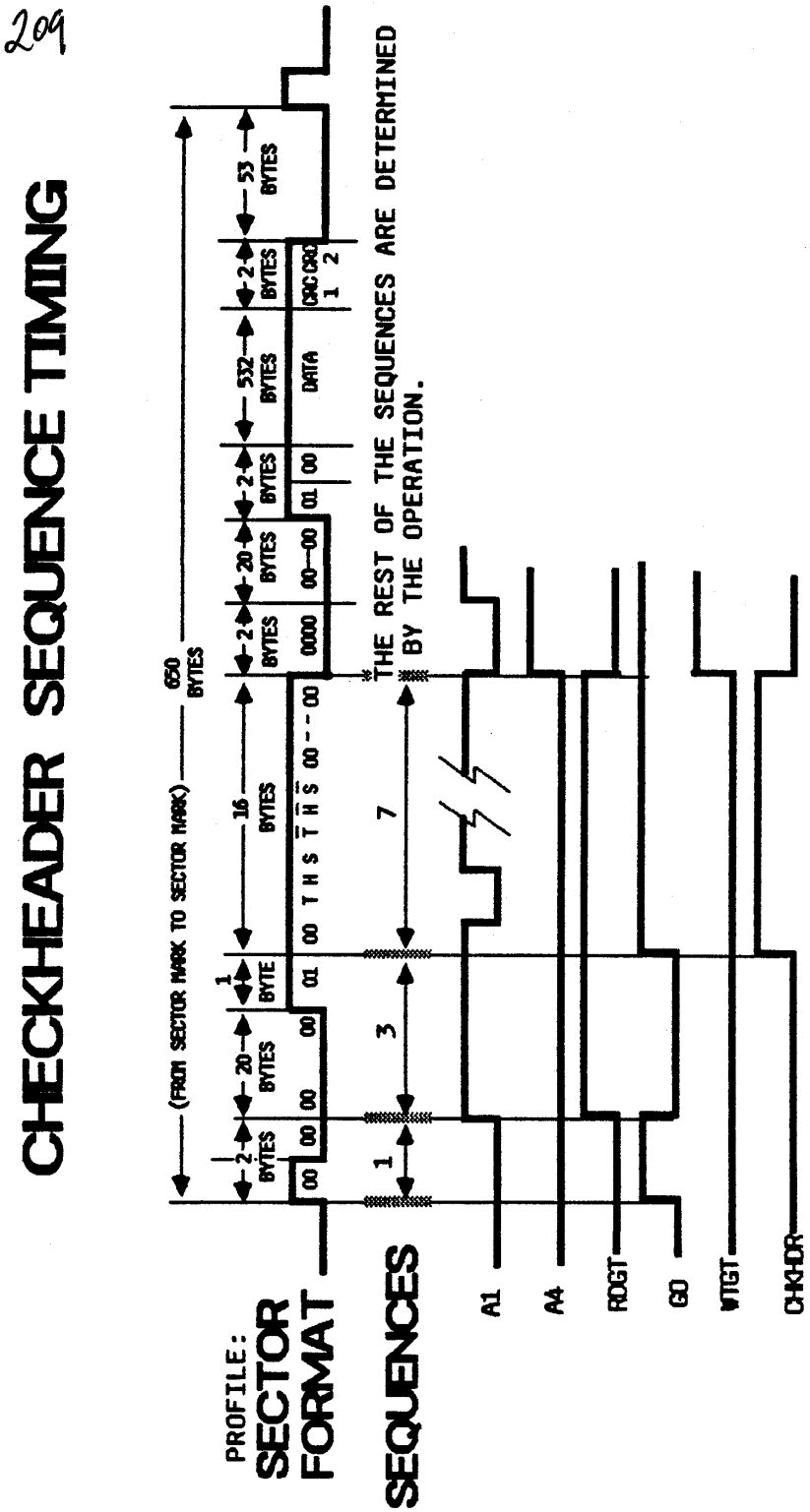
14. The State Machine Phase diagram at the lower part of the opposite page shows when (at which bit time in the processing of a serial byte) a signal may occur during a sequence.

    For example, during sequence 7, every time a complete byte of data has been processed (QC goes high, indicating 8 clocks have been counted by the Divide-by-8 Counter U22) at clock 1 of the next byte, an SSTRB signal is generated out of pin 15 of U38.

    Although neither of the diagrams shows the sequence during which SSTRB will occur, the lower diagram does show the bit time of a byte when it might become active.

(Continued on the next page.)

209



210

15. Sequence 0 is a wait state for the State Machine. When the SECTOR signal occurs to pin 13 of U30, it produces a low to pin 19 of U37, selecting the ROM address that advances the RWTC State Machine to sequence 1.

16. During sequence 1, the GO signal is generated out of pin 14 of U37 (see Timing Diagram). The GO signal enables the Divide-by-8 Counter to count clocks. During sequence 1 the clocks selected by the System Clock Selector U39 are those generated by the crystal oscillator.

17. To review what has happened so far: We are still in the Check Header function. The heads have finished their movement, and the proper head has been enabled. Now the Z8 wants to confirm that the target head is in fact over the target track, which it does by reading three consecutive sector headers and comparing their header information with target head and track information currently in RAM.

    After the initial setup, the Controller PCB waited for the Analog PCB to detect a sector mark that signified that the beginning of a new sector was about to pass under the selected head. When the SECTOR pulse occurred, it advanced the State Machine to sequence 1.

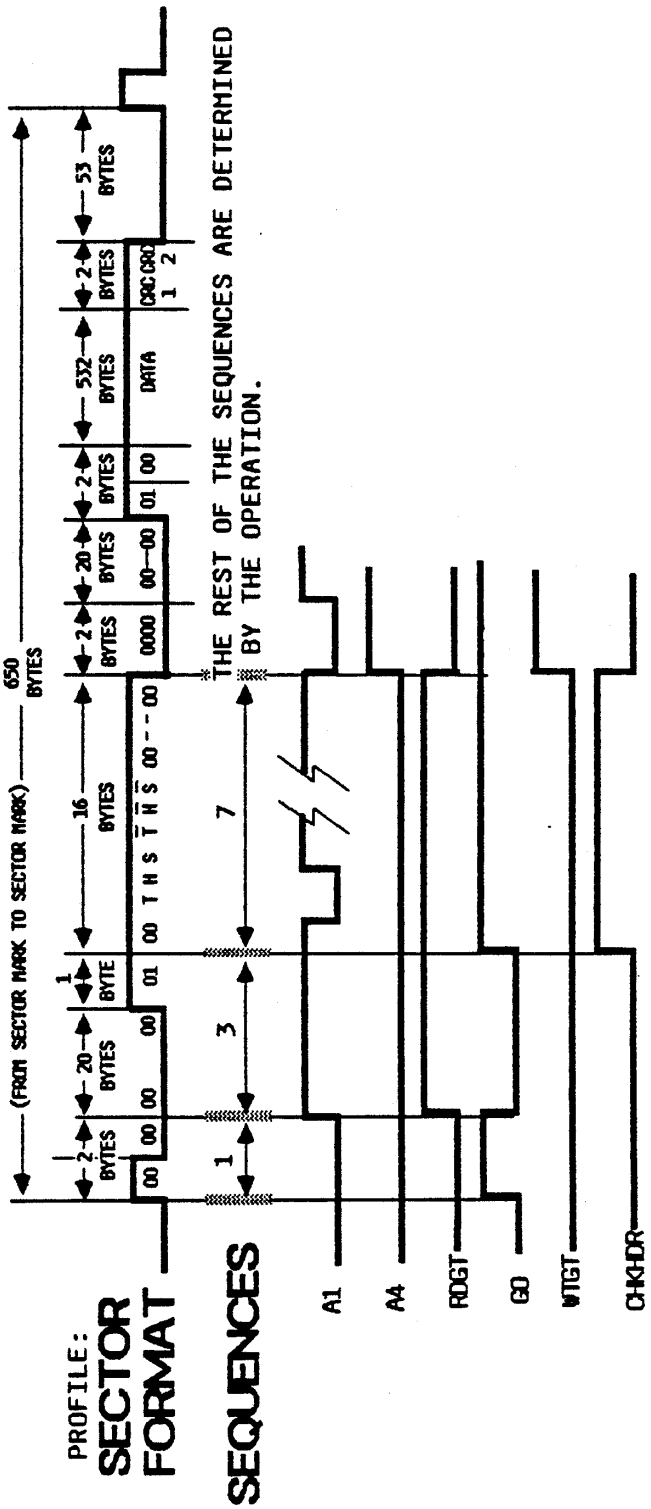
    The sector header is the first part of the sector after the first sync byte field. The Z8 doesn't want to read the first two bytes of any sync byte field because they are unreliable.

    So sequence 1 is simply a control state to hold the circuitry for the first two sync bytes, in case spurious data occur during that period. (For more information on the composition of a sector, refer to the Format description of the Pro-File HDA Description in the Appendices section.)

18. Every time the Divide by 8 counter counts 8 bit-clocks it produces a positive transition on its QC output. This signal goes to the clock input of the Programmable Timer (U33), which decrements each of its counters once for every pulse.

211

CHECKHEADER SEQUENCE TIMING



212

19. Each QC also goes to the State Machine, causing U38 to produce the SSTRB pulse out of pin 15 at phase 1 of the following byte. This signal goes to the PAL on pin 7, causing it to increment to the next Address.

20. The SSTRBs from the first two sync bytes cause the Address Counters U10, 16, and 17 to increment to a count of 1 (from 0). This causes the signal A1 to go to pin 5 of U29, causing the State Machine to produce the ENDCOUNT signal putting the State Machine in sequence 3.

21. Sequence 3 allows time for the VCO on the Analog PCB to sync to the sync byte field. While this is happening the Controller PCB is waiting for data (in this case the sector header). In sequence 3, the RDGT signal (pin 7 of U38) is generated by the State Machine.

    This signal goes to the Analog PCB to enable it to convert the analog signal from the HDA head into NRZ data (NRZRDTA) and to generate RDCLKs in sync with this data.

    RDGTA also goes to the System Clock Selector U31 and U32 to select RDCLKs as the system bit clock.

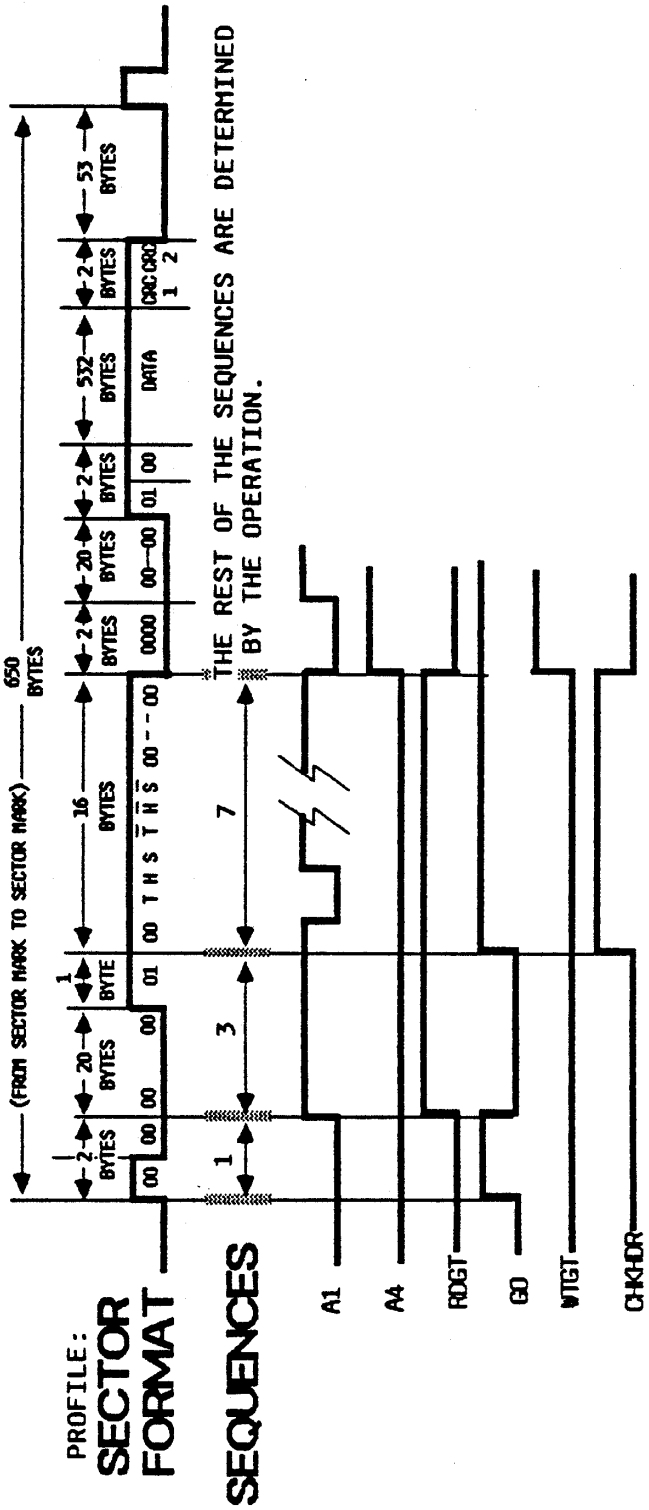
22. Sequence 3 also causes the GO signal to go low from pin 14 of U37 in the State Machine. The low GO signal resets the Deserializer chip (U14). This condition causes a low output from pin 17 of U14, which goes to pin 13 of the Exclusive Or in U28.

23. Pin 12 of U28 is NRZRDTA. At this point in reading the sector, NRZRDTA should be all 0's (because the sync bytes are all 0's), so pin 12 should be low, which, along with the low pin 17, results in a low out of pin 11 of U28.

24. Pin 11 remains low until the sync byte just before the sector header (which contains a 1) occurs; now the inputs to the Exclusive Or are different, making DATA signal on pin 11 high.

213

CHECKHEADER SEQUENCE TIMING



PROFILE: SECTOR FORMAT SEQUENCES

214

25. As can be seen from the State Machine Sequence diagram, the high DATA signal is the condition that causes the State Machine to advance to sequence 7.

    The purpose of sequence 7 is to read the sector header field of a sector. Recall that the Address Counters are pointing to RAM address 01, which contains 0's. The Z8 has the first byte of the target sector header field waiting on the ZR bus (this byte is always a 00).

26. When the State Machine advances to sequence 7:

    a. The LDSDS out of pin 12 of U29 goes high, causing the Deserializer to load the 00 from RAM address 1 (the first byte in the sector header should also contain a 00).
    b. Every QC output from the Divide-by-8 Counter (happens once per byte) to pin 3 of U29 will cause pin 8 to go high, causing the SSTRB signal out of pin 15 of U38.
    c. The CHKHDR signal out of pin 10 of U38 goes high, enabling the PAL to pass SSTRBs through as Address Counter increment pulses to the Address Counters.

27. NRZRDTA that comes in during sequence 7 should be sector header information. The first byte of target sector header information in the Deserializer is clocked out in sync with the NRZRDTA coming in from the Analog PCB.

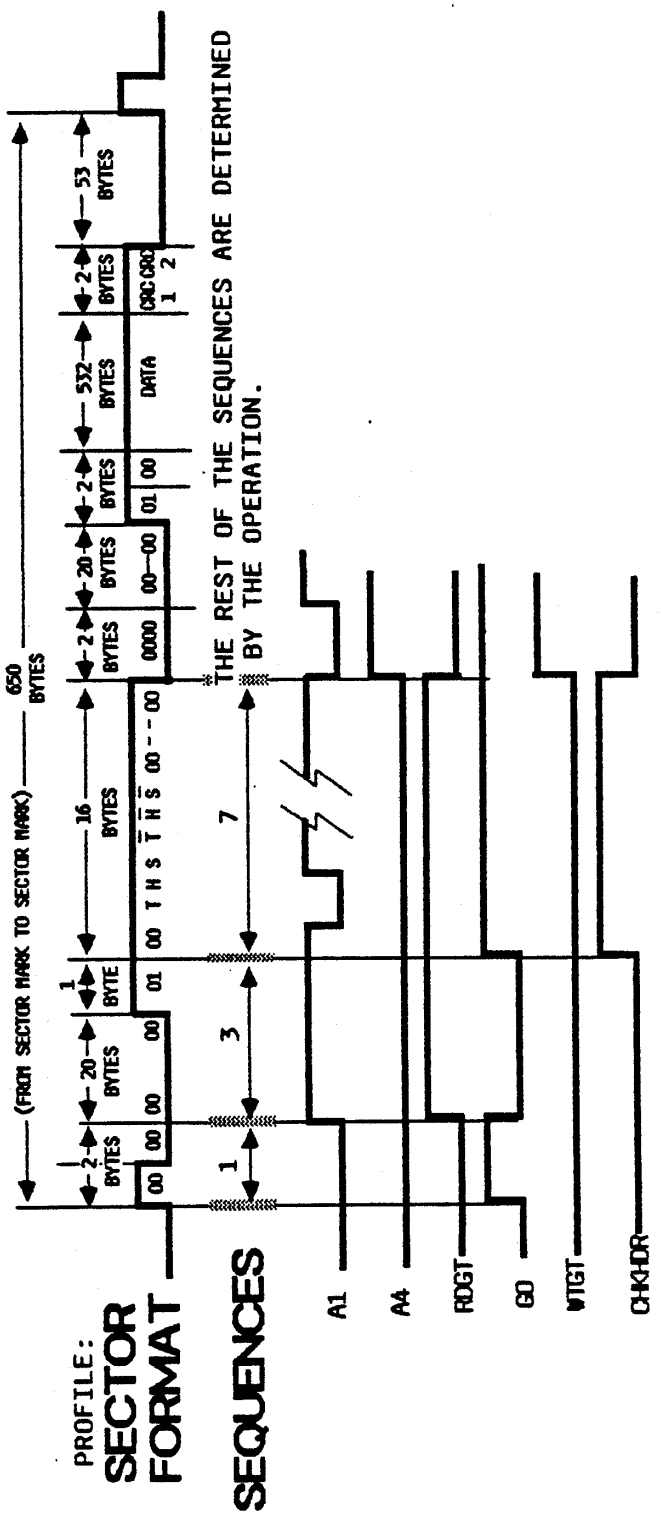
    These two signals are compared in the Exclusive Or in U28. If any bit does not match up, this gate will output high causing the State Machine to generate a SECTDN signal out of pin 2 of U30, resetting the Read/Write Control circuitry to sequence 0.

28. If the first byte is correct, the QC output from the Divide by 8 chip U22 will cause an SSTRB to increment the Address Counters at bit 1 of the next byte.

    This signal puts the next byte of target sector location information on the ZR bus. At bit 7 of that byte, if the comparison is still OK, LDSDS will occur to load the second byte of target sector location information into the Deserializer chip.

215

CHECKHEADER SEQUENCE TIMING



PROFILE: SECTOR FORMAT SEQUENCES

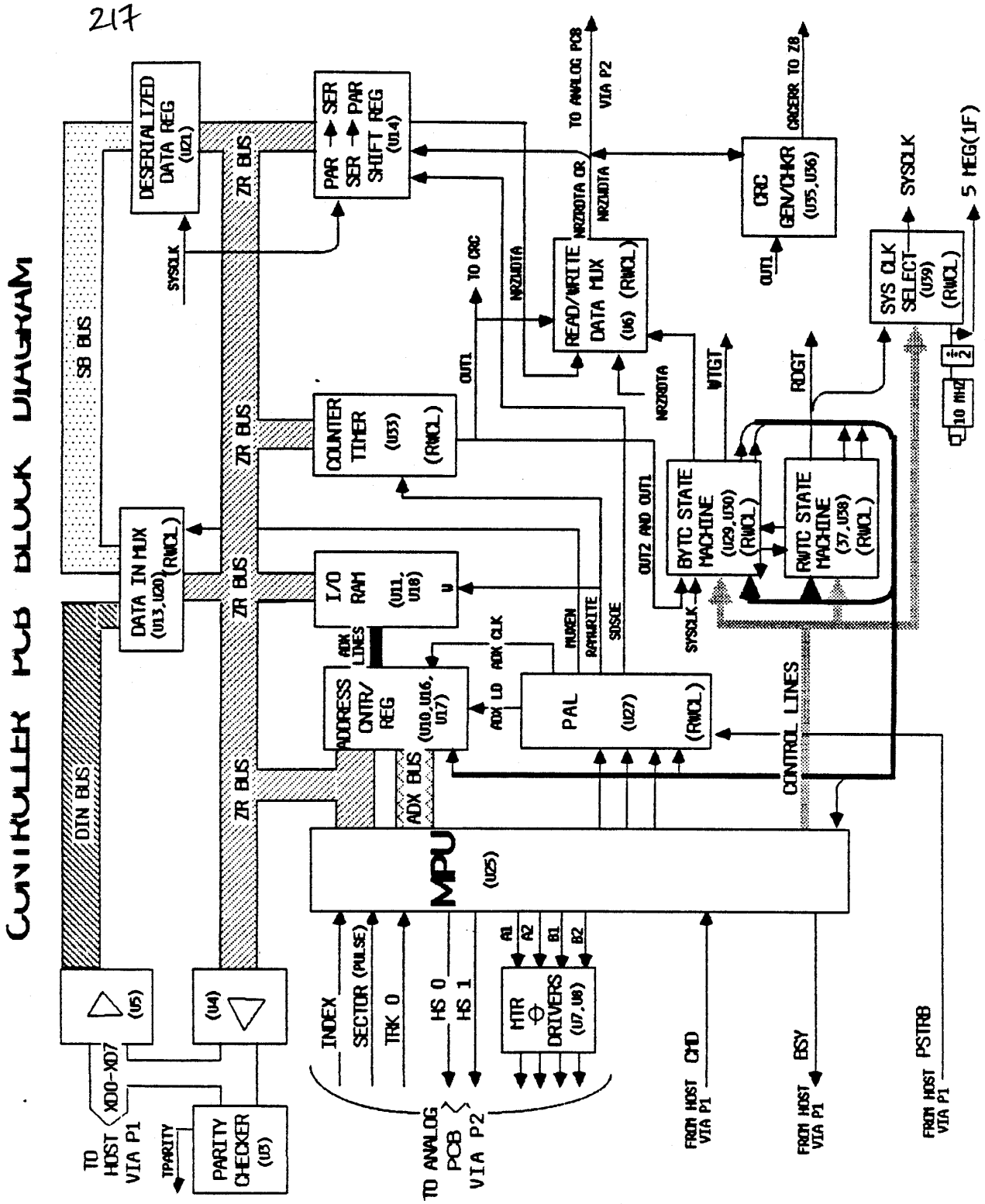
216

29. The 16 bytes of target sector header information were stored in 16 memory locations of RAM. When the Address Counters increment past 16, A4 goes high, which signifies to the State Machine that a complete sector header has been checked and must be OK because the DATA signal must have remained low.

    Signal A4 causes the State Machine to advance to sequence 5 or 6, depending on whether DRW is high or low, which depends on which operation the Host requested.

30. If DRW is high, the State Machine will go to sequence 6 for a Read operation. If DRW is low the State Machine will go to sequence 5 for a Write operation.

217



218

3. Operations

The first part of all operations is the Check Header function (previously described). The following discussions deal with each of the three Pro-File operations after the Check Header function has been performed, which is after sequence 7 of the Read/Write control circuitry.

Once the Seek to the target track is confirmed, and the Z8 has a reference on where the target track is, the Z8 will condition the circuitry for the execution of an operation a number of sectors before the target sector, called the sector interleave (because the data come off the disk too fast for the circuitry on the Controller PCB to process the target sector's header).

The sector interleave is specific to the Host operating system using the Pro-File and to the operation being performed.

After the sector interleave, when the sector pulse for the target sector occurs, the Z8 enables the control logic to activate the circuitry for the operation requested.

3.1.1 Read Operation General Explanation

During a Read operation, after the Check Header function, NRZDTA enters through the Read Data MUX into the Deserializer.

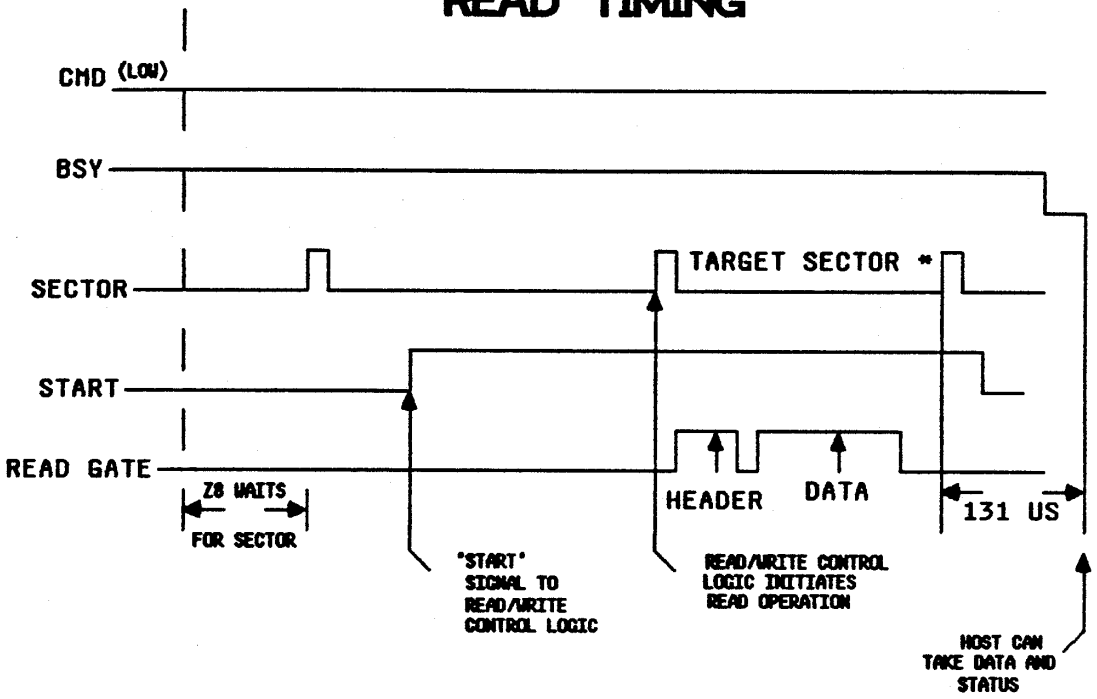
In the Deserializer, NRZDTA is converted to parallel bytes, and then transferred to the Deserialized Data Register, held momentarily, and gated through the Data In MUX to the RAM.

When the sector (block) is complete, the control logic tells the Z8 to lower the BSY line. It does, and then the Host uses PSTRB to strobe the data out of RAM through the output buffer to the Host interface PCB.

The RAM address counter has been set to the beginning of the data, and each strobe sent by the Host increments the Address Counters to the next address.

219

READ TIMING



IF THE HEADER READ, IS NOT FROM THE TARGET SECTOR, THE HARDWARE CONTINUES TO READ SECTOR HEADERS UNTIL THE TARGET HEADER IS FOUND. BSY, AND START REMAIN HIGH.

220

Refer to the Controller PCB schematic at the end of this section for the following discussion.

3.1.2 Read Operation Component Explanation

1. As can be seen in the State Machine Sequence diagram after sequence 7 (the end of the Check Header function), if DRW is high (indicating the command bytes from the Host specified a Read operation) and A4 goes high (indicating all 16 bytes of the sector header field have been checked), the State Machine will advance to sequence 6.

2. Sequence 6 holds the Read/Write control circuitry in a wait state during the first two bytes of the sync byte field preceding the data field. These first two bytes can contain spurious data that could cause an error.

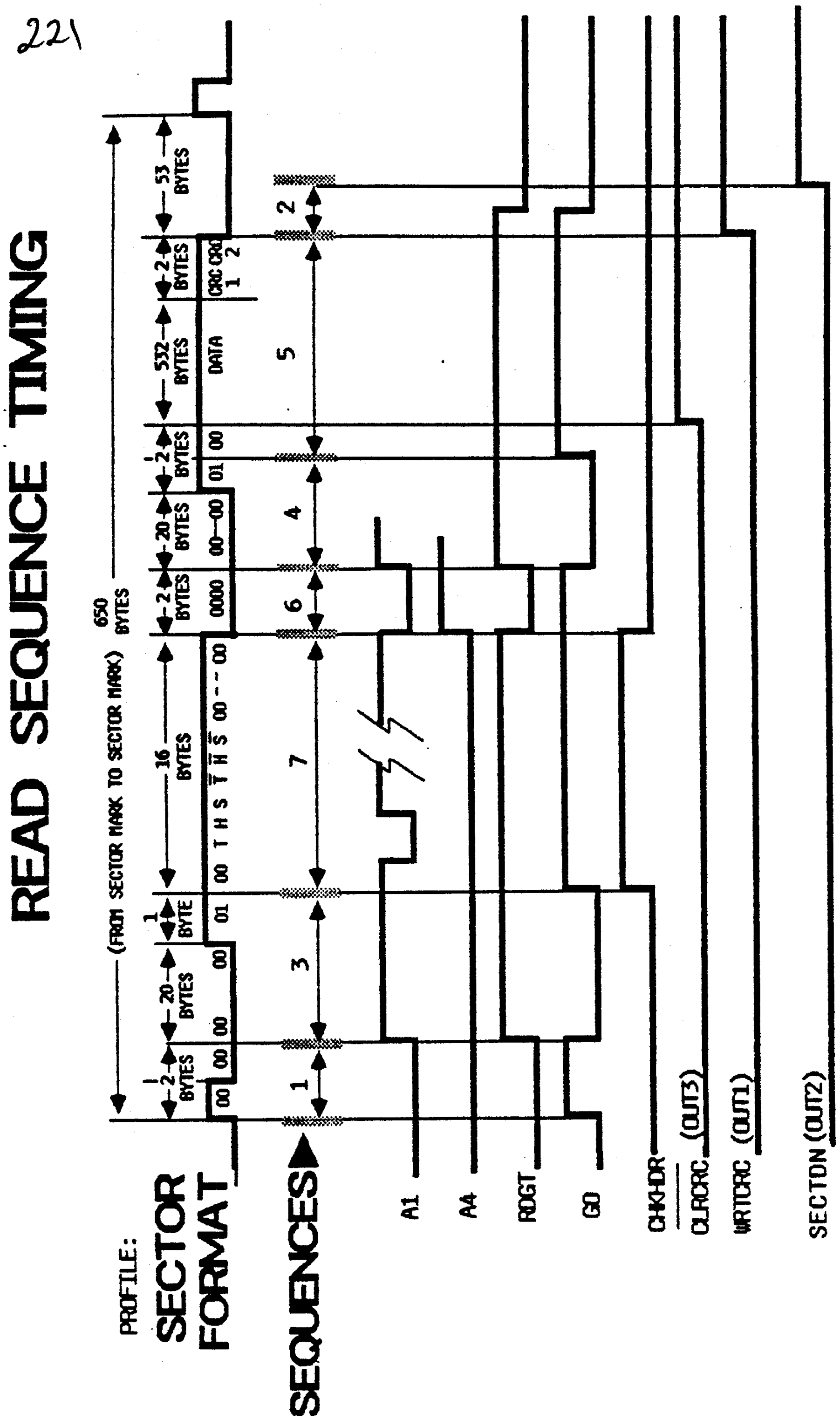
3. Because RDGT is not produced in sequence 6, the System Clock Selector U31 and U32 selects clocks from the crystal oscillator.

4. After two bytes of these clocks have been received, the A1 signal from the Address Counters goes true, to pin 5 of U29. This causes the BYTC (Byte Control) State Machine to produce ENDCOUNT out of pin 10 of U30, putting the State Machine in sequence 4 (as shown in the State Machine Sequence diagram).

5. Sequence 4 allows time for the VCO on the Analog PCB to sync to the sync byte field. While this is going on the Controller PCB is waiting for data (in this case the data field). In sequence 4, the RDGT signal (pin 7 of U38) is generated by the State Machine. This signal goes to the Analog PCB to enable it to convert the analog signal from the HDA head into NRZ data (NRZRDTA) and to generate RDCLKs in sync with this data. RDGTA also goes to the System Clock Selector U31 and U32 to select RDCLKs as the system bit clock.

6. Because the GO signal is low again at the beginning of sequence 4 the Deserializer chip (U14) is reset. This condition causes a low output from pin 17 of U14, which goes to pin 13 of the Exclusive Or in U28.

Controller PCB Circuit Descriptions



222

7. Pin 12 of U28 is NRZRDTA. At this point in reading the sector, NRZRDTA should be all 0's (because the sync bytes are all 0's), so pin 12 should be low, which, along with the low pin 17, results in a low out of pin 11 of U28.

8. Pin 11 remains low until the sync byte just before the data field (which contains a 1) occurs; now the inputs to the Exclusive Or are different, making DATA signal on pin 11 high .

9. As can be seen from the State Machine Sequence diagram, the high DATA signal is the condition that causes the State Machine to advance to sequence 5.

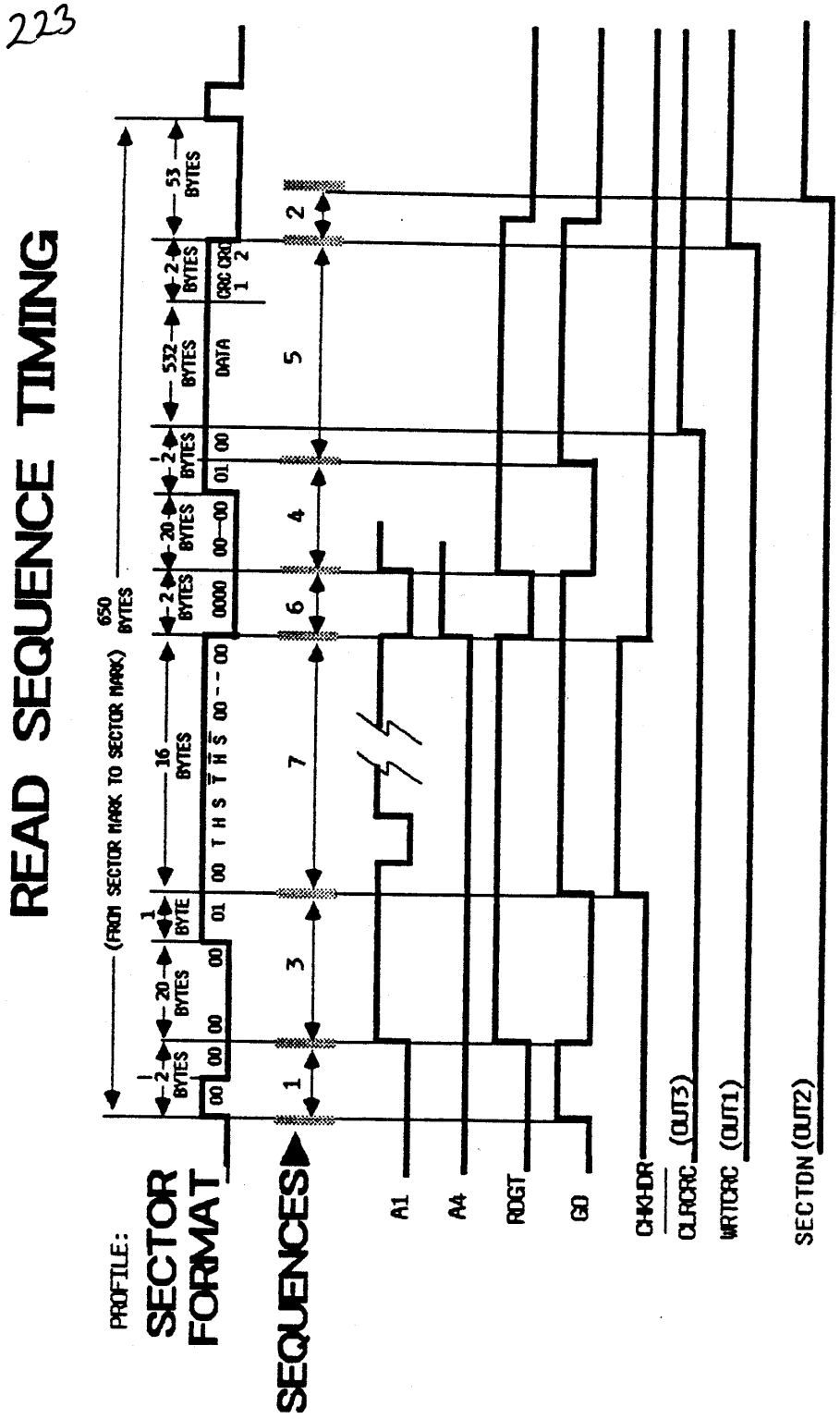
10. Sequence 5 generates either RDGT or WTGT, depending on the level of DRW. If DRW is high, the command bytes from the Host have stipulated a Read operation, so RDGT will remain high. If DRW is low, WTGT will occur. Sequence 5's function during Read is to read the data field from the target sector on the HDA.

11. The GO signal goes high at sequence 5 to enable the Divide-by-8 Counter U22 to begin counting RDCLKs again and to produce QC pulses once per byte to decrement the counting registers in the Programmable Counter/Timer (U34).

12. Since NRZDTA is in serial form coming from the Analog PCB, it must be deserialized and sent to RAM in bytes. This is done by the Serial/Deserial Shift Register U14. NRZRDTA comes in to pins 5 and 6 of the Read Data MUX U6. Because this is a Read operation, pin 2 is low, enabling the Read Data MUX to pass serial NRZRDTA to pin 11 of the Serial/Deserial Shift Register U14.

13. The NRZRDTA is clocked into the Serial/Deserial Shift Register U14 with RDCLKs at pin 12. At the phase 0 following a byte of data, the contents of the Serial/Deserial Shift Register U14 are enabled to merge onto the ZR bus by the signal SDSOE from pin 13 of the PAL U27. This timing relationship can be seen on the State Machine Phase diagram (shown in the Check Header function circuit description).

223



224

14. LDSBF (from pin 12 of U29 the State Machine) also occurs at phase 0, to load the byte from the ZR bus into the Deserialized Data Register U21.

15. Once U21 contains a byte of received data, its contents merge on the SB bus. The Data In MUXers U13 and U21 are enabled by the high MSEL1 signal from the Z8 to pin 1 to select the SB bus, and so pass the byte to the data input of the RAM.

16. In sequence 5, the RWTC State Machine produces SSTRBs each phase 1 from pin 15 of U38. This happens once per byte, triggered by the inputs from the Divide-by-8 Counter U22.

17. During sequence 5, the PAL U27 generates the low RAMWRITE signal, on pin 16, to the RAM each time SSTRB occurs. This enables the byte of information from the Data In MUXers U13 and U21 to be stored into the address currently being selected by the Address Counters.

18. The SSTRBA signal also increments the Address Counters U16, U17, and U10 after storing each byte received from the disk.

CRC Operation

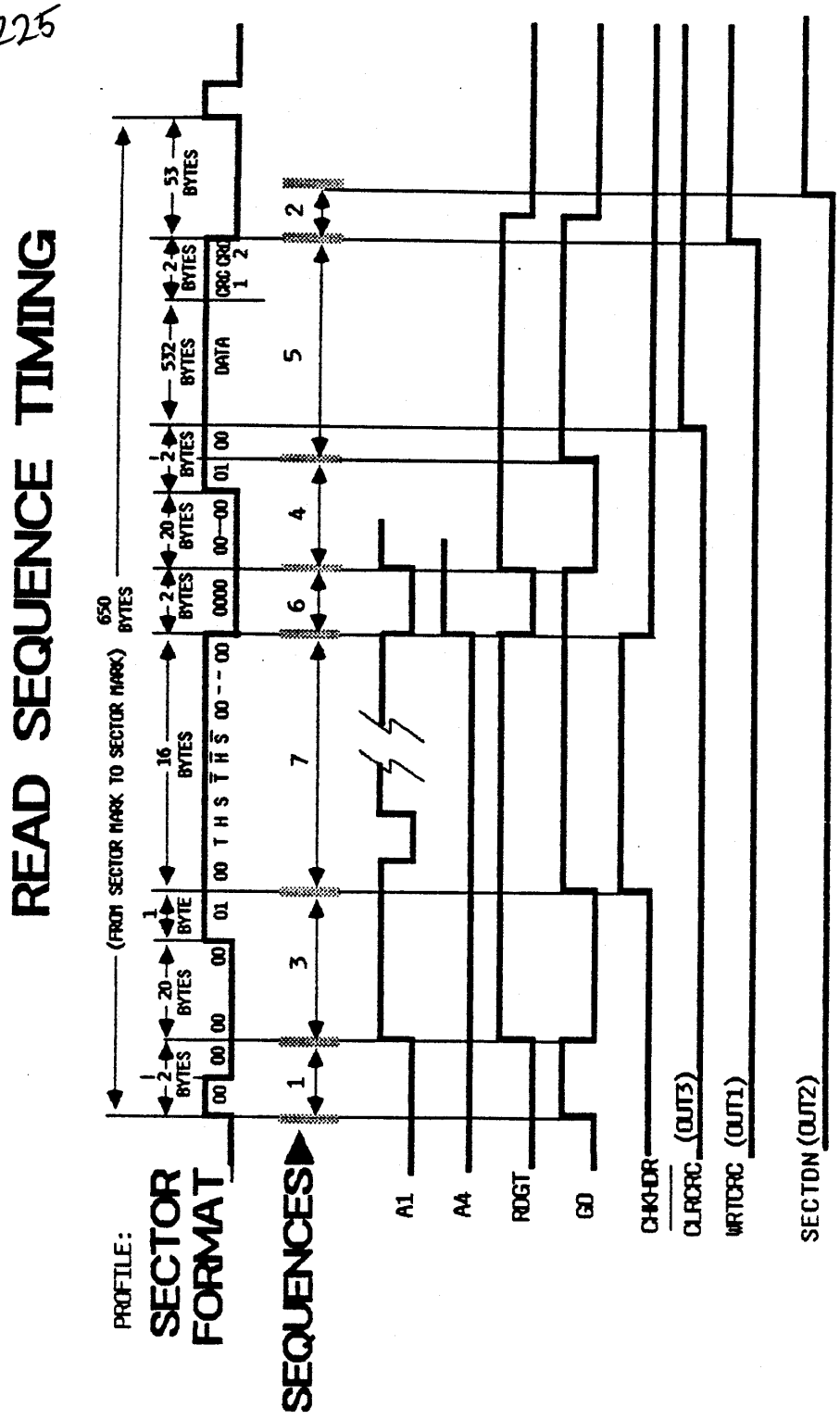
19. Prior to sequence 6, during the Check Header function, the Z8 reprogrammed the Programmable Counter/Timer (U34) so that:
    a. OUT1 register contained 553.
    b. OUT2 register contained 555.
    c. OUT3 register contained 19.

    The Programmable Counter/Timer (U34) is enabled to decrement all of its registers each time an SSTRB occurs, but only while GO is high.

    During the sequences of the first part of the Read operation (including the Check Header function), each of these registers is decremented by the following amounts.

Sequence	Amounts
1	2
7	16

225



226

This brings the amounts in each register to:

a. OUT1 register contains 535.
b. OUT2 register contains 537.
c. OUT3 register contains 1.

20. The second byte received during the data field of sequence 5 is never data but instead is a 00. The GO signal is high for this byte, so the SSTRB generated during its reception also decrements the registers in the Programmable Counter/Timer (U34). This causes the OUT3 register to arrive at 0, causing CLRCRC to go high on pin 1 of an Exclusive Or gate in U28.

NOTE: Although the three U28 gates inputting to the CRC circuit are shown to be exclusive Ors, they are being used as inverters.

21. The other input to this gate (pin 2) is constantly high. Having two highs causes the gate to output a low, removing the high reset on pin 4 of the CRC Generator/Checker U35.

22. NRZRDTA is going to pin 11 of the CRC Generator/Checker U35. This chip is now enabled to use its internal logic to process each received bit from the data field and reflect its CRC status with its output on pin 13. If an error occurs, pin 13 of the CRC Generator/Checker U35 will go high, setting the CRC Error FF U36, which produces the low CRCERR signal to pin 30 of the Z8.

    The Z8 then knows it has faulty data, and will perform an error recovery routine. (For more information on the error recovery routines used in the Pro-File's firmware, refer to the Appendices section of this manual.)

23. The OUT1 register is also decremented by the second byte in the data field, bringing its contents to 534.

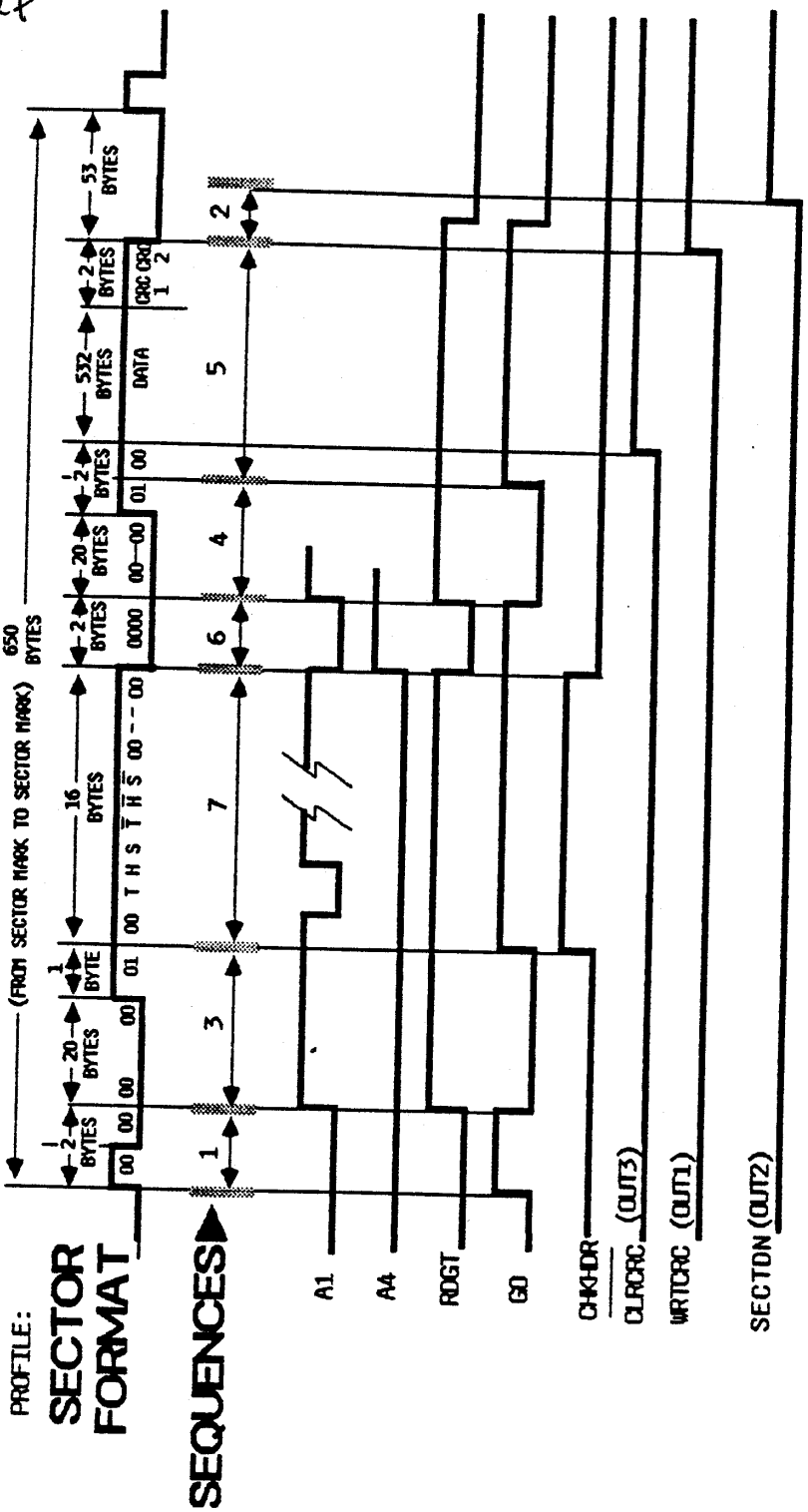
    After the remaining 532 bytes of data, and the two CRC bytes are read from the data field, the OUT1 register completes its count causing:

    a. The CRC Generator/Checker U35 to freeze its status.

    b. The State Machine to generate ENDCOUNT and advance to sequence 2.

227

READ SEQUENCE TIMING



228

24. Sequence 2's purpose is to hold the Read/Write Control Logic in a static state until the Z8 is finished processing the CRC status. This happens when OUT2 decrements to 0, causing the State Machine to output the SECTDN pulse.

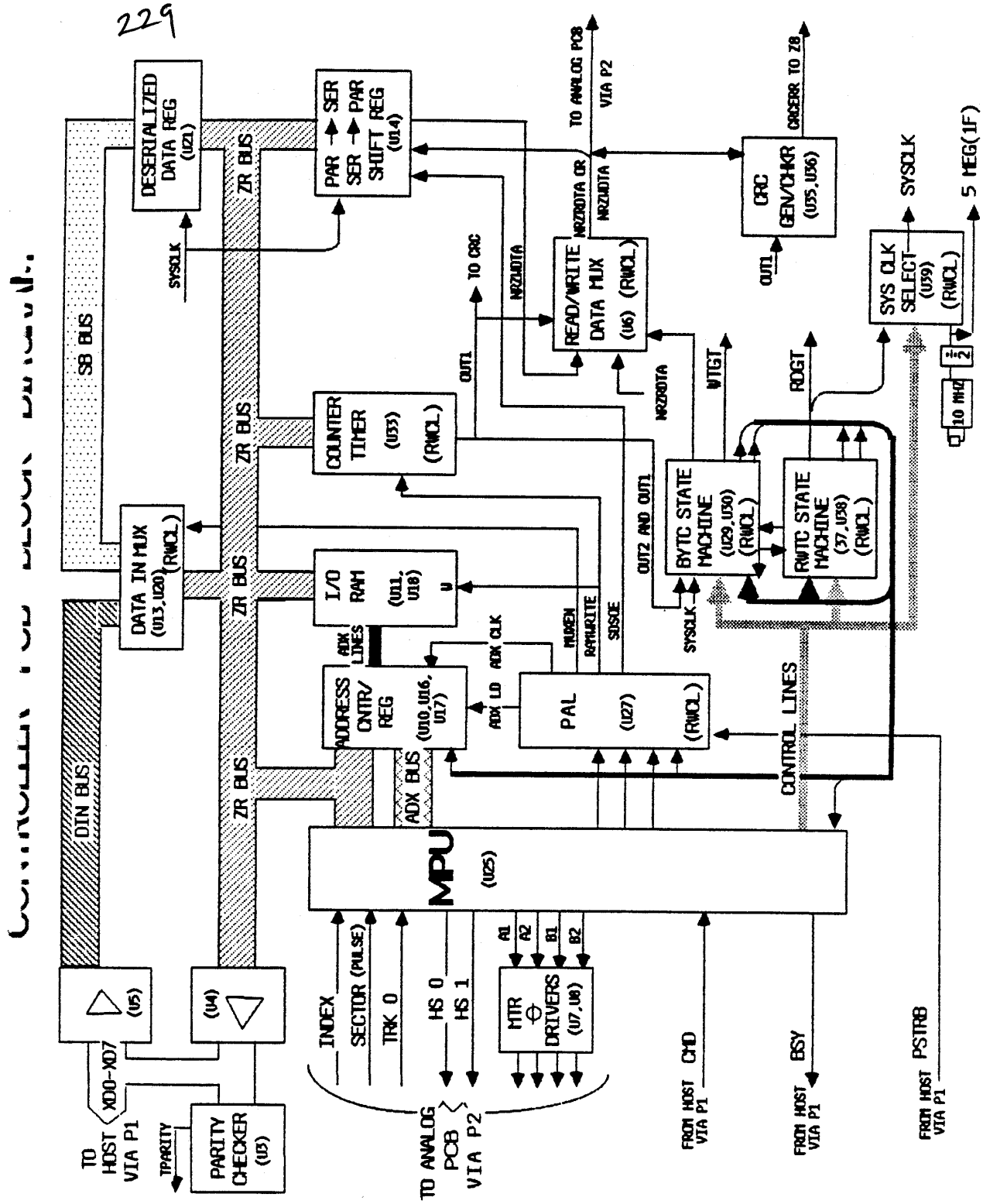
    This pulse accomplishes two functions:

    a. It resets the Programmable Counter/Timer (U34).

    b. It goes to pin 31 of the Z8.

    When the Z8 receives the SECTDN pulse, it samples the CRC status from U35 and, when finished processing, lowers the START signal, causing the Read/Write Control circuitry to reset to sequence 0.

229



230

3.2.1 Write Operation General Explanation
There are 3 handshakes for the Write or Write/Verify operations.

After completing the Command Handshake (the first handshake), the Z8 conditions the Controller PCB logic to accept a block of write data from the Host.

During the second handshake, the Host strobes each byte of data into the RAM with PSTRBs.

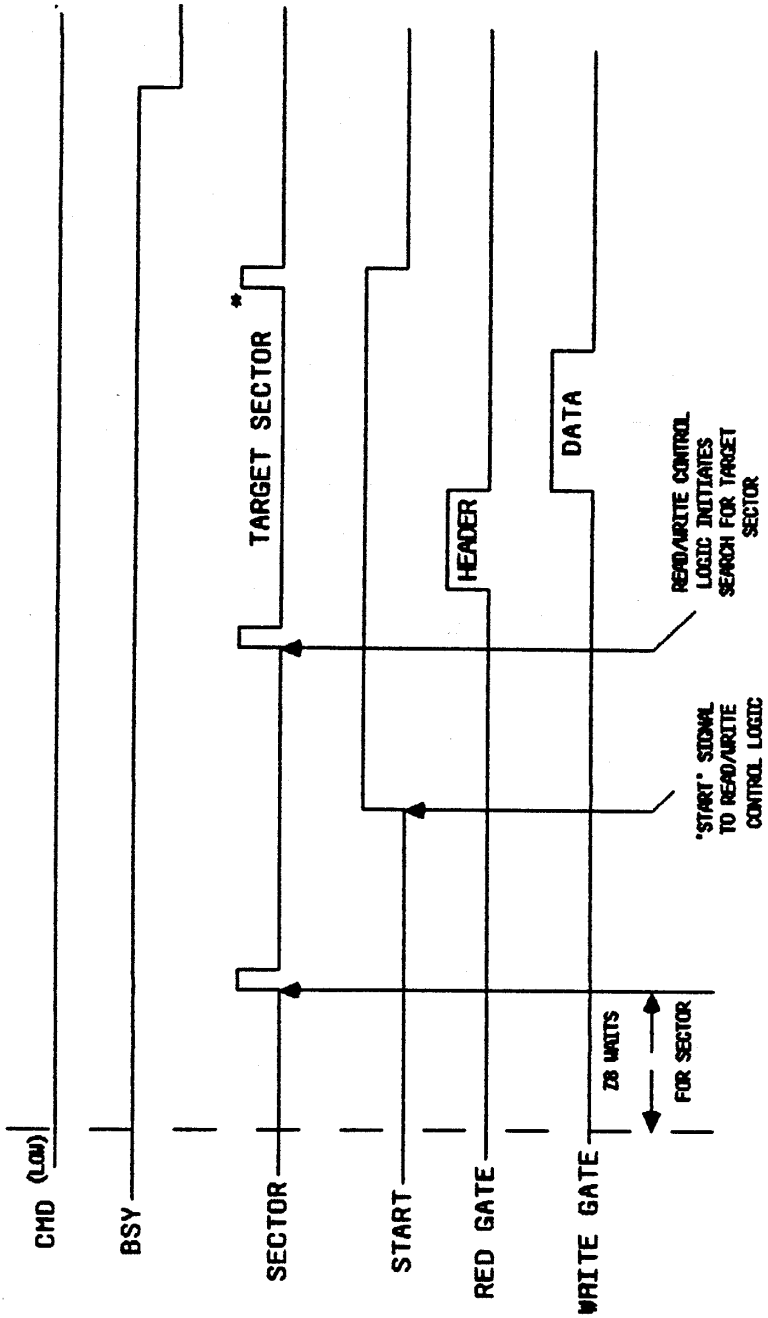
Then the Z8 and Read/Write Control Logic read the parallel write bytes out of the RAM to the Serial/Deserial Shift Register U14. U14 is then enabled to shift these bytes out in serial to the Analog PCB.

During a Write operation the WTGT signal enables the Analog PCB to convert this serial NRZ data to MFM data and then to an analog signal to be written on the disk surface. Once the Write is complete the Pro-File will go through yet another handshake.

The third handshake allows the Pro-File to send the completion status for the operation back to the Host.

231

WRITE TIMING



* THIS SECTOR PULSE IS TEMPORARILY SUPPRESSED BY WRITE OPERATION.

232

Refer to the Controller PCB schematic at the end of this section for the following discussion.

3.2.2 Write Operation Component Explanation

The following discussion explains the Write operation only so far as it differs from the Check Header function.

1. As can be seen in the State Machine Sequence diagram following sequence 7 (the end of the Check Header function), if DRW is low (indicating the command bytes from the Host specified a Write operation), and A4 goes high (indicating all 16 bytes of the sector header field have been checked) the State Machines will advance to sequence 5.

2. Sequence 5 will generate either RDGT or WTGT depending on the level of DRW. IF DRW is low, that means the command bytes from the Host stipulated a Write operation so WTGT will go high.

   If DRW were high RDGT would occur. Sequence 5's purpose during a Write is to read the parallel write data out of RAM (this data was sent to the Pro-File during the second handshake of a Write operation), convert it to serial NRZ write data, and send it to the Analog PCB in sync with the system clock from the crystal oscillator.

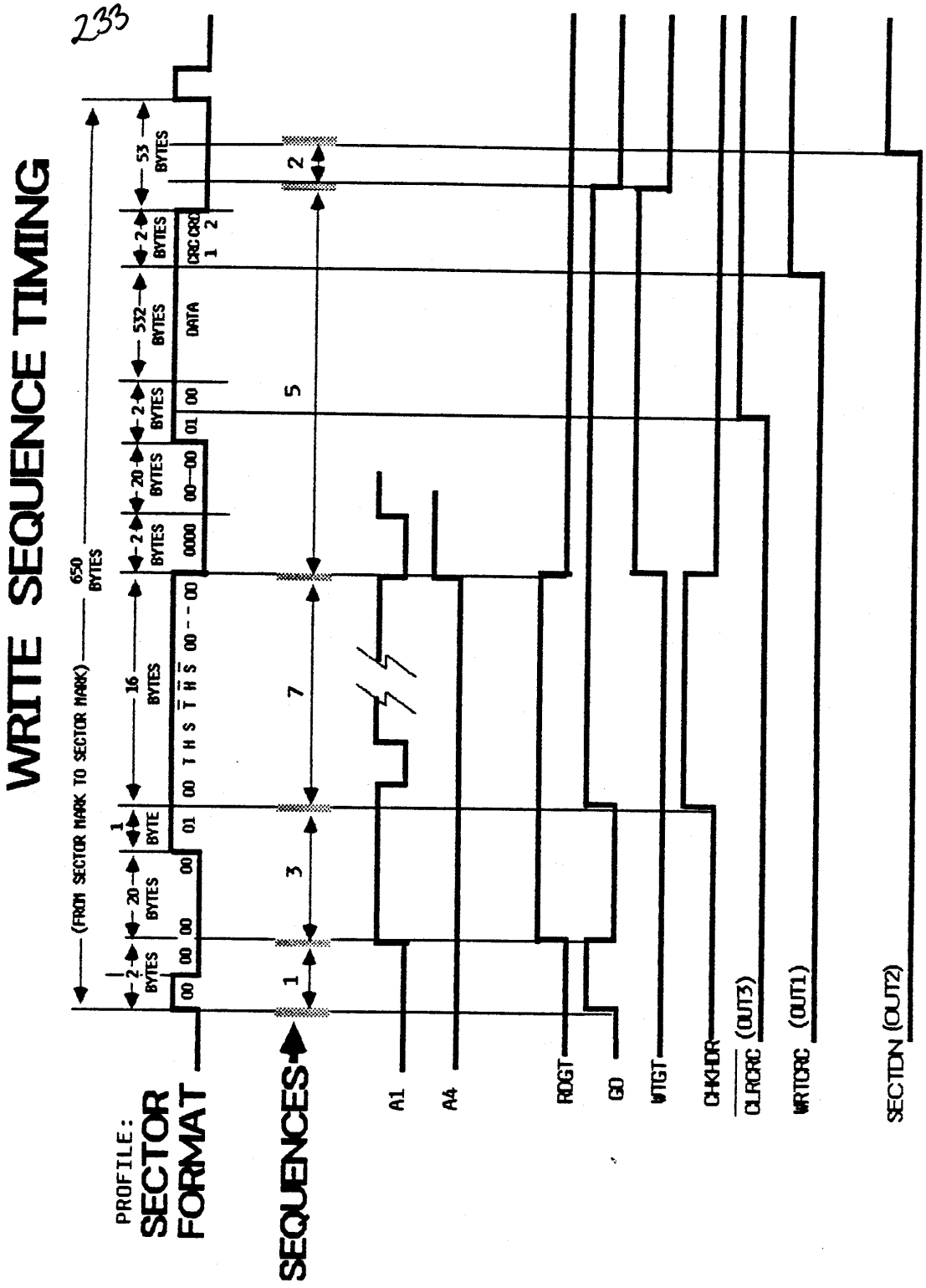
3. Because RDGT is not being generated the System Clock Selector U31, and U32 will select the output from the crystal oscillator to use as the system bit clock for the rest of the Write operation.

4. The WTGT signal, pin 5 of U30, is generated by the State Machine as a result of being in sequence 5. This signal goes to the Analog PCB to enable it to convert the serial NRZ write data to serial MFM data, and then to an analog signal for the selected head to store on the disk surface.

6. After sequence 7 the Address Counters will be set to the RAM address of the first byte in the data block to be written.

   The write data block in RAM is composed of 22 bytes of zeroes, then a sync byte containing a 1, then another zero byte, and then 532 bytes of data. During sequence 5, each write byte is read sequentially from the RAM.

233



234

7. Because the State Machine is in sequence 5, the Divide-by-8 Counter U22 will be enabled by the high GO signal to count system bit-clocks (since RDGT is low the System Clock Selector U39 will select the clocks from the Controller PCB crystal oscillator).

   Each QC output from the Divide-by-8 Counter U22 will produce an LSDS output from pin 12 of U29 in the State Machine to pin 19 of the Serial/Deserial Shift Register U14 enabling it LOAD a byte of write data from the ZR bus. As can be seen on the State Machine Phase diagram (shown in the Check Header function circuit description) LDSDS occurs at phase 7 of a byte to load the next byte from the bus.

8 In sequence 5 the RWTC State Machine will produce SSTRBs from pin 15 of U38. This happens once per byte as a result of the inputs from the Divide-by-8 Counter U22.

9. As can be seen on the State Machine Phase diagram, SSTRBA will be generated at phase 1 following the generation of LDSDS. And so the SSTRBA signal increments the Address Counters U16, U17, and U10 after the Serial/Deserial Shift Register U14 has accepted one from the ZR bus.

10. Prior to sequence 5 of the Write operation, during the Check Header function, the Z8 reprogrammed the Programmable Counter/Timer (U34) such that:
    a. OUT1 register contained 573.
    b. OUT2 register contained 575.
    c. OUT3 register contained 39.

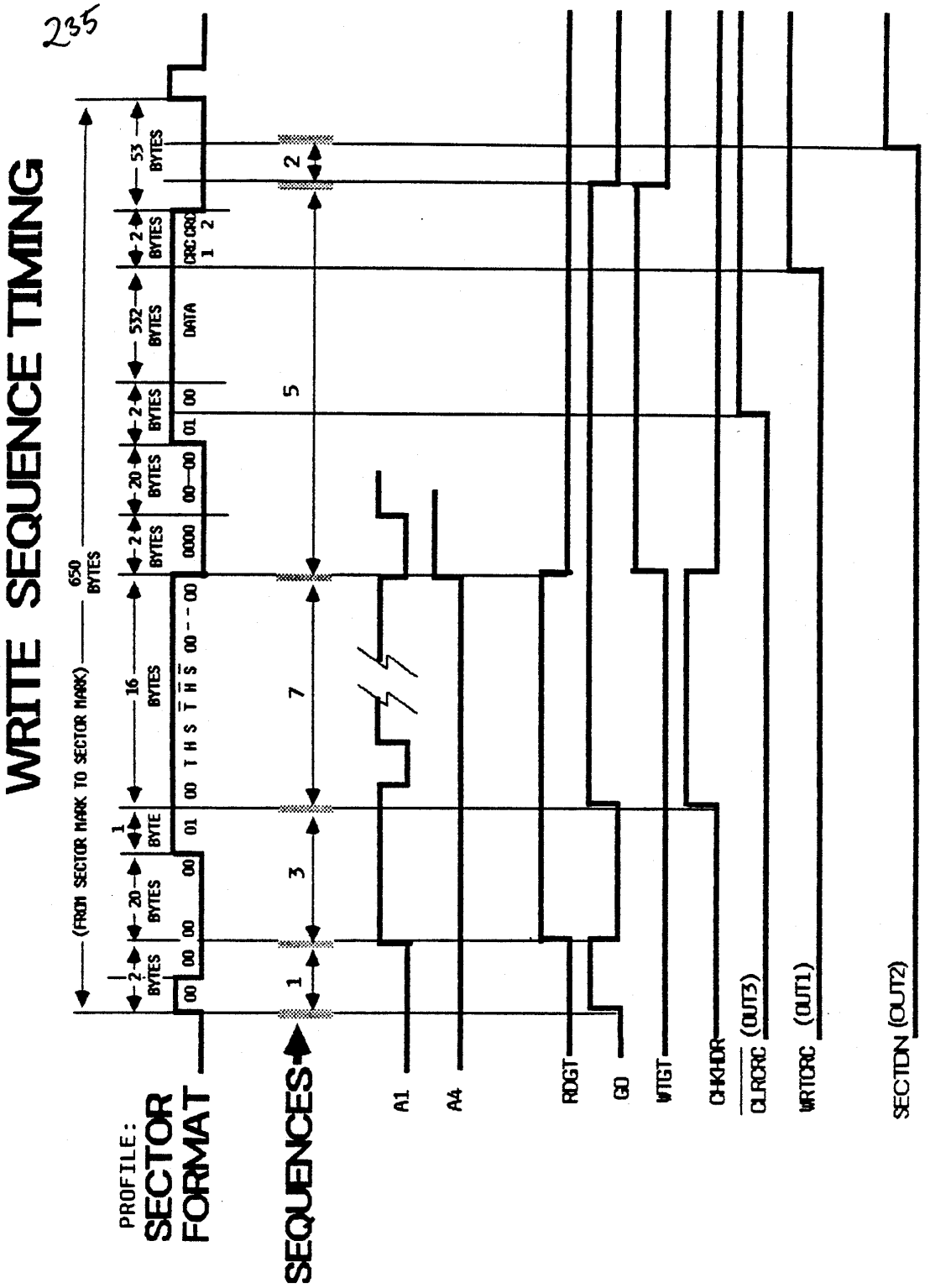
    Programmable Counter/Timer (U34) is enabled to decrement all of its registers each time an SSTRB occurs, but only while GO is high.

    During the sequences of the first part of the Write operation (including the Check Header function), each of these registers was decremented by the following amounts.

Sequence	Amounts
1	2
7	16

    This brings the amounts in each register to:
    a. OUT1 register contains 555.
    b. OUT2 register contains 557.
    c. OUT3 register contains 23.

235



Controller PCB Circuit Descriptions

236

CRC Operation

11. After the 22 sync bytes (which were zero's), and the "one" sync byte (which was a one), have been read out of RAM the OUT3 register will be decremented to 0, and the OUT1 register to 532.

    The OUT3 register reaching 0 causes a high from pin 15 of U34, removing the high Reset signal to pin 4 of U35 the CRC Generator/Checker. This chip is now enabled to process write data bits to produce 2 unique CRC bytes for the 532 bytes in the data field.

12. Meanwhile the Read Data MUX U6 is enabled by the high WTGT signal to select the serial output of the Serial/Deserial Shift Register U14 at pin 19.

    This serial write data is sent up to pin 11 of the CRC Generator/Checker U35. When 532 bytes have been counted, the OUT1 output from the Programmable Counter Timer U33 goes high causing the FRZN/WRTCRC signal to pin 5 of U28.

NOTE: Although the 3 U28 gates inputting to the CRC circuit are shown to be exclusive Ors, they are being used as inverters.

13. The resultant low on pin 10 of U6 causes the CRC Generator/Checker U35 to change modes to freeze the CRC generator, and clock out the 2 CRC bytes as NRZ Write data.

14. The OUT1 signal also goes to pin 19 of the BYTC State Machine to produce ENDCOUNT causing the State Machine to produce SECTDN and advance to sequence 2.

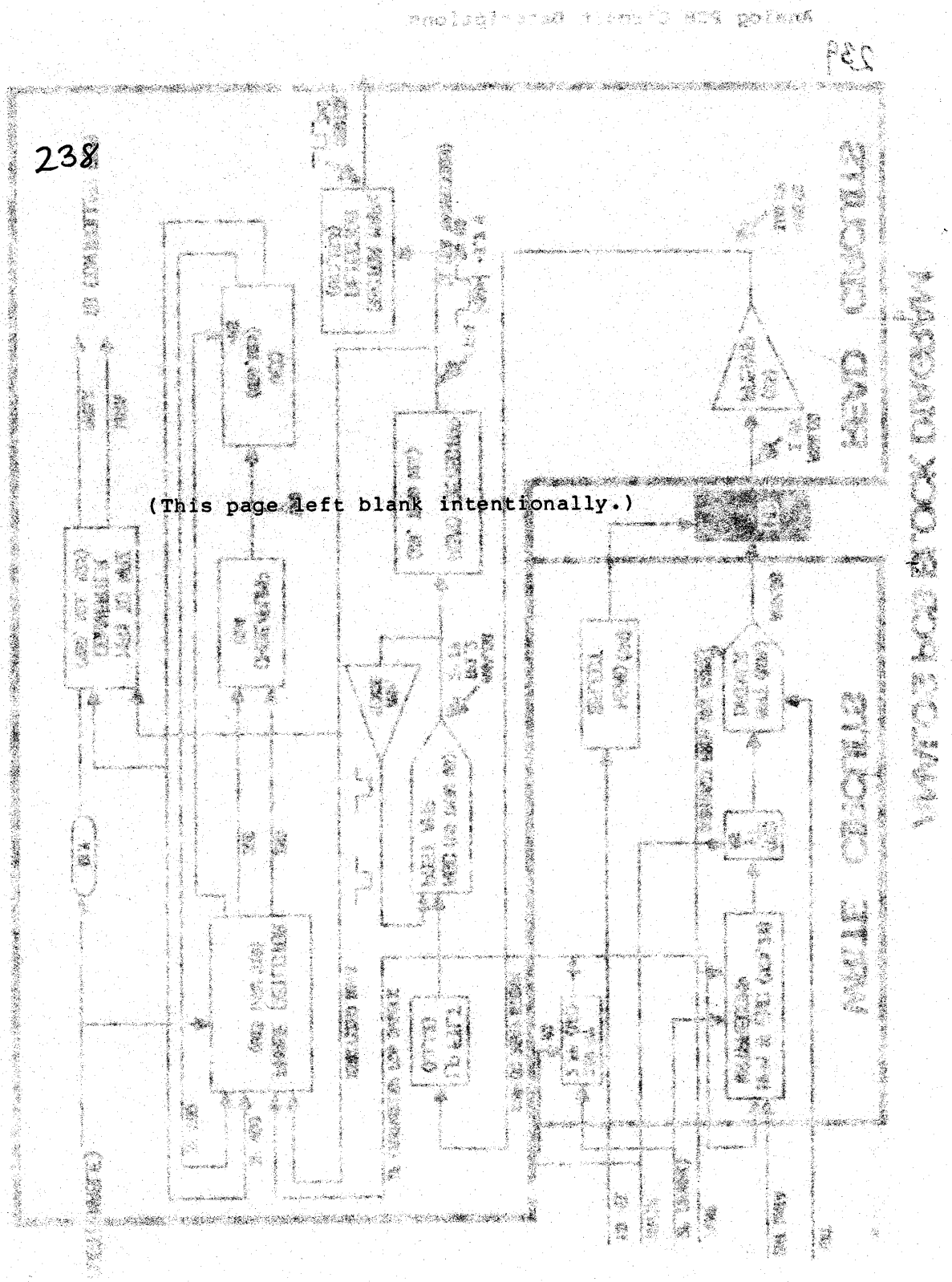
15. SECTDN goes to pin 31 of the Z8 to tell it that the processing of the sector is finished, and sequence 2 simply holds the Read/Write Control circuitry until the Z8 can process the SECTDN signal and drop the START signal. Otherwise spurious data could be stored in RAM.

237

3.3.1 Write/Verify Operation General Explanation

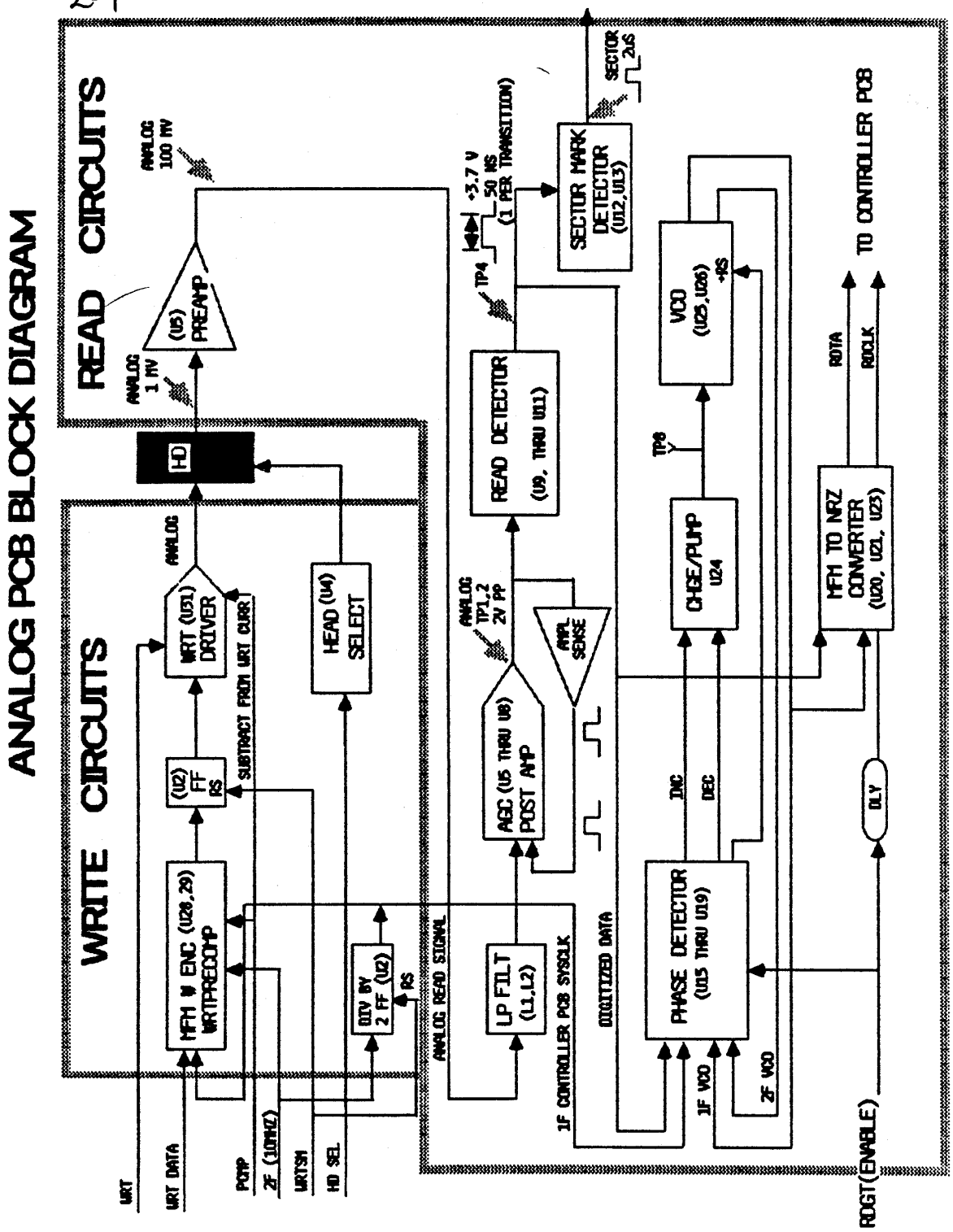
The Write/Verify operation is a combined Write and Read. The data comes from the Host in the same way as it does for a Write.

When the Write opereation is complete the Z8 goes through a verification of the data written, much like a Read operation.



(This page left blank intentionally.)

239



240

ANALOG PCB

1. General Explanation of the Analog PCB Write Circuits

For the following discussion refer to the Overall Block Diagram of the Analog PCB on the opposite page.

The Analog PCB serves as the interface between the Controller PCB and the Seagate HDA.

It consists of two main parts **Write** circuitry and **Read** circuitry.

The one Head Selector is shared for both Read and Write.

The Write circuitry consists of the following circuits:

a. Non-Return to Zero () to Modified Frequency Modulation (MFM) Write Encoder and Data Precompensater.

b. Write Driver.

c. Head Select Matrix.

1.1 NRZ to MFM Write Encoder and Data Precompensator Circuits

Prior to any operation (Write or Read), the Z8 on the Controller PCB sends the head select matrix inputs to cause it to select one of the four heads (disk surfaces).

During a Write operation, serial NRZ WRTDATA (Non Return to Zero simply means the data signal will be high when a one and low when a zero with no clocks) is converted to MFM data by the MFM Encoder/Precompensator. (For information on Modified Frequency Modulation refer to the MFM explanation at the end of this discussion.)

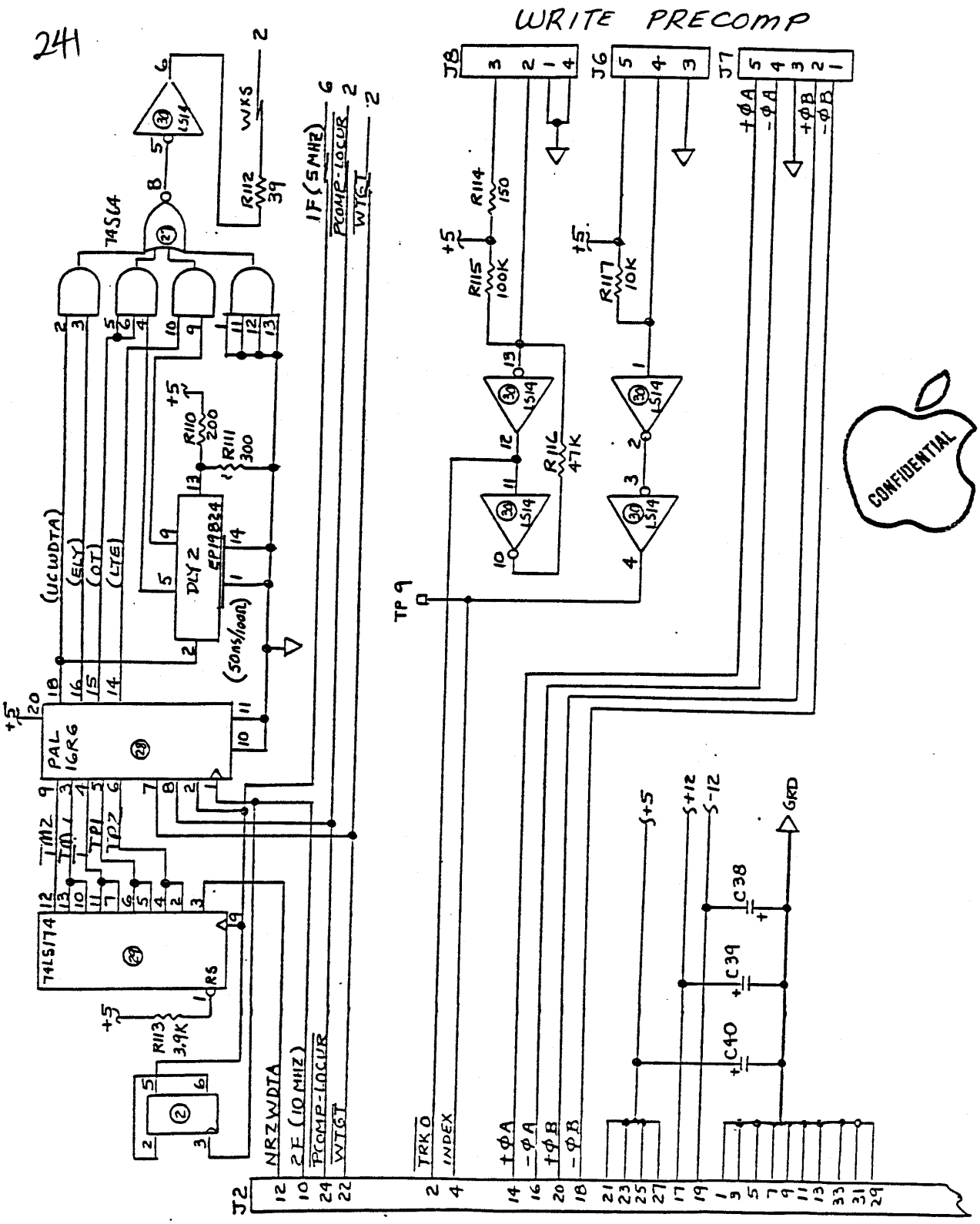
The Write Driver converts the MFM write pulses to alternating head current on the selected head.

If the data is being written on an inside track (tracks 128 to 152), the MFM write transitions will be peak shift precompensated, and the Write current will be reduced from 24 ma to 19 ma.

The Write Driver and Head Select Matrix are covered in more detail later.

For information on Write Precompensation refer to the Write Precompensation explanation at the end of this discussion.

241



242

2. Component Explanation of the Analog PCB Write Circuits

Refer to the diagram on the opposite page for the following discussion.

Some minor component value changes exist from this drawing to the production version. But, the illustration should be adequate for training purposes. The component population of the Write circuitry is quite small. However, the PAL (programmed array logic-U28) makes up for that by having a large amount of logic.

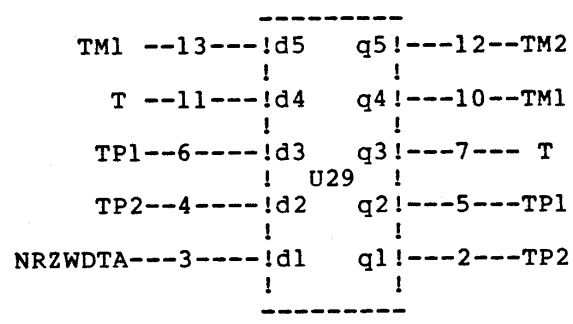
2.1 Clocking-

The 10MHz 2F signal is the master clock frequency from the crystal oscillator on the Controller PCB. It is divided to 1F (5MHz) by U2 (a simple divide-by-two stage). The 1F signal is the bit shift clock for the precompensation shift register. It is also used to keep the VCO synchronized to write data during the Write operation. The 2F signal provides the clock enable signals necessary to perform precompensation on the data as it is output from the PAL.

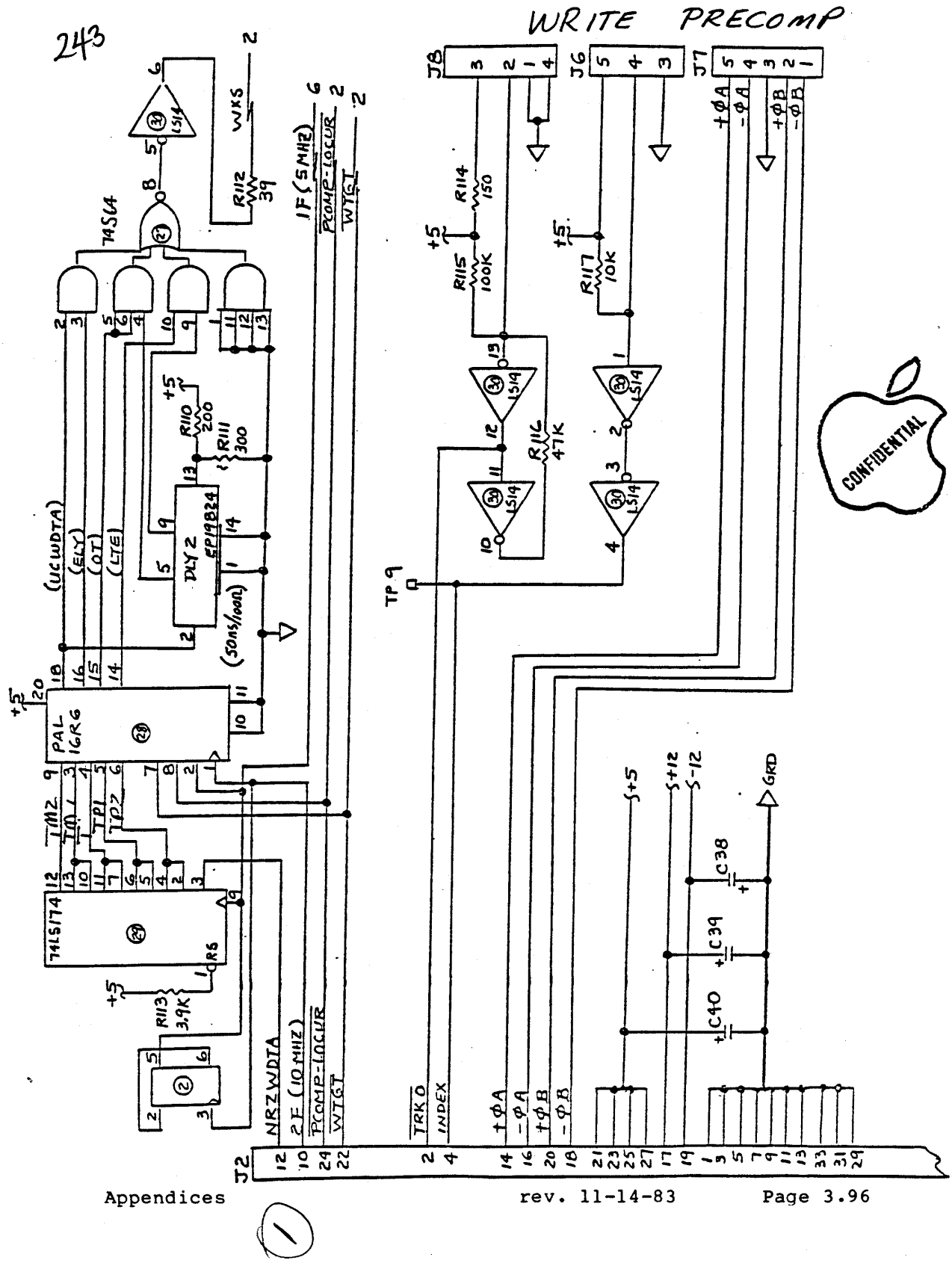
2.2 Precomp Shift Register U29

U29 has been externally wired into a shift register. Its output shift pattern provides five of the inputs to the PAL.

For clarity, the following is a sketch of U29 and its signals with the "in's" on the left and the "out's" on the right.



It makes things easier if you visualize the data stream in reference to the bit named "T". The TMx signals are what T was one or two clocks ago, and the TPx signals are what T will be in one or two clocks. The initial NRZ data is clocked into q1, the next clock places q1 at q2, and so forth. The whole intent is to give the PAL a window on the data stream to make the encoding and the early/late/ontime precompensation decisions discussed earlier.



244

2.3 PAL Chip U28

U28 is a custom array of logic designed specifically for this circuit. It consists of a huge array of AND/OR gates feeding each D input of its output register.

The PAL chip is involved in two functions:

1. To convert the NRZ digital stream into a stream of pulses conforming to the rules of MFM as stated earlier.

2. To provide the gating signals to advance, retard or leave alone the stream of write data pulses called "UCWDTA" (uncompensated).

As a bit is being processed through the PAL chip, its internal array of AND/OR gates receives inputs from the Precomp Shift Register U29, telling the PAL which bit combinations surround that bit.

This enables the PAL to determine when to generate a transition to convert it to MFM data and what type of precompensation (if any) to provide.

2.4 Precompensation Gating U27

UCWDTA goes to the input of a gate enabled with ELY (Early) and to a delay line DLY2, which delays the pulses approximately 15 ns per stage. The outputs of the delay line are partial enables to the OT (Ontime) and LTE (Late) gates respectively.

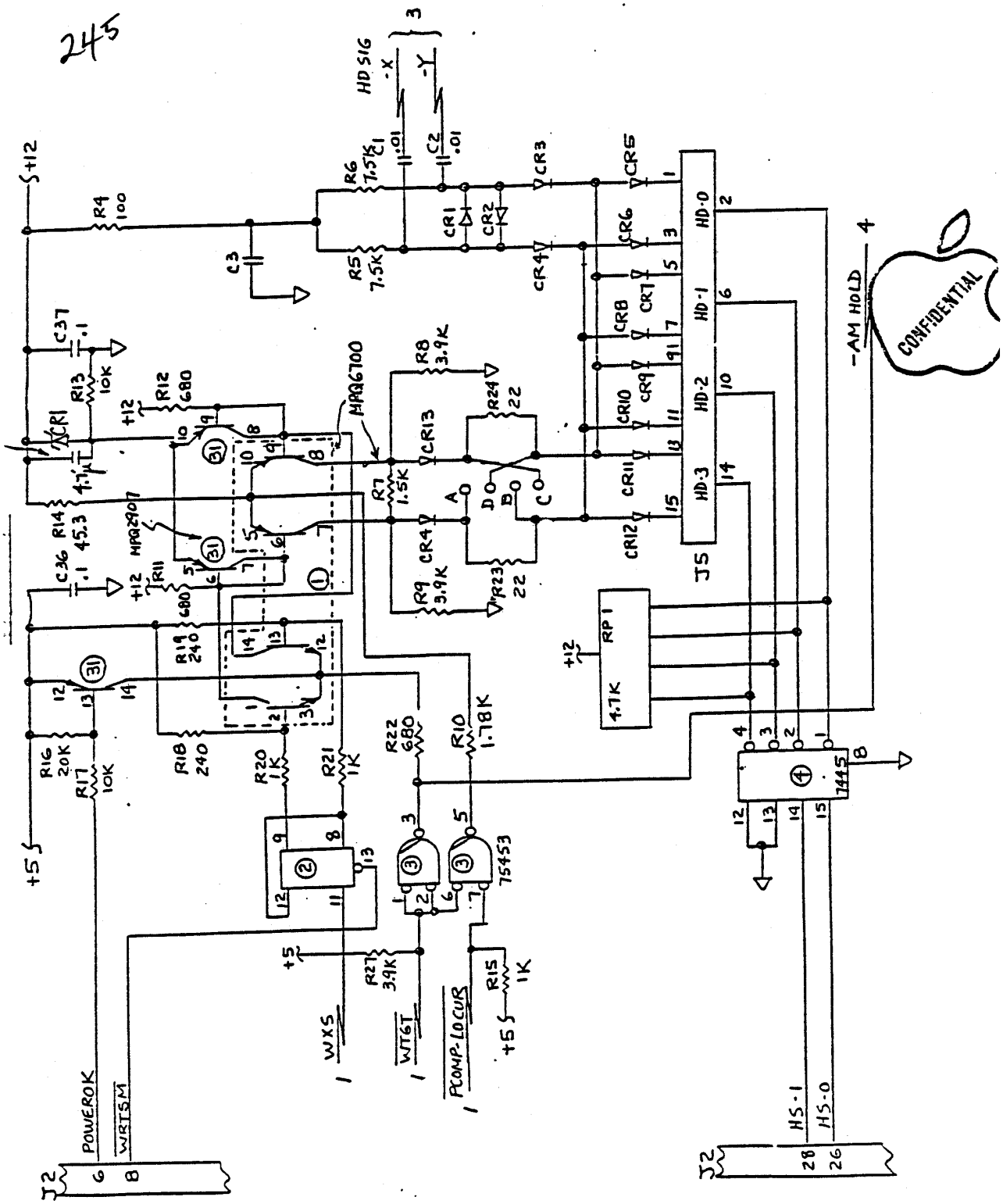
If there is no precompensation, (such as when writing to the outer tracks), the signal OT gates the first delayed data line to the WXS line.

If precomp'ing determines that the data transition needs to be advanced, then the ELY signal will gate UCWDTA to the WXS line; and if the signal needs to be retarded, the signal LTE will gate the second delay of data to the WXS line.

DLY2 provides the amount of delays needed for each type of precompensation. Only one signal (UCWDTA, ELY, OT, or LTE) can be true at any one time.

Analog PCB Circuit Descriptions

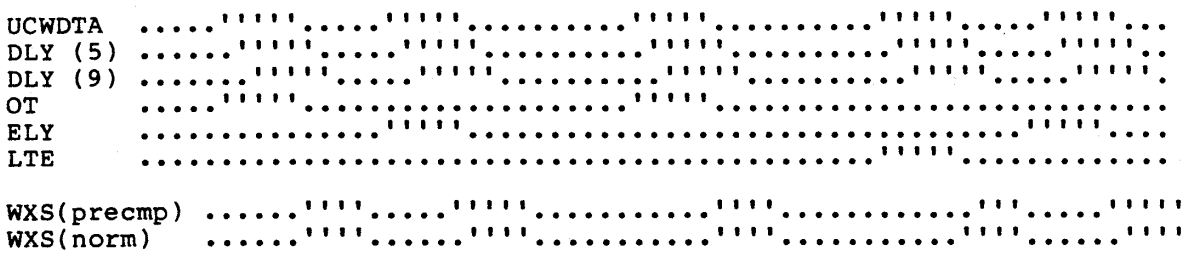
245



2

246

2.4 Precompensation Gating U27 (continued)
The following timing diagram might help make this more clear.



NOTE: this diagram is for training purposes and does not necessarily conform to the rules of precomp....OK?

The leading edge of the WXS signal is the working edge that toggles the WXS FF. The slight shift of timing between the precomp and unprecompensated signal is enough to increase Read data recovery by several magnitudes.

2.5 Miscellaneous Circuits U30
These drivers shape the TRACK 0 sensor signal and the INDEX pulse from the HDA. The Track 0 circuit forms a noise latching function too (the source signal is pretty ragged).

2.6 Write Current Gating U1, U2, U3, U31
Each WXS pulse from the precomp gating circuit toggles the WXS FF, U2. The result is a divide by two function.

The output levels of the WXS FF are level shifted by the first set of transistors in U1.

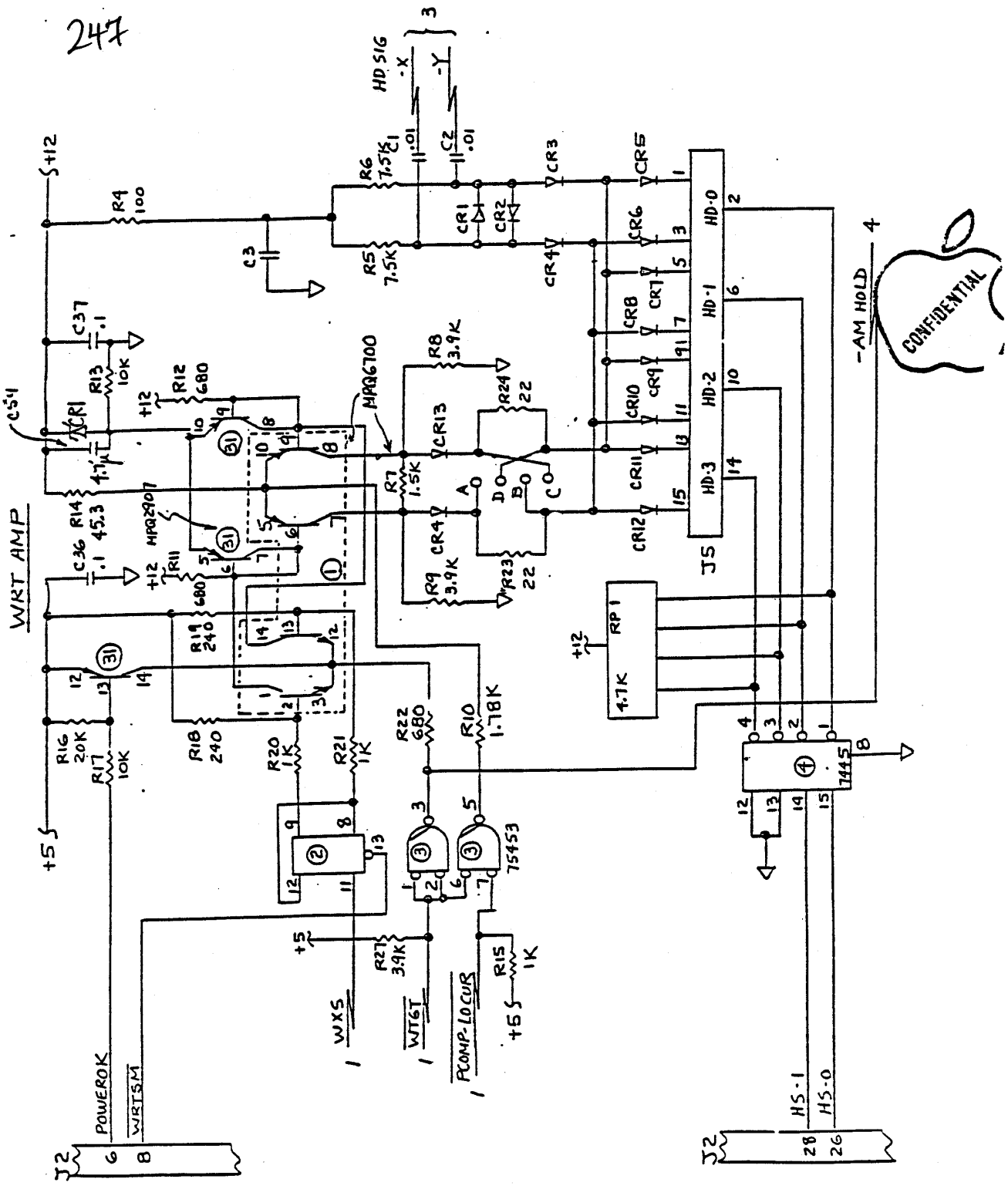
The remaining transistors comprise a constant current-switching network. These transistors are turned on and off depending on the state of the WXS FF.

During Formatting, at the end of every sector the Z8 on the Controller PCB directs the Analog PCB to write a sector mark. It does this by putting a constant low on pin 13 of the WXS FF to clamp it in the reset state for a 10-byte period.

Each time this happens, the heads output a DC level (no transitions) for a field (10 bytes long), creating a sector mark.

Analog PCB Circuit Descriptions

247



248

2.6 Write Current Gating (continued)
A string of zeros would not create the DC level because they would be converted by the MFM encoder to a 5 MHz signal at the heads.

The Z8 causes WTGT to go low and turns on the current switching circuit when it wishes to perform a Write operation.

WTGT (Write Gate), when inactive (high), kills both sides of the level shifter and, therefore, turns off both sides of the current source, effectively killing all write current.

The signal $\overline{\text{PRECOMP-LOCUR}}$ is produced by the Z8 to cause a reduction of write current to the heads when writing to the inner tracks (128 to 152).

Normal write current is about 24 ma. But, in the inner tracks it is reduced to slightly less than 20 ma because the bits are packed closer together.

The POWEROK signal is generated by the power supply. If it goes low that means that the power supply has detected a power problem.

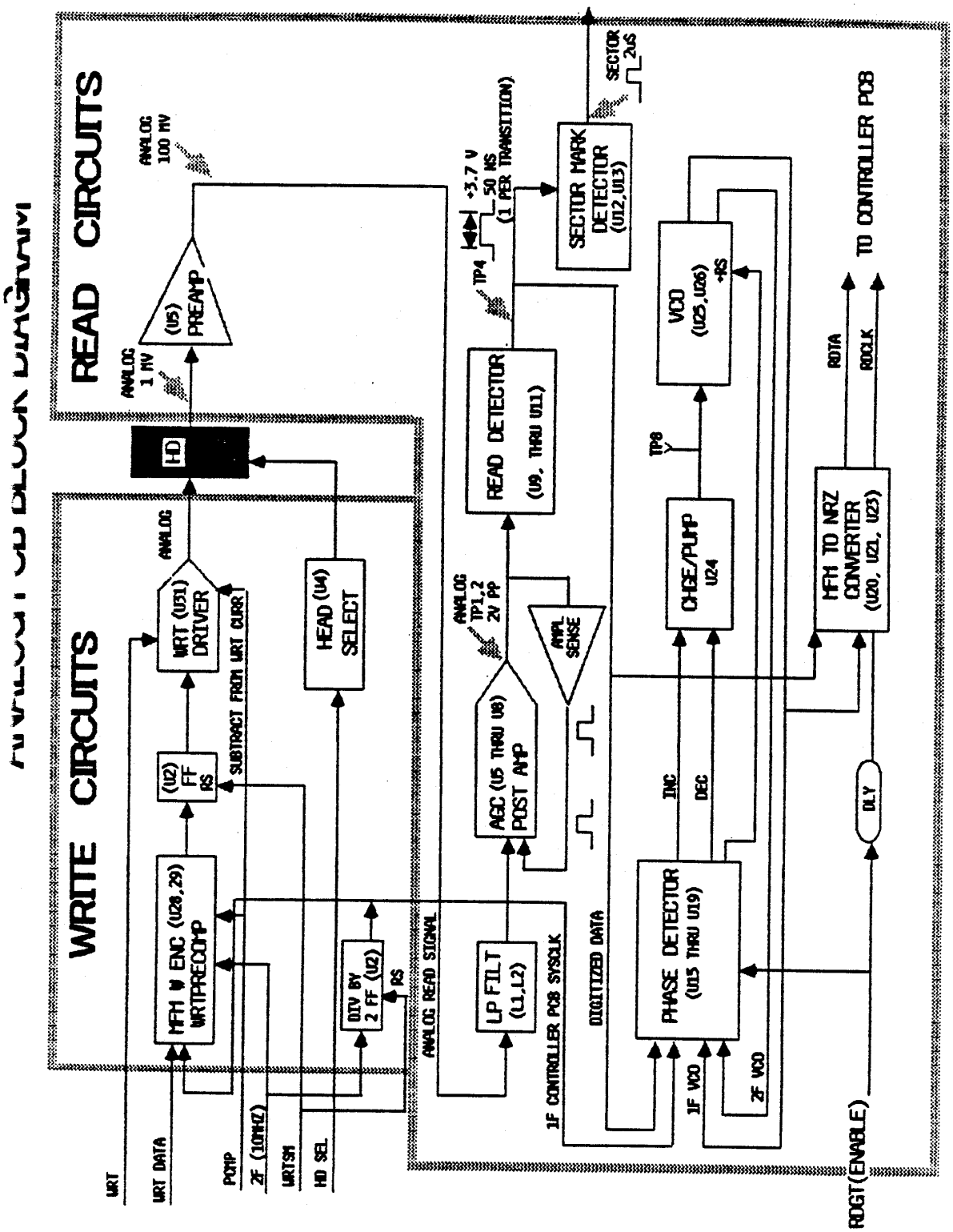
If the POWEROK signal does go low, this biases pin 13 of a transistor in U31 to turn it on inhibiting any write from taking place.

2.7 Head Selection U4
Head selection is done by U4 (7445, one of ten decoder). The signals HS0 and HS1 are code outputs from the Z8 to select the proper head (00 selects head 0, 01 selects head 1, 10 selects head 2, and 11 selects head 3).

Only one head can be selected at a time. The heads are center tapped, and the center is connected to the 7445 to create a current path for Reading or writing.

249

Analog PCB Circuit Descriptions



250

3. GENERAL EXPLANATION OF THE ANALOG READ CIRCUITRY

The following circuits are included in the Read circuitry:

1. Preamplifier and Automatic Gain Control (AGC).

2. Read Detector and Sector Mark Detector.

4. Phase Locked Loop (PLL) consisting of:

   a. Phase Detector.
   b. VCO Charge Pump.
   c. Voltage Controlled Oscillator.

5. MFM to NRZ Read Decoder.

During a Read operation, the preamplifier amplifies the low level (.6 to 2.0 mV) signal from the selected head to a 200 mV.

The analog read signal is then filtered and amplified to a fixed 1.0 V output signal by the AGC circuit.

This analog read signal is fed to the differentiator and gated detector to derive the MFM read pulses.

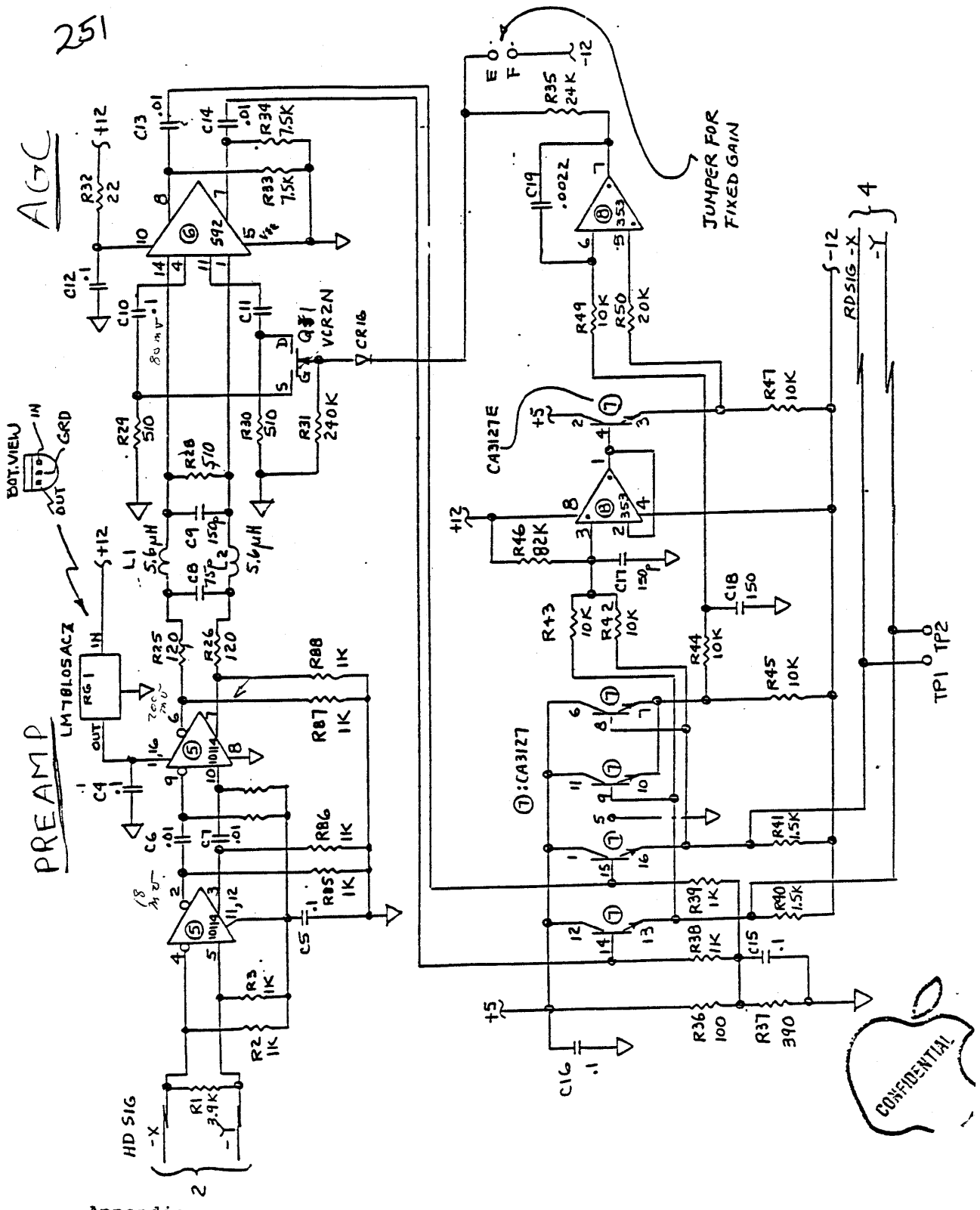
The PLL (Phase Lock Loop) synchronizes its VCO to the MFM read pulses and provides the clock for the MFM to NRZ data converter.

The serial NRZ data and clock are sent back to the Controller for deserialization.

During format operation, when the initial sector identifier information is written on the disk, sector boundaries are written as DC-erased 10 byte fields.

During a Read operation, the sector boundary detector looks for the absence of Read signals for about 10 ms (to detect the DC-erased 10 byte field). When it detects one, it sends a sector pulse to the Z8 on the Controller PCB.

Analog PCB Circuit Descriptions



252

4. Component Explanation of the Analog Read Circuitry

4.1 Analog Preamplifier, and AGC (Automatic Gain Control) Circuits

The Read circuitry is always on. However, because of the amplitude of the write signal through the heads, it goes into saturation during the Write operation and does not reliably recover the data.

During Reading (not writing), the selected head develops a signal on the -x -y lines.

This signal is about 1 mv and cannot usually be seen with conventional techniques.

Probing dampens the signal and induces so much noise that the signal is obliterated. So don't expect to see data directly from the heads.

The read signals from the head are brought to the first stage preamplifier, U5.

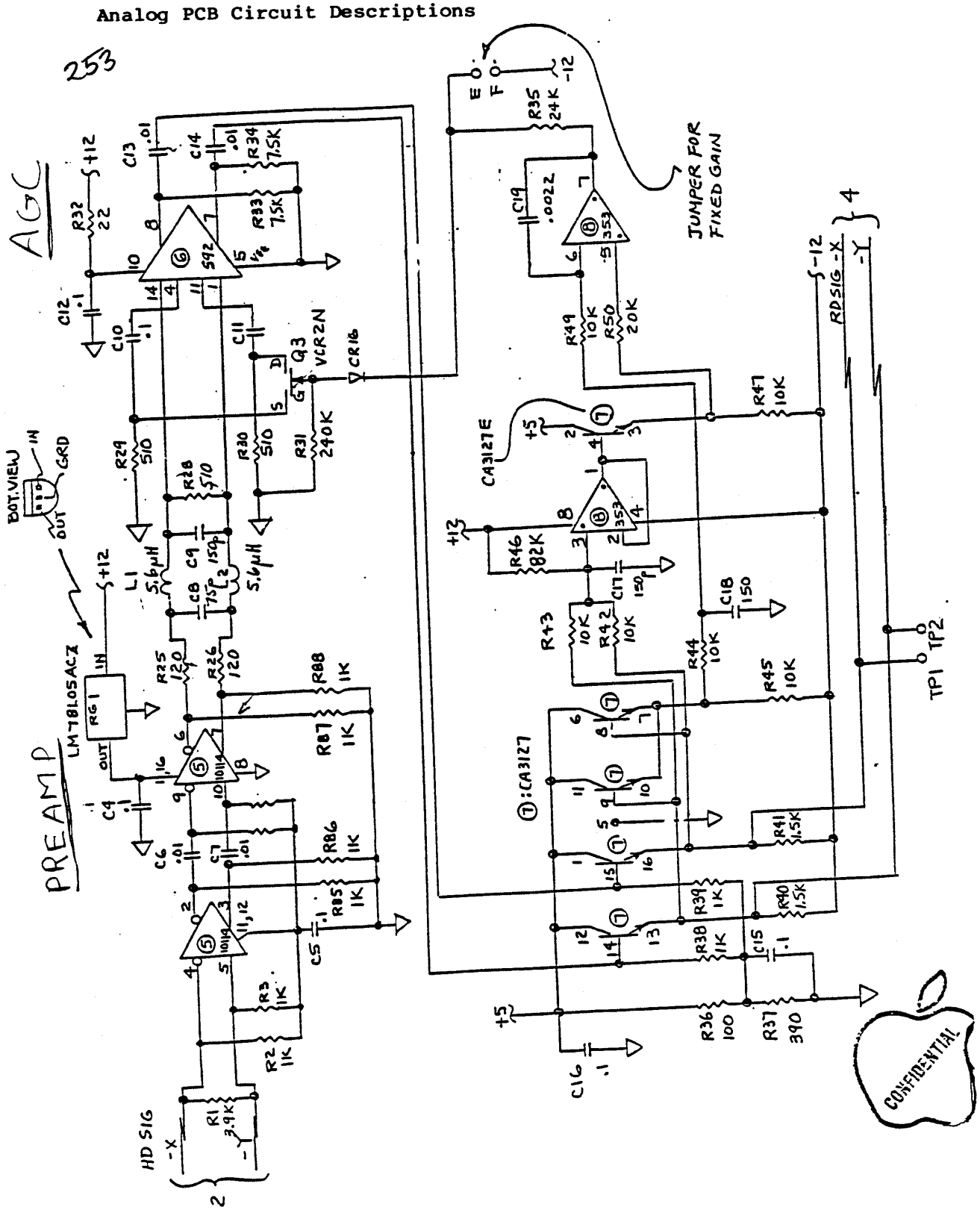
The devices used here are ECL (Emitter Coupled Logic), a high speed logic family. Each stage of the preamplifier has a gain of 12.

This means that the approximate 1 mV head signal leaves the first stage at about 18 mV. The signal is amplified again, and leaves this stage at about 200 mV.

That's nice because the filter network (L1, L2, C8, C9) attenuates the signal about 60%.

The analog read signal then enters the AGC (Automatic Gain Control) circuit. FET transistor Q3 controls the reference voltage to to the amplifier U6.

The input to the gate of Q3 comes from the final output of the AGC circuit.



Analog PCB Circuit Descriptions

254

If the output goes up, the bias on Q3's gate will cause it to increase the reference voltage on pins 11, and 4 of U6. This will cause the output of U6 to decrease.

If the output from the AGC circuit goes down, the bias on Q3's gate causes it to decrease the reference voltage on pins 11, and 4 of U6. This causes the output of U6 to increase.

The result of this process is that the AGC circuit is continually striving to regulate the input from the heads into a constant output level.

However, it takes a certain amount of time for this regulation to take place. That is why if you monitor TP1 or TP2 on the Analog PCB while Reading, you will see an increase in signal amplitude at the beginning of each sector.

During a sector mark, there is little amplitude at pin 7 of U8 (the final output of the AGC circuit), so Q3 is conducting, putting the reference voltage for U6 at a very low level.

The low reference voltages on pins 11, and 14 of U6 cause it to amplify at maximum gain.

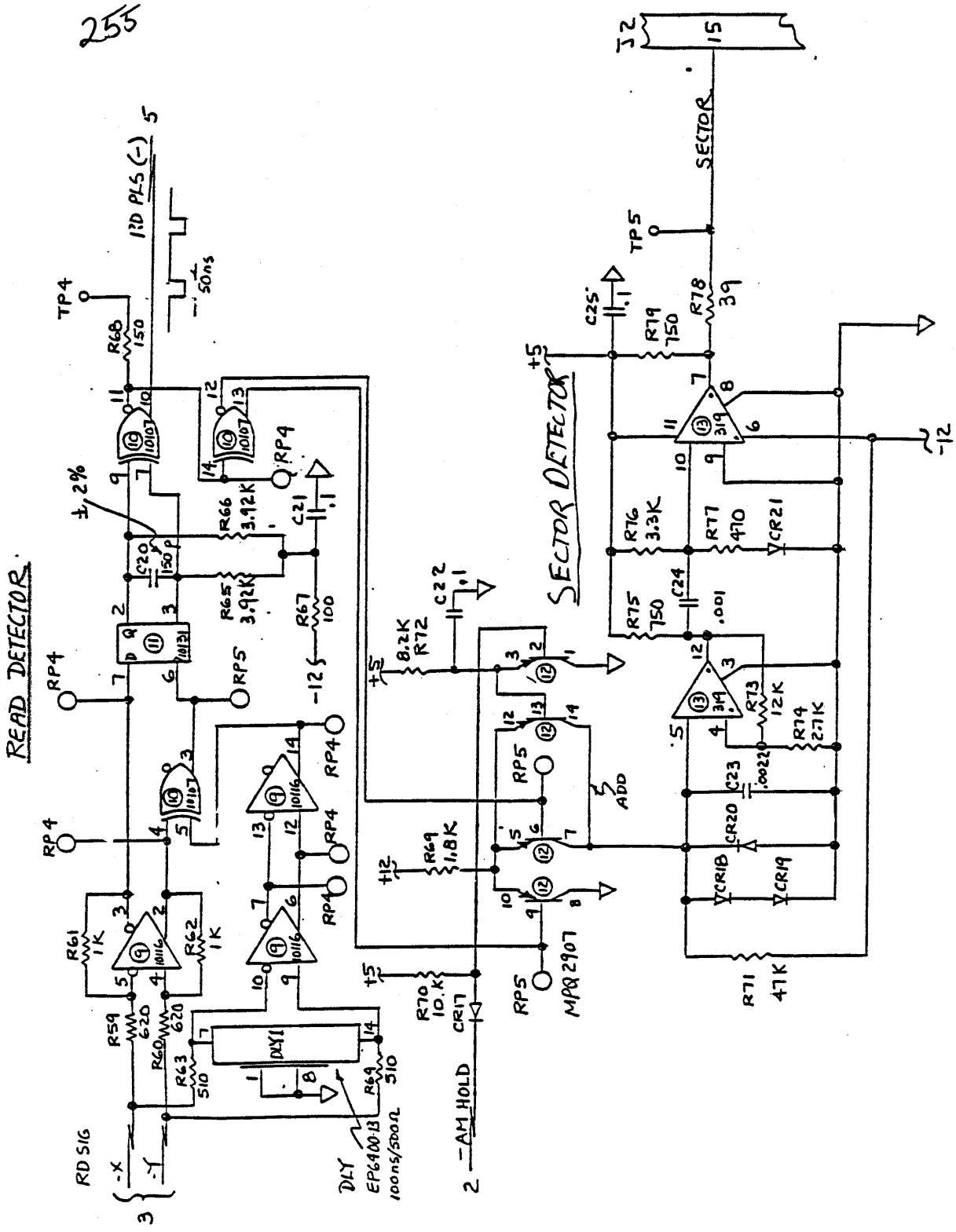
When the selected head starts receiving the next sector, U6 is still at maximum gain, so the sector read signal leaves U6 at a comparatively high amplitude.

Since the gain for the rest of the amplifiers in the AGC circuit is constant, this high output from U6 is amplified even further.

When pin 7 of U8 finally goes up, it causes Q3 to conduct less. This increases the reference voltage to U6, which decreases the output of U6, bringing the output of the AGC circuit back to its normal level.

This whole process occurs after every sector mark, and can be seen at TP1 and TP2 as a flare in amplitude at the beginning of each sector.

Analog PCB Circuit Descriptions



256

4.2 Read Detector, and Sector Detector Circuits

The Read Detector detects transitions in the analog MFM read signal and produces a 50 ns pulse for each one that occurs. The result is called digitized data.

U9, U10, and DLY1 set U11 every time a transition occurs.

C20, R65, and R66 make up an RC network that allows the set output of U11 to keep pin 9 of an Exclusive Or in U10 above its threshold and pin 7 below its threshold until the RC network's time constant is satisfied (TC = 50 ns).

This process produces a 50 ns digitized data pulse (called RDPLS) out of pins 10 and 11 of this Exclusive Or for every transition in the received analog signal.

(Refer to the timing diagram on page * for an example of the timing relationship between the analog MFM read signal, digitized data, MFM data, and NRZ data.)

The RDPLS signal is actually digitized MFM data, so it needs to be converted to NRZ data and to be sent to the Controller PCB.

It is also used to detect Sector Marks.

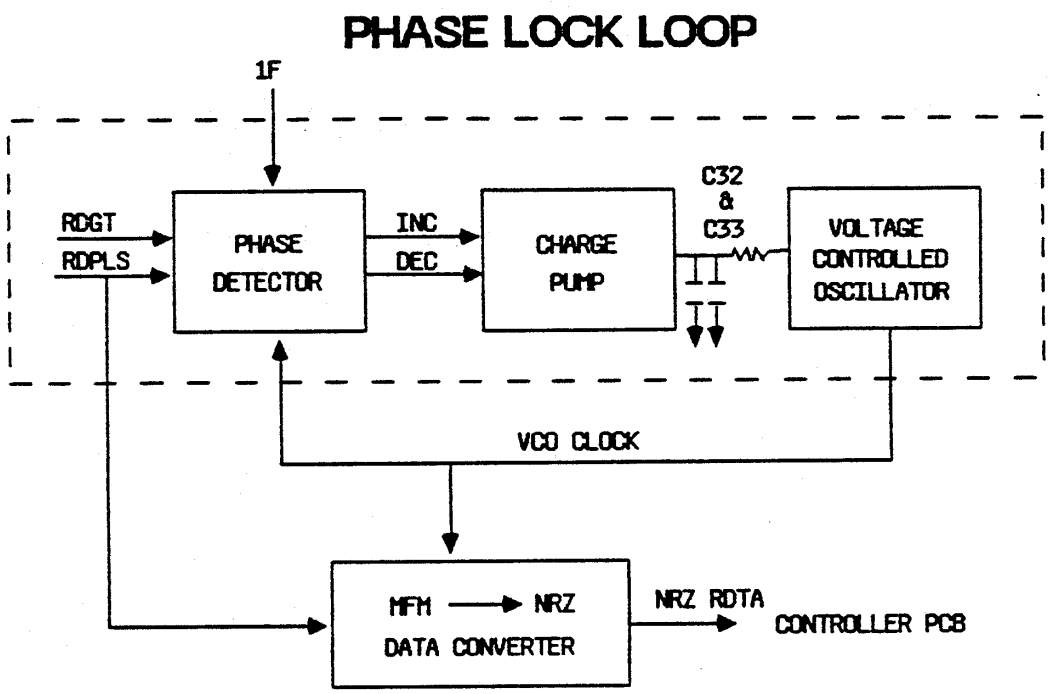
The output of pins 12 and 13 are sent down to the Sector Mark Detector. The RDPLS is gated down through transistors on U12, which pump a charge to C23 at the input of U13.

If there are no read pulses for approximately 10 ms (this can only happen during a sector mark), R71 slowly discharges C23 and causes U13 pin 12 to go high.

This signal is integrated to pin 7 of U13, which produces a 2 us pulse, called SECTOR, to the Z8 on the Controller PCB to notify it that a sector mark has occurred.

When read pulses reappear (as in the sync field at the beginning of a sector), C23 quickly recharges and U13 pin 12 goes low again.

257



258

4.3 PLL (Phase Locked Loop)

All of the circuitry described prior to this point is constantly enabled.

However, from this point on, signals will be allowed to pass only when enabled by the RDGT signal from the Z8 on the Controller.

The block diagram on the opposite page shows the PLL (Phase Locked Loop) on the Analog PCB.

The PLL provides a clock frequency in sync with the incoming MFM digitized data signal to enable the MFM to NRZ Data Converter to perform its function and to forward the serial NRZ data stream to the Controller PCB.

(for information on Modified Frequency Modulation refer to the MFM explanation at the end of this discussion).

The PLL has two modes:

1. The PLL is put in the **Idle** mode when the RDGT signal from the Z8 on the Controller PCB is inactive (any time the Z8 is not actually performing a Read operation; i.e., Write or Idle).

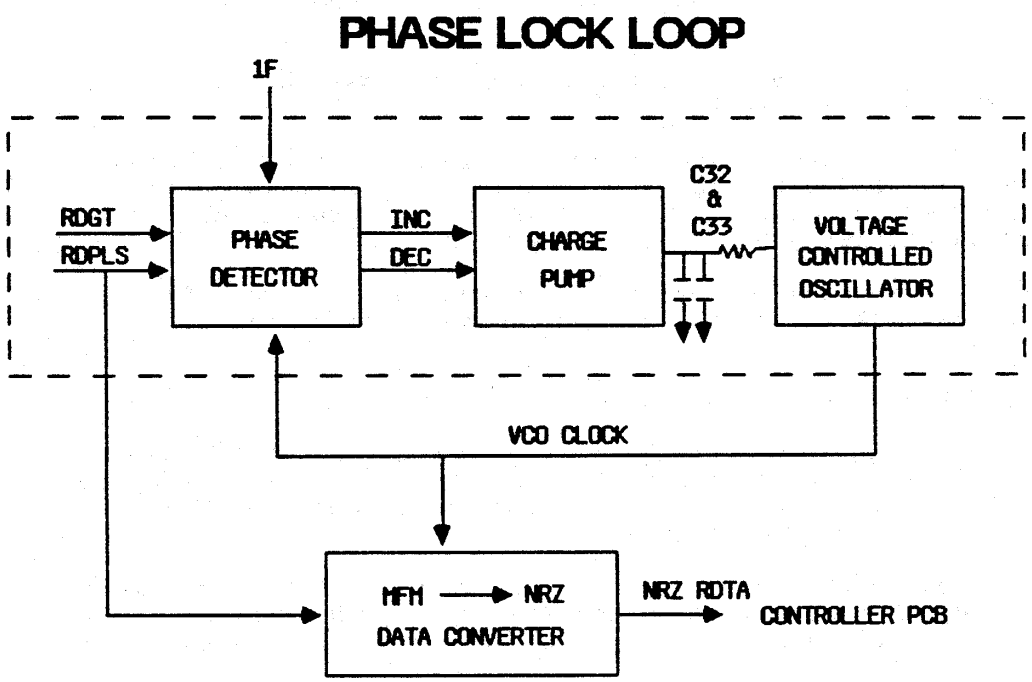
   During the Idle mode the PLL syncs the VCO to the master clock frequency from the Controller PCB. This mode has no real purpose and so will not be discussed any further.

2. The PLL is put in the **Read** mode when the Z8 makes RDGT active during a Read operation. During the Read mode, the PLL syncs the VCO to the incoming MFM digitized data signal. This is performed as follows.

The Phase Detector compares the phase of the VCO clock frequency with the digitized data pulses (RDPLS) from the read analog signal.

If the Phase Detector receives a RDPLS before it receives a VCO pulse, then the VCO's phase is lagging behind the incoming data, and it needs to speed up. So, the Phase Detector generates the INC (increase) signal to the Charge Pump circuit.

259



260

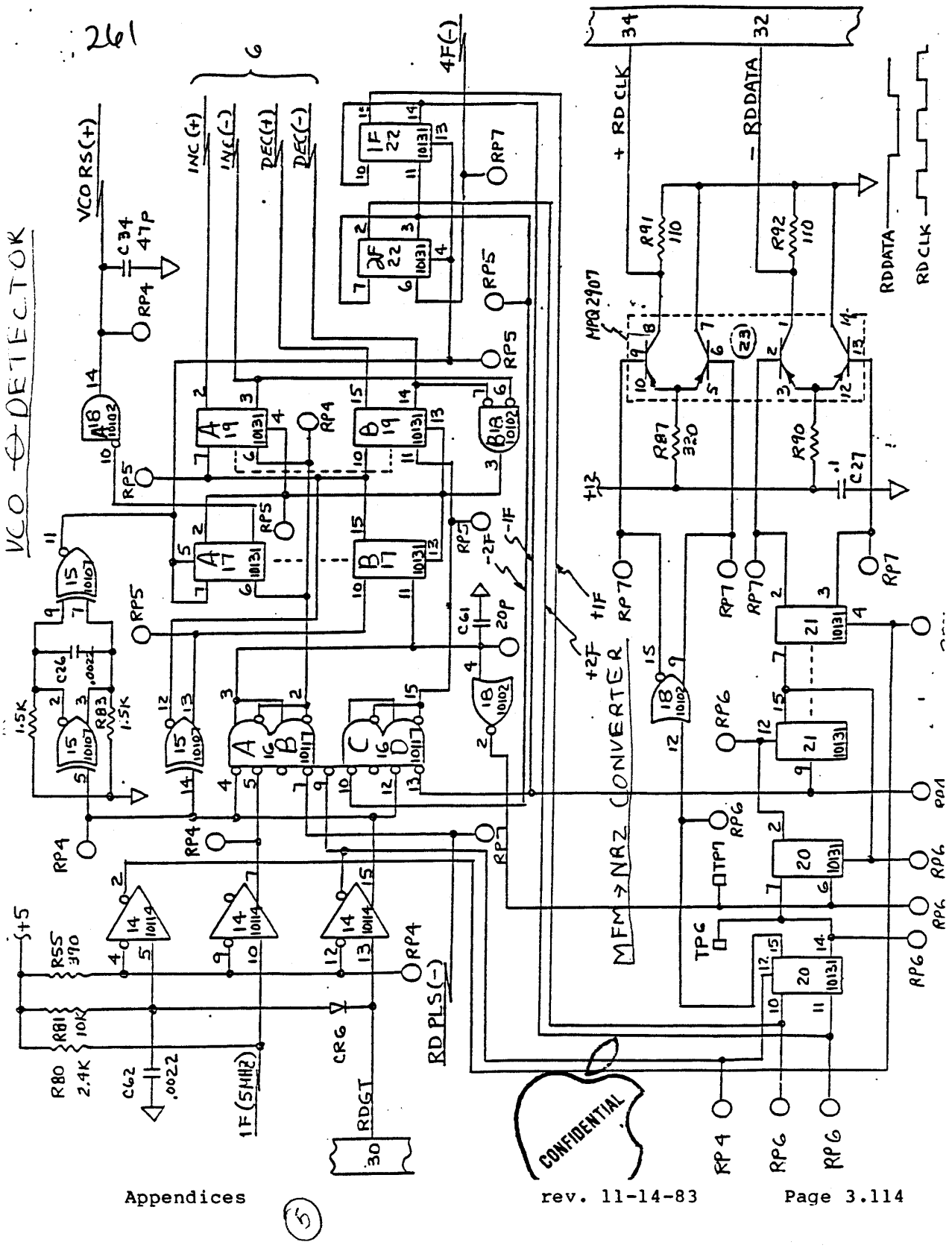
4.3 PLL (Phase Lock Loop) (continued)
The Charge Pump circuit is then enabled to put a positive charge on C32 and C33.

C32 and C33 store the VCO control voltage level that controls the frequency produced by the VCO. The more positive input to the VCO now causes it to increase its output frequency to catch up to the frequency of the incoming data.
If the Phase Detector receives a VCO pulse first (before a RDPLS), then the VCO's phase is leading that of the incoming data, and it needs to slow down.

So, the Phase Detector generates the DEC (decrease) signal to the Charge Pump circuit. The Charge Pump circuit is then enabled to put a negative charge on C32 and C33.

The more negative input to the VCO now causes it to decrease its output frequency, to try to slow down to the frequency of the incoming data.

The different circuits in the PLL are explained in greater detail on the following pages.



Analog PCB Circuit Descriptions

262

4.3.1 VCO Phase Detector and MFM to NRZ Data Converter Circuits

There are two main circuits shown on the opposite page, the VCO Phase Detector (U15, 16, 17, 18, 19, and 22) and the NRZ to MFM converter (U20 and 21).

The VCO Phase Comparator compares the phase of the clocks from the VCO (Voltage Controlled Oscillator) with those of the incoming analog read signal.

The 20 MHz (4F) output of the VCO comes in to pin 6 of FF "A" on U22. FF "A" of U22 divides 4F to 2F (10 MHz), FF "B" divides 2F into 1F (5 MHz).

Pins 2, 14, and 15 from U22 go to the MFM to NRZ data converter. Pin 3 goes to Selector Gate "C" on U16.

The Selector Gates are the inputs to the Phase Dectector and are actually four separate Nand gates.

VCO Phase Detector Idle Mode

During Idle mode, RDGT from the Z8 through pin 15 of U14 is low. This enables Selector Gate "A" to pass the 1F master clock frequency and Selector Gate "D" to pass the 1F VCO frequency.

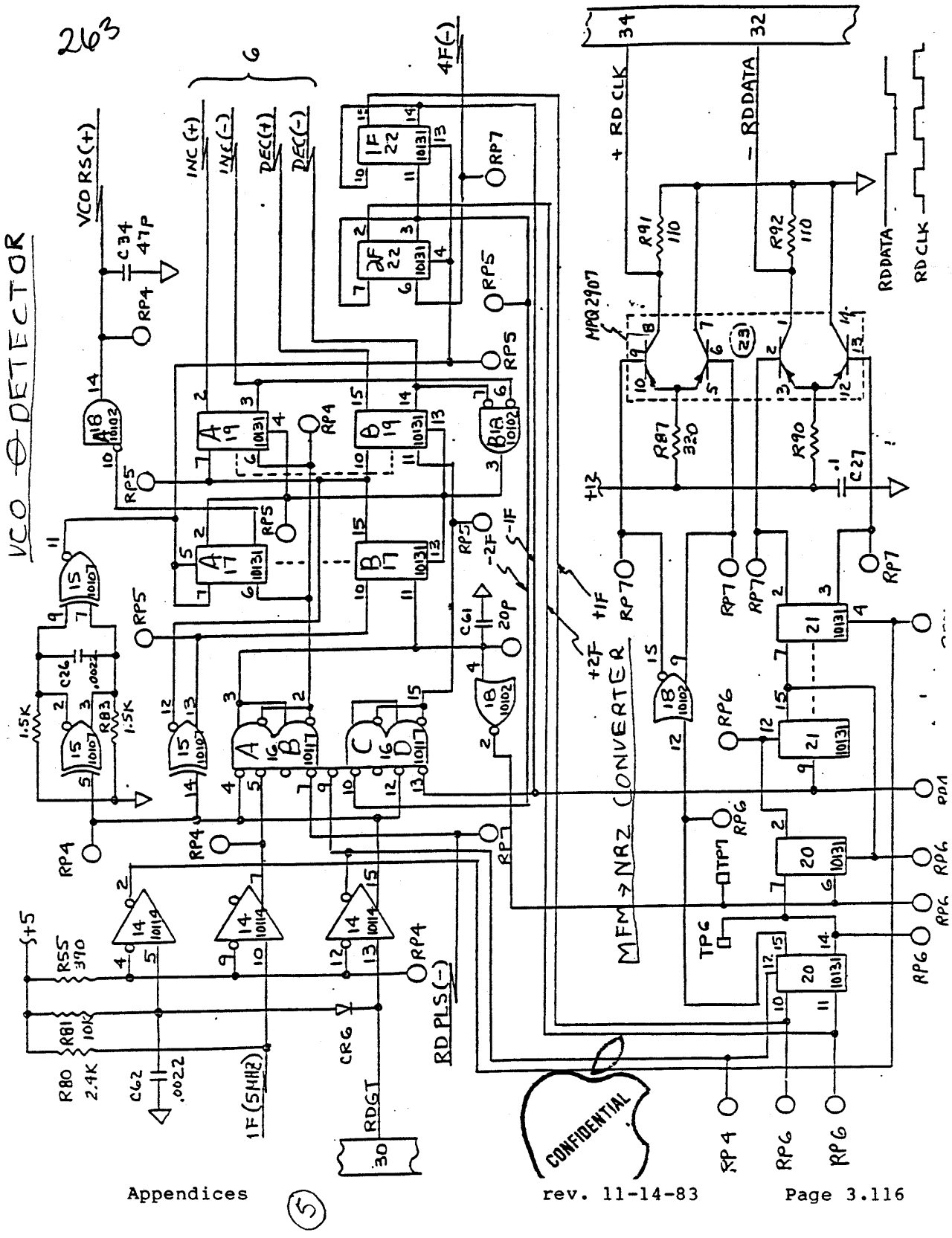
This keeps the VCO in sync with the master clock on the Controller PCB (there is no real reason for this, because the output NRZRDTA is not enabled to go to the Controller PCB.).

VCO Phase Detector Read Mode

When the Z8 is performing a Read operation, RDGT is high, enabling Selector Gate "B" to pass RDPLS (digitized MFM read data) and Selector Gate "C" to pass the 2F VCO frequency. (Pin 9 is actually common to both Selector Gates "B" and "C".)

When the Z8 first initiates the Read mode by setting the RDGT signal high, RDGT enters pin 5 of U15 to generate a reset signal to flipflops "A" and "B" in U19 and flipflop "B" in U17.

The reset signal from pin 11 of U15 also presets flipflop "A" in U17 causing the VCRS (VCO Reset) signal out of gate "A" of U18. (Incidently it resets the VCO as its name implies.)



Analog PCB Circuit Descriptions

264

If the Phase Detector receives a RDPLS before it receives a VCO pulse, then that means the VCO's phase is lagging behind the incoming data so the VCO needs to speed up.

In this case the low out of pin 2 of Selector Gate "B" caused by the RDPLS sets FF "A" in U19 through pin 6, causing it to generate the INC (increase) signal to the Charge Pump circuit.

This signal enables the Charge Pump to put a positive charge on C32, and C33 causing the VCO to speed up.

Later, when the 2F signal from the VCO causes Selector Gate "C" to output a low, it will set FF "B" in U19 through pin 11, gate "B" in U18 will then be fully enabled to reset the FFs U17, and 19.

If the Phase Detector receives a VCO pulse first before a RDPLS, then that means the VCO's phase is leading that of the incoming data so it needs to slow down.

In this case the low out of pin 15 of Selector Gate "C" caused by the VCO 2F signal sets FF "B" in U19 through pin 11 causing it to generate the DEC (decrease) signal to the Charge Pump circuit.

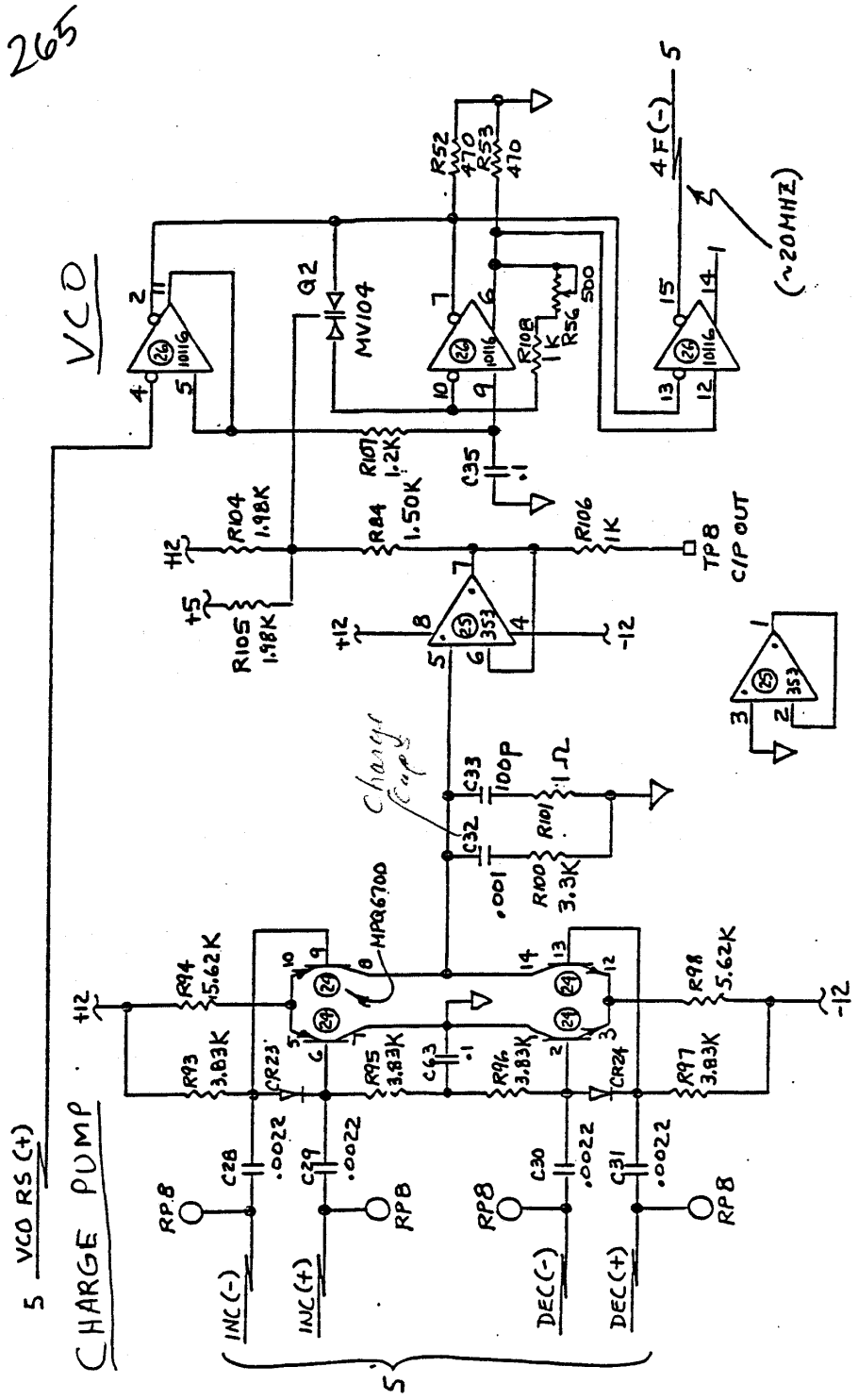
This signal enables the Charge Pump to put a negative charge on C32, and C33 causing the VCO to slow down.

Later when the RDPLS signal arrives, it will cause Selector Gate "B" to output a low, setting FF "A" in U19 through pin 6, gate "B" in U18 will then be fully enabled to reset the FFs U17, and 19.

The RDPLS's are gated down to U20, which takes the digitized MFM data stream and beats it against the function of 1F and 2F ((VCO) now in sync with the incoming data) to convert it to NRZ data.

The results, RD CLK and RD DATA (this data is in NRZ form) are sent to the Controller PCB.

Analog PCB Circuit Descriptions



2166

4.3.2 Charge Pump and VCO circuits

There are two circuits shown on the opposite page, the Charge Pump on the left and the VCO on the right.

If the Phase Detector receives a RDPLS before it receives a VCO pulse, the VCO's phase is lagging behind the incoming data, so the VCO needs to speed up.

In this case, a high INC + (increase) and a low INC - signal come from the Phase Detector. The low INC - turns its NPN transistor on to send a negative voltage to C32 and C33.

U25 amplifies the negative charge on C32 and C33 to Q2 in the VCO. Q2 is biased by the charge on C32 and C33 to control the frequency produced by the VCO.

If the Phase Detector receives a VCO pulse before it receives a RDPLS then the VCO's phase is leading the incoming data, so the VCO needs to slow down.

In this case, a high DEC + (decrease) and a low DEC - signal come from the Phase Detector.

The high DEC + turns its PNP transistor on to send a positive voltage to pin 5 of an amplifier in U25. Which puts the positive charge on to C32 and C33 for storage, etc.

Analog PCB Circuit Descriptions

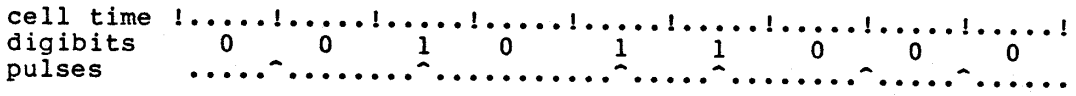
267

WHAT IS MFM (Modified Frequency Modulation) ?:

MFM is a scheme in which the ones and zeros of a digital bit stream are converted to a stream of transitions (phase reversals) according to the following rules:

1. Ones always cause a pulse at center cell time.
2. No more than two cells can occur without a transition.
3. Only one transition can occur per cell time.
4. Zeroes will cause a pulse at the start cell boundary, if not preceded by a one.

Look at a bit stream of 001011000.



Pulses occurred at the start cell times for the first two zeros. Then a pulse occurred at center cell time for the first one.

However, the pulse for the third zero had to be dropped because the following one would have caused another transition to occur less than one cell time away. This would have broken Rule three.

The second and third pulses occur at center cell. (All ones cause pulses at center cell, it's the zeros that get played with.)

The pulse for the fourth zero gets dropped because it was preceded by a one. The fifth and sixth zeros cause pulses at start cell boundaries because no ones interfere.

The diagram on the following page might help you to gain a better understanding of MFM data and its relation to the other methods of modulating data that are used in the Pro-File.

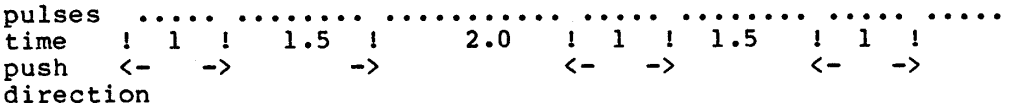
For the Pro-File, the data frequency is 5 MHz, sometimes referred to as 1F (2F would be 10 MHz, 3F would be 15 MHz, etc.).

268

WHAT IS PRECOMPENSATION ?

When one records things at high density on magnetic media, a phenomenon occurs, called, "don't get too close or I'll push you away." This refers to the polarity of the small magnetic fields, on the disk surface associated with each write pulse.

Example:



Once data is laid down on the disk, the fields associated with each pulse tend to bend away from their neighbors.

If the data were to be written with no procompensation, three conditions could occur when the data are read back from the disk.

1. **Early** - a bit would be picked up too soon in relation to the last bit read.

2. **Late** - a bit would be picked up too late in relation to the last bit read.

3. **On time** - if no pushaways occur, the data bits will be picked up right where they should be, according to MFM rules.

To counteract the bit shifting in condition one or two, when the NRZ to MFM Write Encoder and Data Precompensator converts NRZ data to MFM data, it also examines the relationship between each bit in the data stream and its immediate neighbors to determine if it is a one followed by a one, a zero followed by a one, and so forth.

The Data Precompensator uses this information to determine if the pushaway phenomenon would occur between the bits in that pair.

The Precompensator can shift the actual pulse timing for Write data in one direction or the other in 15 ns increments.

Precompensation: (continued)

This slight compensation makes the pulses read back during the Read operation appear to be occurring at the correct time.

If a bit pair looks like pushaway might cause a bit to be read early, the Precompensator will direct that bit to be written that much later.

If a pair looks like pushaway would cause a bit to be read late, the Precompensator will direct that bit to be written that much earlier.

And if the bit pair looks like no pushaway will occur, the Precompensator will allow it to be written as is.

Again, for clarity, if the pushaway effect was not compensated for, the data bits, when they were read, would be time shifted, causing read data errors.

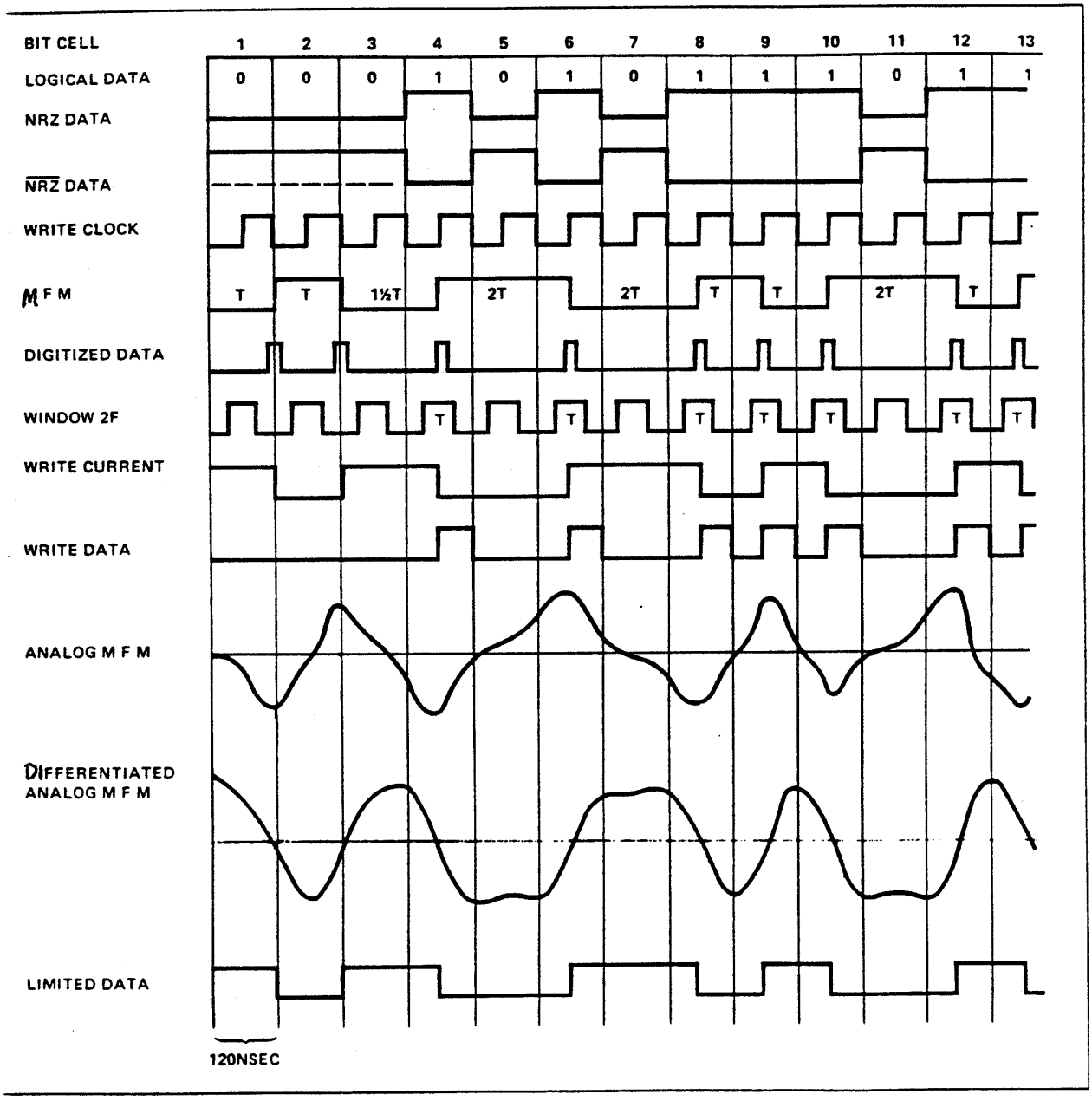
The Write Precompensator identifies the shift that would occur and compensates for it by shifting the write timing the opposite direction of the shift.

That is if a bit would have the tendency to come early it would be retarded. If a bit would have the tendency to come late, it would be advanced. If it was going to be on time, why mess with it.

That's precompensation, and it only occurs on the inside tracks, on the outside tracks the bits are far enough apart that pushaway is not a factor.

Analog PCB Circuit Descriptions

270



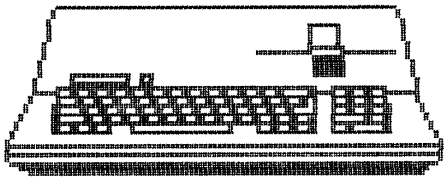
FINIS

271

5

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

DOCUMENT TITLE

Pro-File Level II Phase 1
Service Manual (Very Preliminary)

Author:

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

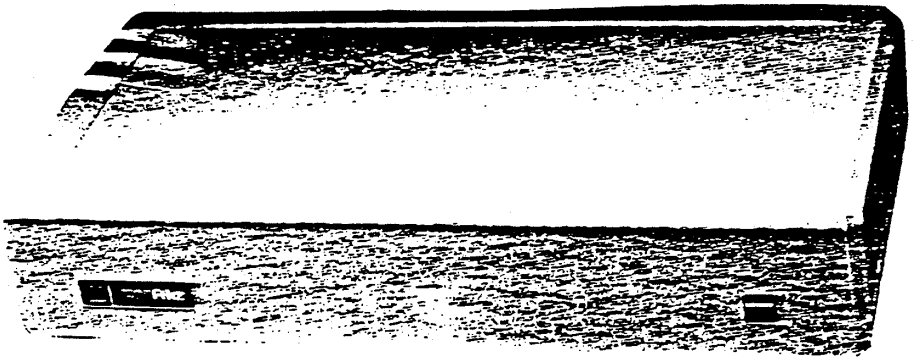
272

Pages: 101

(p. 1 not present)

PRO-FILE LEVEL II
PHASE 1 SERVICE MANUAL

VERY PRELIMINARY



273

ILE DEBUG DATA 04/19/82

US #1:

- OTHER THAN 55 RESPONCE FROM HOST
- 3 BYTE SEPARATOR BETWEEN WRITE BUFFER AND STATUS TABLES BAD
  OR S2-3 SET.
- SPARE TABLES UPDATE OCCURED,BUFFER DATA IS CHANGED.
- CANNOT READ 3 SECTORS AFTER 2 RESEEKS

- READ ERROR
- CANNOT FIND TARGET HEADER IN 9 ROTATIONS
-
- OPERATION FAILED

US #2:

- CANNOT READ 3 SECTORS AFTER SEEK
- MORE THAN 32 SPARES
-
- MORE THAN 100 BAD BLOCKS

- CANNOT READ STATUS SECTORS
- SPARING OCCURRED
- SEEK TO THE WRONG TRACK
-

US #3:

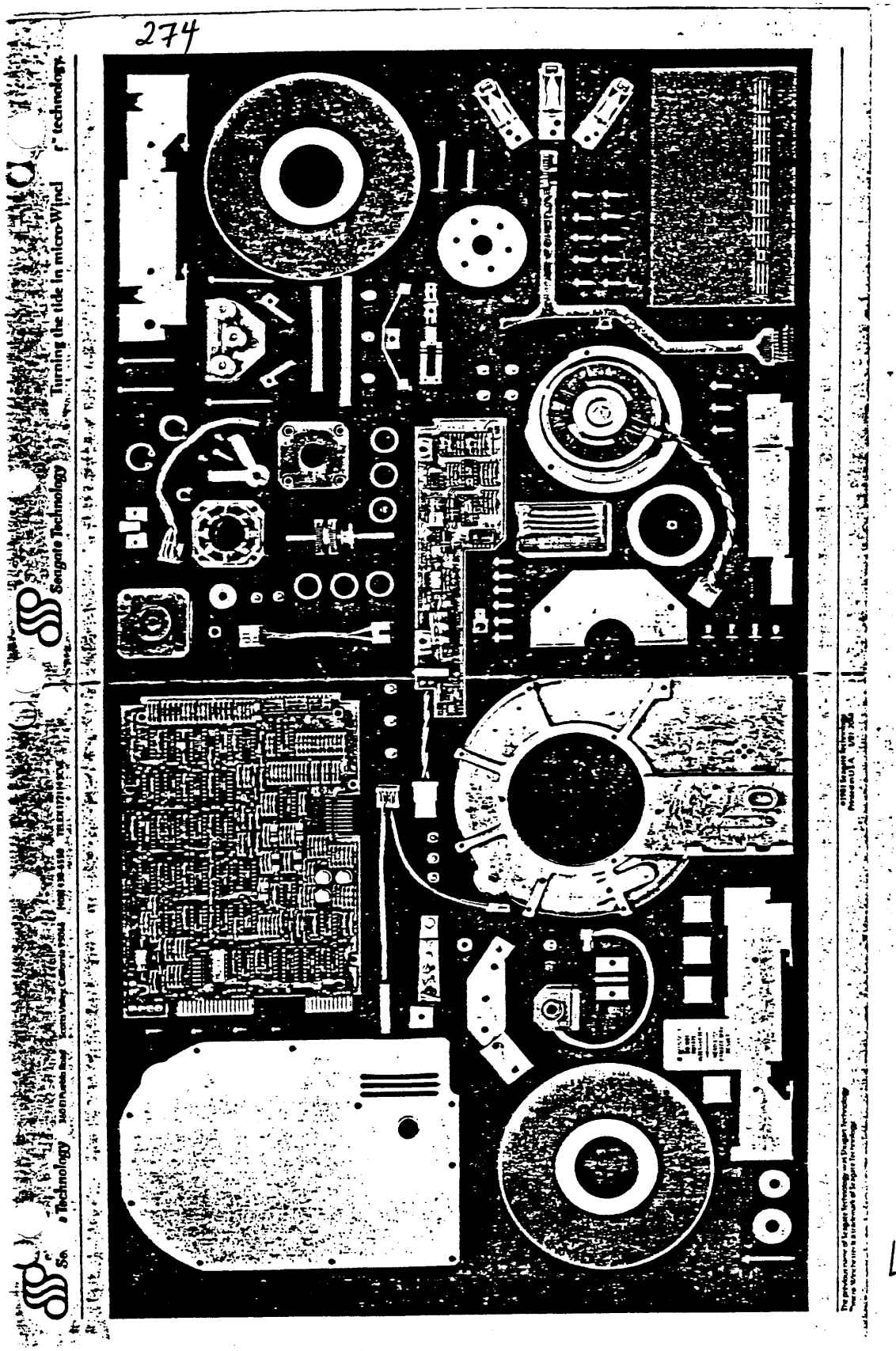
- PROFILE WAS RESET
- REQUESTED BLOCK # OUT OF RANGE
- BLOC I.D. MISMATCH SET BY HOST
-

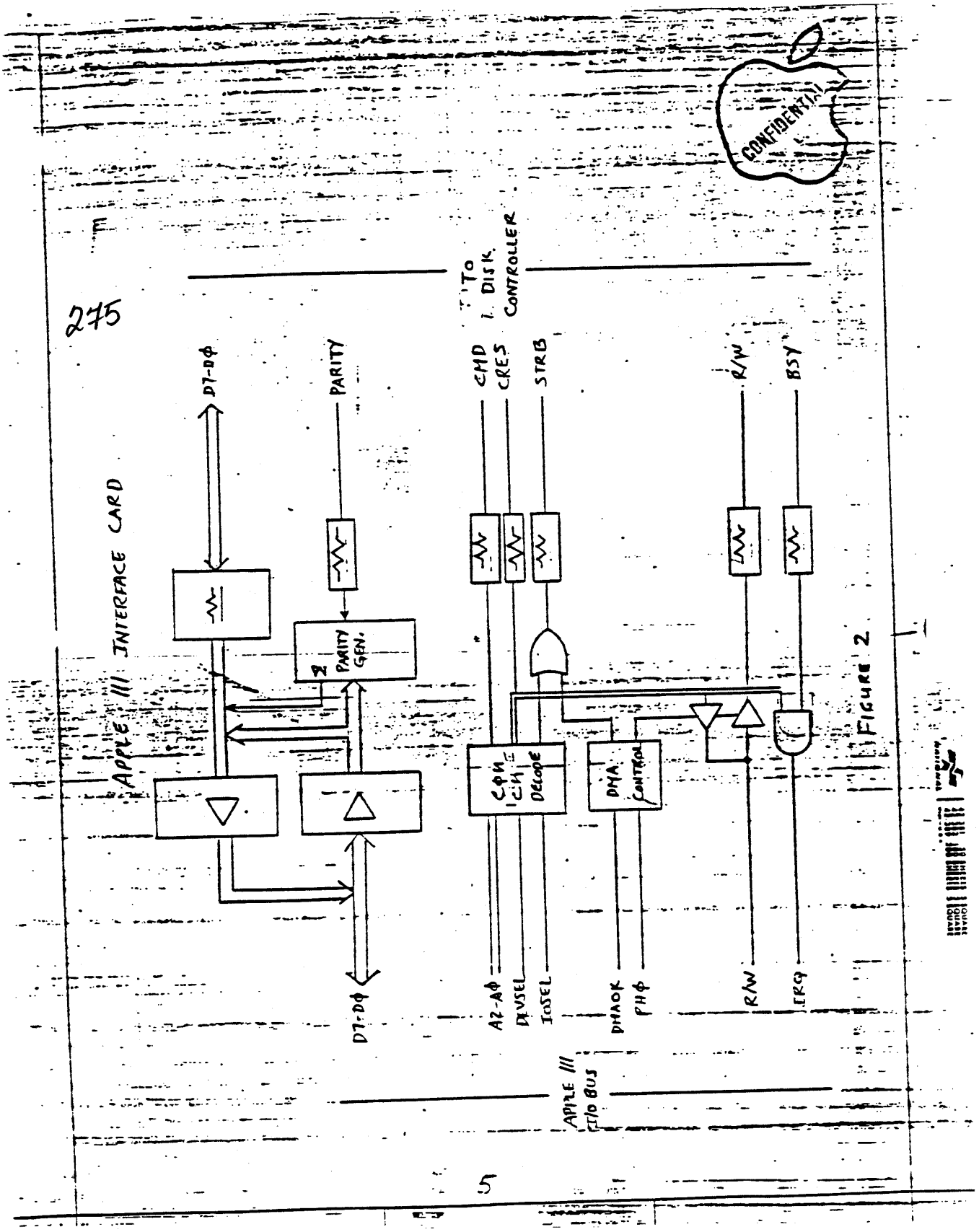
-
- HOST RESET PROFILE
- BAD COMMAND RESPONCE FROM PROFILE
- PARITY ERROR ON CABLE

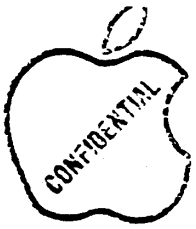
US #4:

PERCENAGE OF READ FAILURES AFTER READ COMMAND

3







276

currently supported by SOS.

E. Controller Reset (CRES)——Cn08 and Cn0C

Cn0C sets the signal high; and Cn08 resets it low. CRES resets the Z8 MPU on the controller card. Used in initializing the system after unseccessful seek attempts.

F. Write Buffer (WBUF)——C0n0

A write reference to C0n0 writes a byte of data to the RAM on the controller card. For example, STA C0n0 writes the contents of the host MPU's accumulator to a predetermined location in the controller's RAM. A read reference to C0n0 does nothing.

G. Clear Parity Error (CLRPE)——C0n3

A write reference to C0n3 resets the parity error flip-flop.

H. Read Buffer (RBUF)——C0n1

A read reference to C0n1 reads a byte of data from the controller's RAM.

I. Read Status (RSTAT)——C0n2

A read reference to C0n2 loads the state of BSY, the Parity FF, and "Open Cable Detect" onto the I/O bus (bits 7,6, and 0 respectively).

NOTTYPED val/7

6

277

~~Interface Circuitry.~~

Lets take a tour of the interface schematic. Get your pencils ready to make a few notations as we go, so later on we won't have to stop and repeat ourselves.

Data Bus

in the upper right of the drawing U3, U6 are the major data buffering elements of the bus. U1 is the resistor network. U8 a parity generator/checker is used to verify data integreity on the bus, not to generate a parity bit used elsewhere in the system, except for a error status bit.

Data from the host is clocked into U3, an 8 bit latch with tri-state outputs. The outputs are enabled when data is being transferred from the host to the controller, when DRW is low.

Data from the controller is allowed to pass through U6 to the I/O bus when DRW is high, host Q3 is low, and there is a high PSTRB. This gating is accomplished at U5-12.

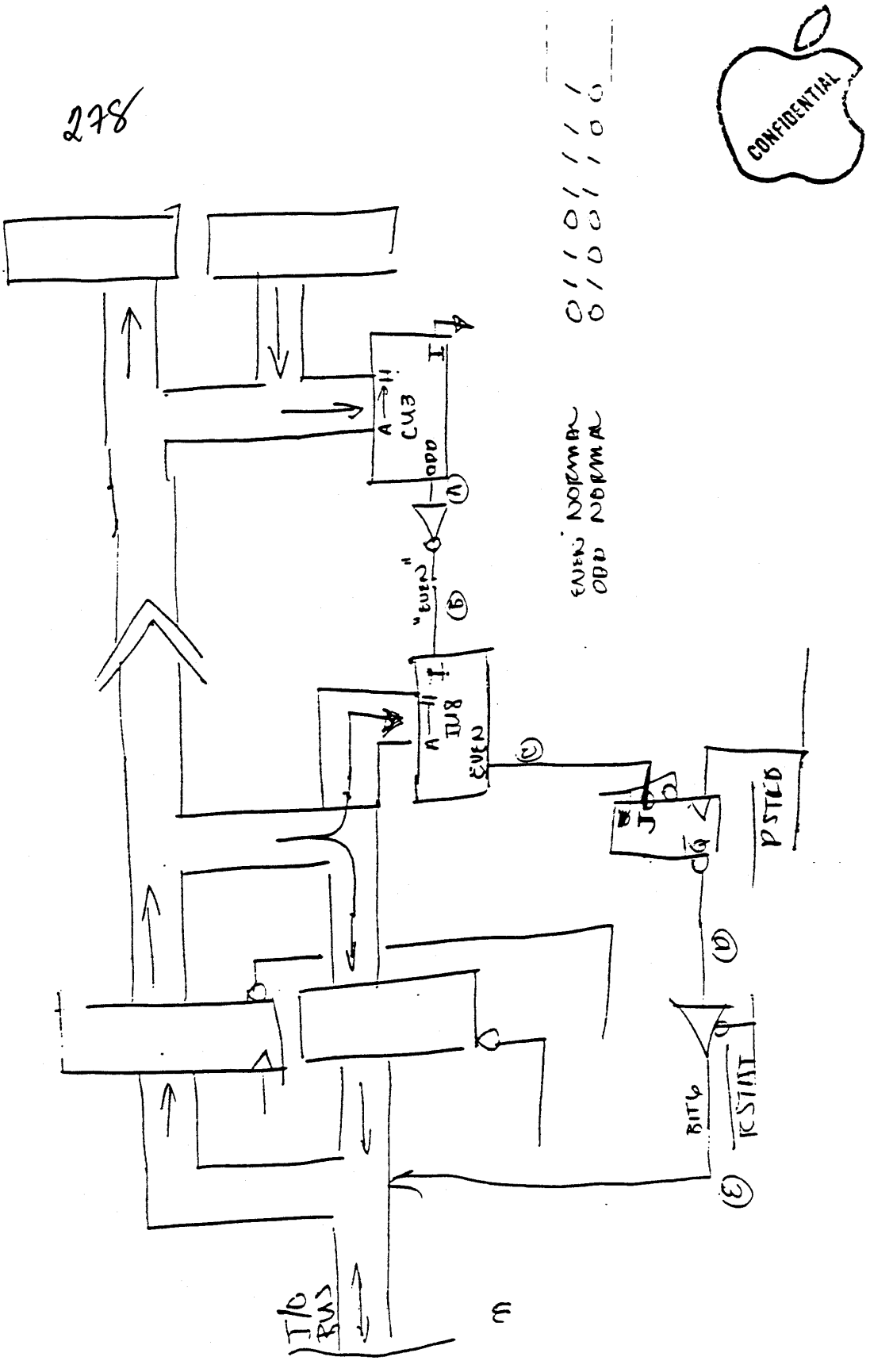
The parity circuit here is only half of the circuit, the remaining half is on the controller card. Simply there are two parity checkers, one on each end of the cable. And if the total sum of parity (number of "one" bits) doesn't compare a status bit 0 is set and stays set until parity reset occurs.

Normal parity operation. Lets say a byte of even parity is sent from the interface to the controller. The parity generator on the controller sees the Even condition and the RPARITY line is taken high. This is placed on the "carry input" of the generator on the interface. This generator sees the even data byte plus the input bit as a ODD sum and therefore takes its EVEN output low. This will be placed on the J input of U7-2, and remembering JK theory a low causes nothing to happen, the FF stays in reset condition. (/K input is strapped high, therefore the JK cannot be reset from data inputs.)

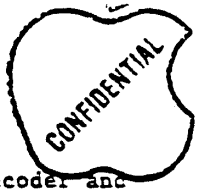
If an odd byte is on the bus, the checker on the controller sees the odd sum and takes the RPARITY line low. This on the Input of the interfaces checker with the odd byte on the bus will again cause the EVEN output to be low, again not allowing the Parity FF to be clocked to the set condition.

When EVEN output of the interfaces checker is high at the trailing edge of /PSTRB the Parity Error FF will set. When the host checks the interface status byte (RSTAT, C0n2) bit 0 will have a "one". The error could be caused by a very noisy bus line, an open or shorted cable connection or some such condition. It should never really occur, and is pretty much an over kill, but it is nice to have, for security. It is purely a passive circuit, it will not stop data transfer, only report the occurance of the first error.

Control/Command Decoders



279



In the lower left section of the drawing locate U15 device control decoder and U13 I/O command decoder and latch.

U15 is a simple 1 of 8 decoder. Once the ChEn Latch is set by an I/O Selection of the slot U15 decodes each successive DEVSEL reference and developes appropriate strobes. See address reference description in intro for interface section. It should be noted that read or write references equal to or greater than C0n4 have no decoded action.

Write References
C0n0 /WBUF low
C0n1 not used
C0n2 not used
C0n3 /CLRPE low

Read References
C0n0 not used
C0n1 /RBUF low
C0n2 /RSTAT low
C0n3 not used

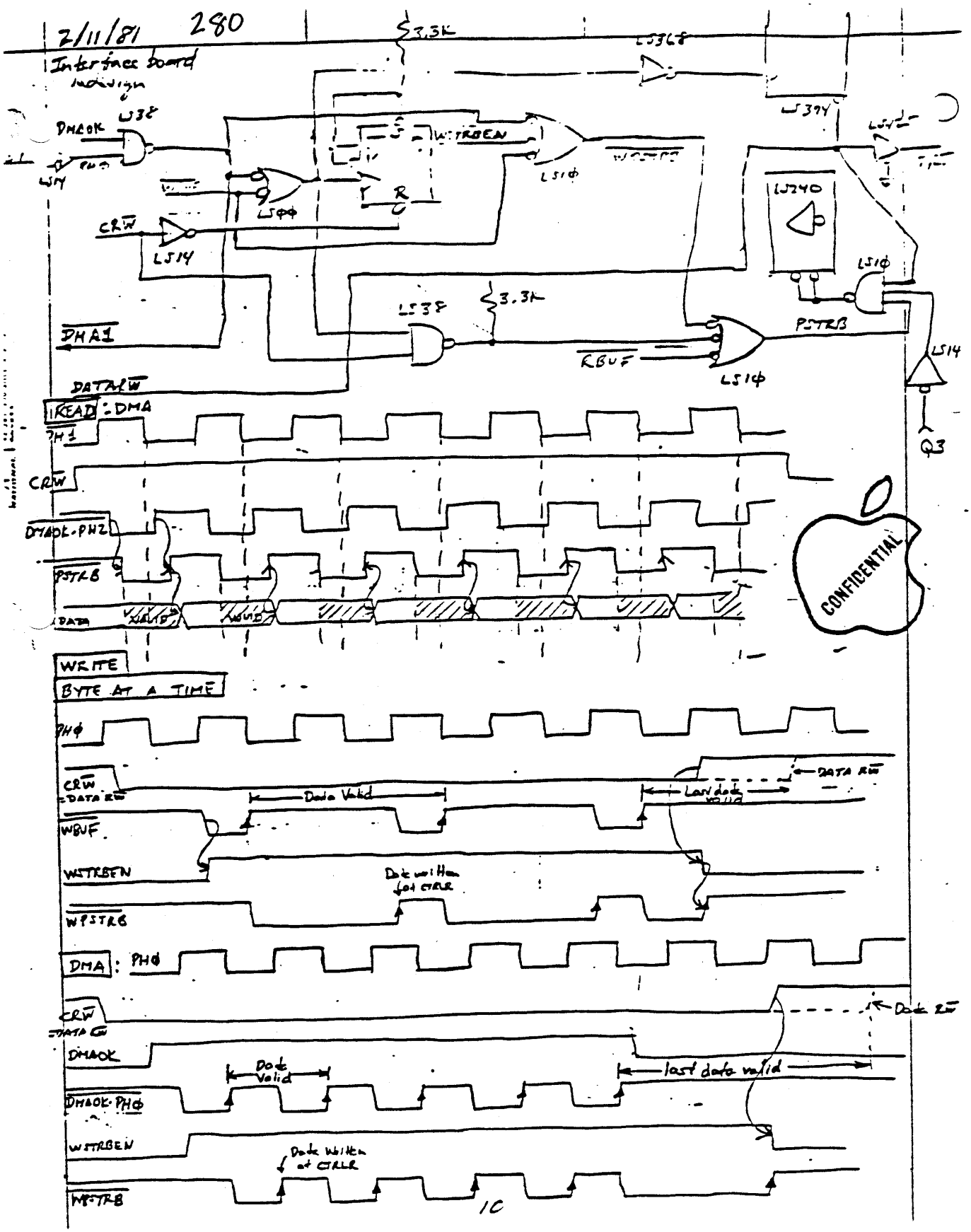
U13 is a 8 bit addressable latch (as used on the Apple II and /// main logics) It decodes which bit is being addressed by the action of A0, A1 and A3; A2 is used as the "data" input to the latch. On the trailing edge of the the I/O Select reference the condition of those four address lines are strored. It is safe to say that these are flags that are being set/reset, or at least not strobes.

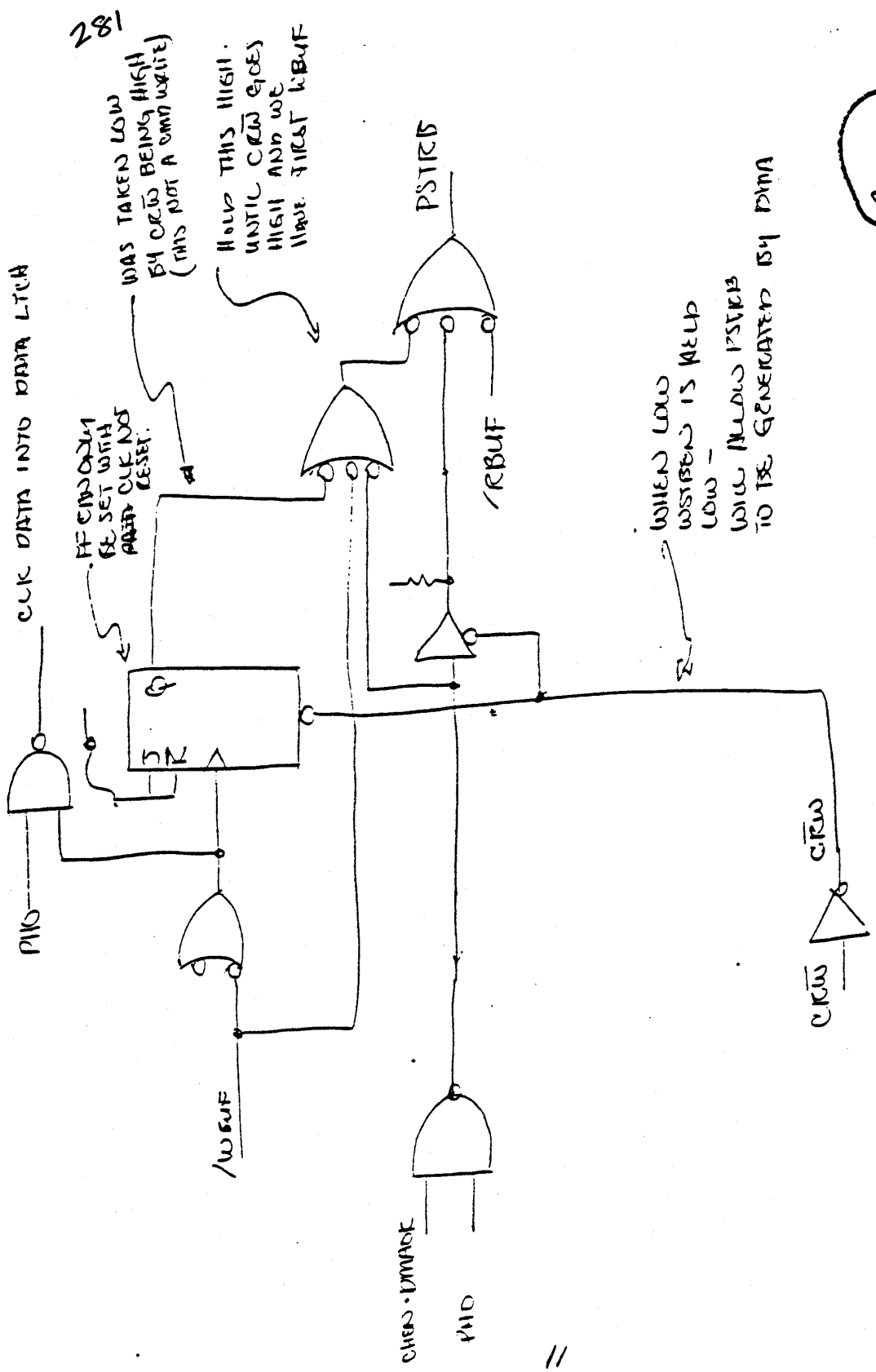
Table of Activity for U13 Latch (All addresses are in the CN0x range)

Hex Address	A3	A2	A1	A0	Action
CN00	0	0	0	0	Resets CMD low
Cn01	0	0	0	1	Resets CRW low
Cn02	0	0	1	0	Resets INTEN low
Cn03	0	0	1	1	Resets DRW low
Cn04	0	1	0	0	Sets CMD high
Cn05	0	1	0	1	Sets CRW high
Cn06	0	1	1	0	Sets INTEN high
Cn07	0	1	1	1	Sets DRW high
Cn08	1	0	0	0	Resets CRES low
Cn09	1	0	0	1	not used
CnOA	1	0	1	0	not used
CnOB	1	0	1	1	not used
CnOC	1	1	0	0	Sets CRES high
CnOD-OF	1	1	x	x	not used (where A1 and/or A0 =1)

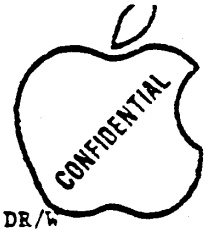
The next big highlight is the circuitry behind PSTRB. This is the master data transfer strobe which is generated anytime a byte is transferred between the controller and the host, be it read or write and/or single byte or DMA modes.

PSTRB is generated four ways. The easiest to see is through /RBUF. Each time the host requests a single byte from the controller /RBUF goes low and is or'd





283



at U5-11 and develops PSTRB, which in turn enables the data from the controller to be passed to the I/O bus. (so long we have previously set DR/W high....and Q3 is low) (aside,,, Q3 is used to gate data onto the bus to avoid signal contention on the bus and the resultant noise when the various drivers switch on and off)

The next method is byte at a time writing. The host must first set CR/W low which enables the WSTRBEN FF to be set with the first edge of /WBUF. Then a PSTRB is developed with each /WBUF. Because of the action of the gate which develops /WRSTRB (U5-6) the real write strobe is the space after the /WBUF signal and the next working edge of /WBUF. After the last /WBUF the last byte is transferred into the controller by the action of CR/W being taken back high; thus, resetting WSTRBEN FF which will give the last positive edge of /WRSTRB. This method is used in writing the commands string to the controller.

It should be noted that if CR/W is held high /WBUF can be used to write a byte of information to the data latch which will not be transferred to the controller RAM....*****used to load the "55"?

PSTRB can also be generated in a DMA transfer of data to the controller. Again the WSTRBEN FF must be set with the first positive tranisition of U11-11. The Apple three conditions itself for a DMA and lowers /DMAOK this is gates /PH1 (PH0) to U11-13 which develops automatic clocks to the data latch and to the WSTRBEN FF. This auto strobing goes through the enabled U5-6 and then to U5-8 as PSTRB.

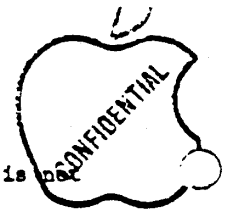
(Background: The DMA talked about here is not a true DMA where the remote device takes control of the address and data lines. It is really more of a block transfer mode which is controlled by a special routine in the boot rom. The data direction is controlled by the Cr/W and action of the automatic pulses generated by u12-6 described above. SOS sets a page in the ZPAGE register and then calls a routine in ROM which causes the MPU to cycle through the address of a page (itself running in its ZPAGE which causes the ZPAGE register to be gated onto the Address buss. It is all very fixed in nature so once called it will go through the looping routine in ROM until it falls out and goes back to normal stuff.)

The remaining method of generating PSTRB is a DMA transfer from the controller to the host. In this case CR/W is left high, which is inverted and is present at the input of U10-9. The Apple /// conditions itself for the block transfer and lowers DMAOK. The same autogeneration of pulses occurs at U12-6, but this time the buffer U10-4 is enabled and passes these pulses directly to U5-10 and PSTRB is toggled.

Remember we are transferring data from the controller to the host. That means we must be writing to the Apple /// memory. The pseudo write pulse on the R/W line is accomplished with the output of U10-8 which is tied to the R/W line of the I/O bus. The auto-strobes of U12-6 enable the low at U10-9 to be passed to the output.

The gate U12-11 can be used to enable an IRQ to the host when INTEN is high

284



and the appropriate state of the BSY/INT line is reached. This however is not implemented in the initial SOS Driver.

The remaining logic is either drivers or very elementary bit buffers used to gate the interface status bits to the I/O bus.

SPECIAL NOTE: the R-C network at U12-3 and the inductor on U12-8 are used to squelch an oscillatory condition which could occur in the logic supporting the DMA process.

285

Analog card

The analog card serves as the interface between the controller card and the Seagate HDA. It consists of the NRZ to MFM write encoder, write driver, head select matrix, AGC preamplifier, read detector, PLL, MFM to NRZ decoder, and sector detector.

The head select matrix selects one of the four heads for a read or write operation. During a write operation, serial NRZ write data and clock (5 MHZ) are converted to MFM write data. The MFM write transitions are peak shift precompensated on the inside cylinders, 128 to 152. The write driver converts the MFM write pulses to alternating head current. On the inside cylinders, the write current is reduced from 24 to 19 ma peak.

During a read operation, the automatic gain controlled preamplifier amplifies the low level (.6 to 2.0 mv) head signal to a fixed 1.0 v output signal. This analog read signal is fed to the differentiator and dated detector do derive the MFM read pulses. The phase lock loopp synchronizes its VCO to the MFM read pulses and provides the clock for the MFM to NRZ data seperator. The serial NRZ data and clock are sent back to the controller for deserialization.

During the format (initial set up of the disk) operation, sector boundaries are written as DC erased 10 byte fields. During a read operation, the sector boundary detector looks for the absence of read signals and sends a sector pulse to the controller at the end of the 10 byte field.

Block diagram

For the first pass lets divide the diagram into sections. The top left is the only write circuit which drives the heads during the write operations. The rest of the drawing is used in the read mode. The one of four decode head selector is shared for both read and write.

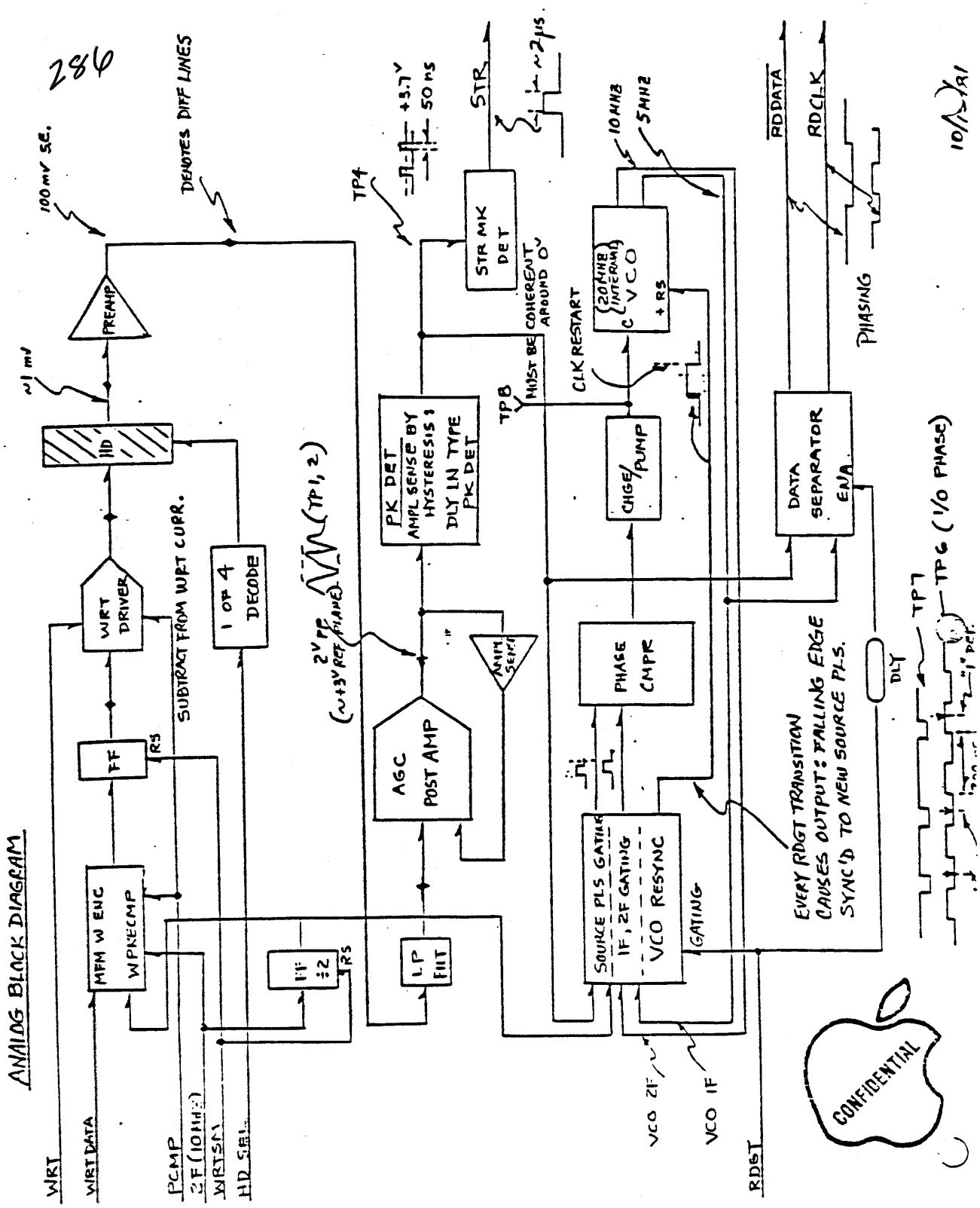
During a write WRTDATA is shifted through the MFM encoder/precompensator to the write driver which drives the selected head of the drive. It is time to define MFM and precompesation. MFM is a scheme in which the ones and zeroes of a digital bits stream are converted to a stream of transitions (reversals) following certain rules. In reference to the bit cells and their respective boundaries, a zero causes a reversal at a boundary and a one causes a reversal at center cell time. However no reversal can be 1/2 cell time from a previous reversal. So the following rules come about:

A. Ones always cause a pulse at center cell time.
B. Zeroes cause a pulse at leading cell boundary if not precedeed by a one.

Simple? Well lets look at a bit stream of 001011000.

```
cell time     !.....!.....!.....!.....!.....!.....!.....!.....!.....!
digibits        0     0     1     0     1     1     0     0     0
pulses        ^.....^.....^.....^.....^.....^.....^.....^.....^.....
```

You can now see that pulses occured at the leading cell time of the first two



16

287

zeroes. Then a pulse occured at center cell time of the first one. However the pulse for the third zero had to be dropped because it would have occured at a half cell distance from the preceding pulse. The second and third ones occur at center cell (remember all ones cause pulses at center cell and its the zeroes that get played with). The pulse for the fourth zero gets dropped because it was preceeded by a one. The fifth and sixth zeroes cause pulses at leading cell boundaries normally.

Now, that is MFM, Modified Frequency Modulation. Frequecy? Why I thought you'd never ask. Its 5 MHZ or sometimes referred to as 1F.

What pray tell, then, is "precompensation". Well, when one records things at high density on magnetic media a phenomena occurs called, "don't get too close or I'll push you away." This is in reference to the small magnetic fields of associated with each one of the pulses. Lets look at the pulse stream of the above example again:

```
pulses      ^.....^.......^..........^.....^.......^.....^.....^
time        !  1  !  1.5  !    2.0   !  1  !  1.5  !  1  !
direction   <     >       >          <     >       <     *     >
of push
```

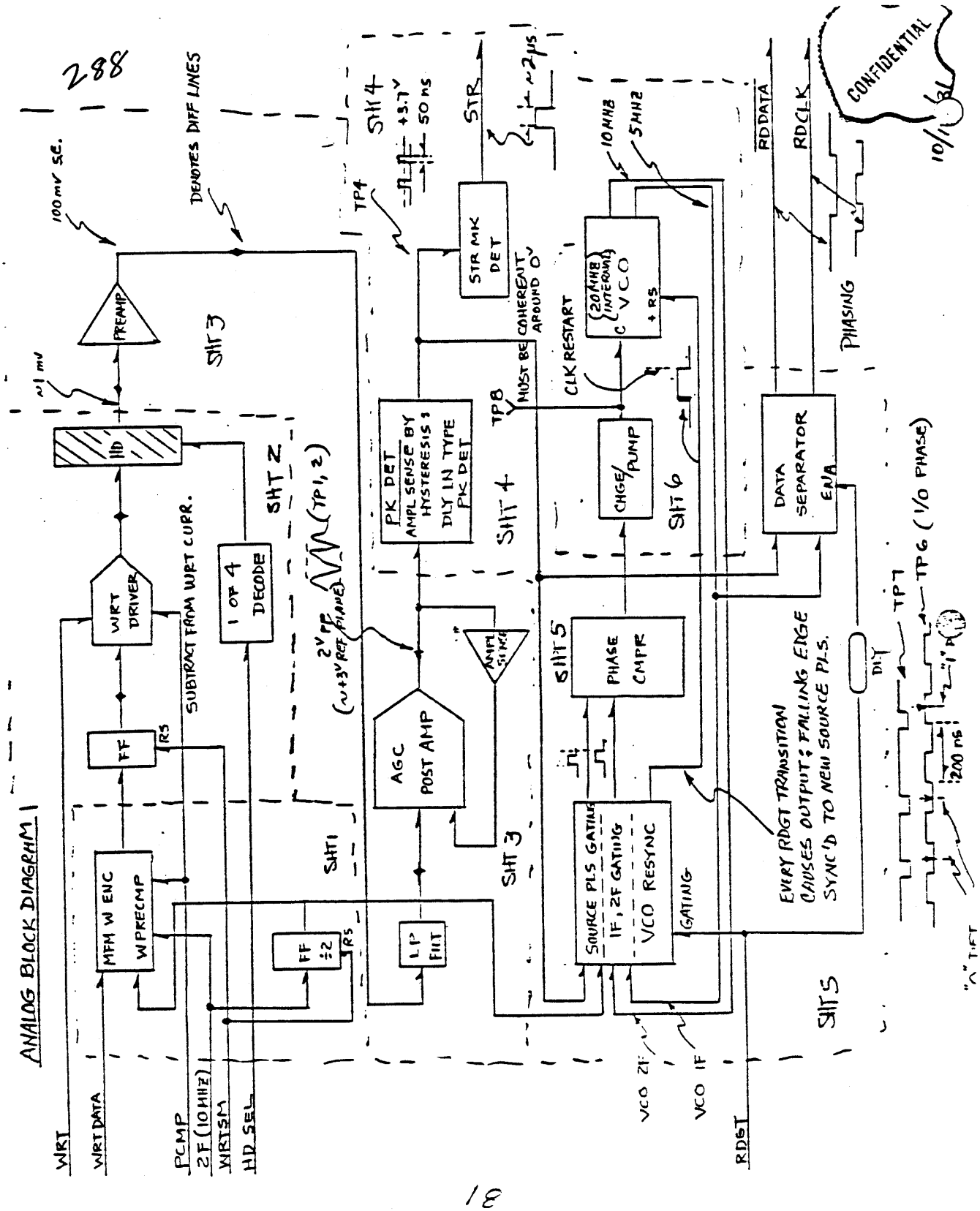
The fields associated with each pulse tend to bend away from their neighbor, which results in a slight time shift, early, ontime, or late in reference to the pick up (read) head. To combat this shifting the precompensation circuit which is encoding the pulse stream looks at the string and can shift the actual pulse timing one direction or the other in 15ns increments (during write operations). This very slight compensation will make the pulses read back during the read operation appear to be occuring back at the "correct" time.

Again, for clarity. If left alone the fields when read would be time shifted. The write precompensator shifts the write timing the oppposite direction of the shift. That is if a bit would have the tendency to come early it would be retarded, if a bit would have the tendency to come late it would be advanced. If it was going to be on time, why mess with it. That's precompesation, and it only occurs on the inside tracks, on the outside tracks the bits are far enough apart to not mess with one another.

Their is more to the read circuitry. The very small signal from the selected head is amplified and then amplified again (with automatic gain control) then the peaks are detected and are used to keep the VCO in tune and are decoded from MFM to NRZ. ~~If~~ there are no peaks ~~for a~~ while a sector mark is detected.

That wasn't too much description, but thats all that happens. It just takes a lot of circuitry to do it.

17



289

ANALOG CIRCUITRY IN A LITTLE MORE DEPTH

For easy of working with the document a multisheet drawing of the analog schematic has been prepared. (there are some very minor component value changes from this drawing to the production version....but not enough to not understand the way it works)

A copy of the analog block diagram with dotten boxes around circuit groups have the sheet numbers of the analog schematic for reference so you can relate the detail to the overall picture.

Section One Write MFM Encoder and Write Precompensation.

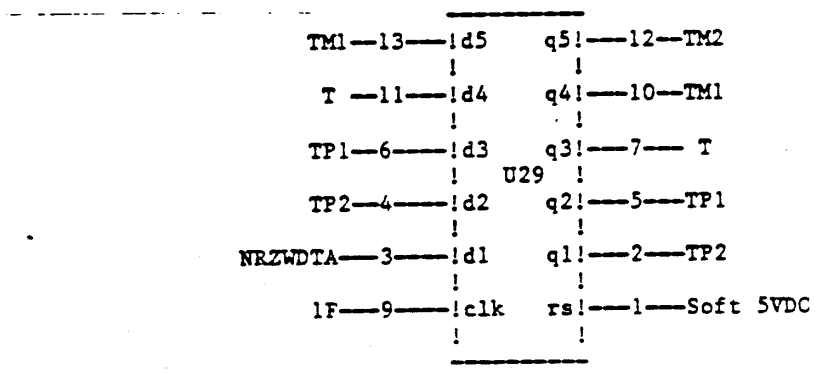
The circuit population of this group is quite small however the PAL (programmed array logic-U28) makes up for that in actual equivilant logic. Refer to appendix on PAL's

Clocking—

The 2F (10MHZ) is the master controller crystal oscillator frequency. It is divied to 1F by U2 (a simple divide by two stage). The 1F (5MHZ) is the bit shift clock for the precompensation shift register. It is also used to keep the VCO sync'd during the write operation. 2F is used to update the outputs of the PAL register.

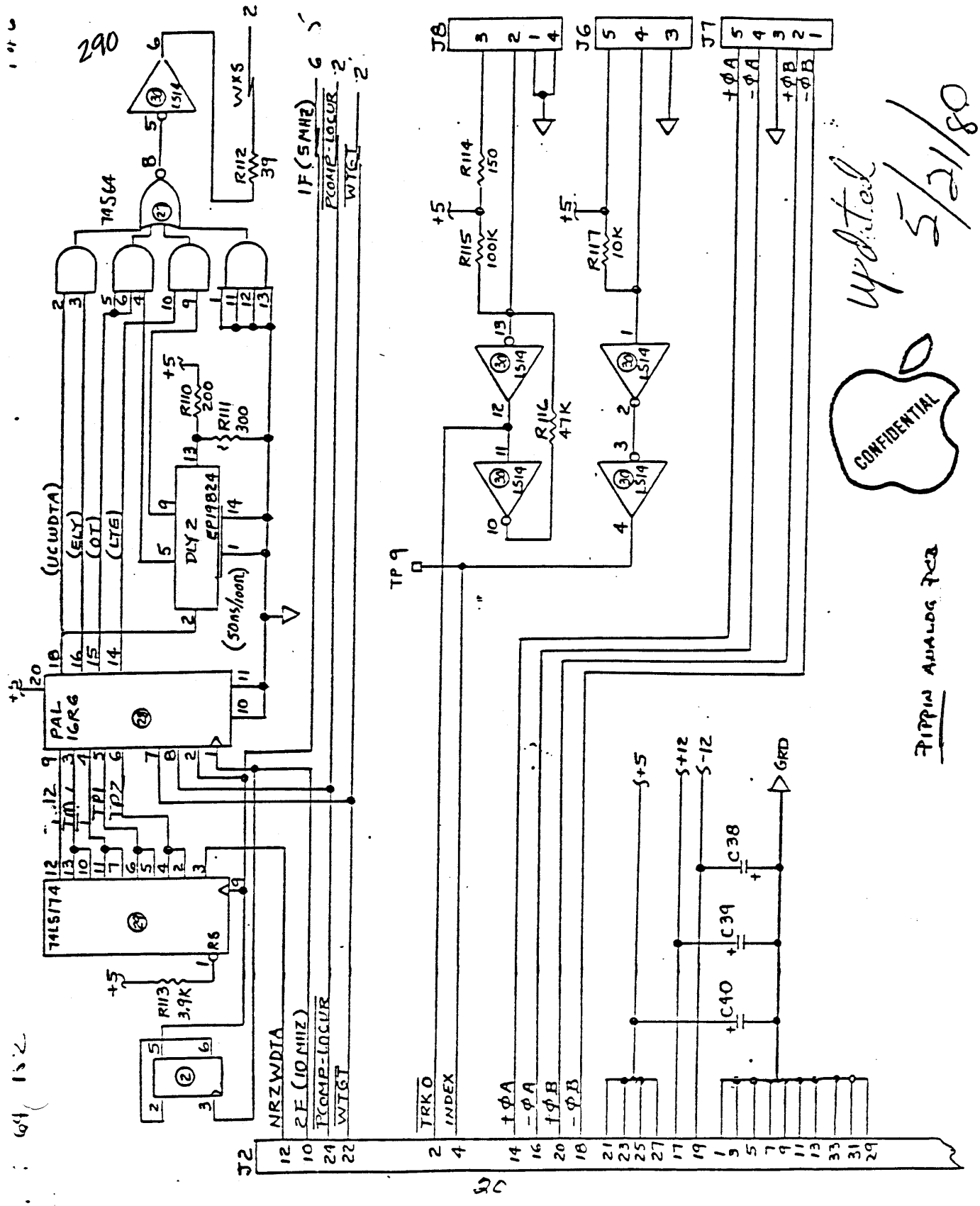
PRECOMP SEIFT REGISTER

NRZ data is shifted into U29 (74LS174) which is externally wired into a shift register (normally a hex d-type latch). This string of delayed bits form five of the inputs of the PAL. For clarity the following is a schetch of the same part with the "in's" on the left and the "out's on the right with the names of the signals (used in the PAL program listing) supplied.



The initial NRZ data is clocked into q1, the next clock places q1 at q2 and so forth. The whole intent is to give the PLA a window on the data stream to make the encoding and early/late/ontime precompensation decisions discussed

19



291

earlier. It makes things easier visualize the data stream in reference to the bit named "T". The TMx signals are what T was 1 and 2 clocks ago, and the Tx signals are what T will be in 1 and 2 clocks.

MFM/PRECOMP PAL

To keep us going the PAL (16R6) is a custom array of logic designed for just this circuit. It consists of a hugh array of AND/OR gates feeding each D input of the output register. Each output is represented in a formula in the appendix. The equivilant logic is also in the appendix, and if conventional 7400 logic were to have been used the chip count would be increased by about 16 to 18. This little gem saves a lot of landscape (and money).

Using the information previously divulged the PAL's function is simply to make the NRZ digital stream in to a stream of pulses conforming to the rules of MFM stated earlier. The other function is to provide the gating signals to advance, retard or leave alone the stream of pulses called "UCWDTA" (uncompensated).

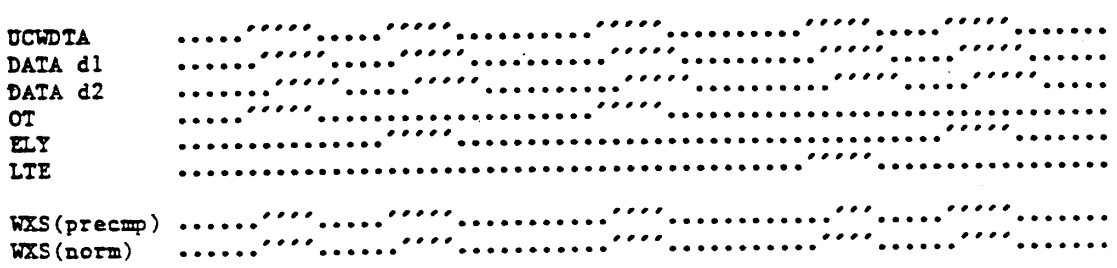
PRECOMP GATING

UCWDTA goes to the input of a gate enabled with ELY and to a delay line which delays the pulses approximately 15 ns per stage. The outputs of the delay line are gated with OT and LTE respectively.

Simply if there is no precompenstation (outer tracks) the signal OT gates the first delayed data line to the WXS line. If precomp'ing then if the data transition needs to be advanced ELY gates UCWDTA to the WXS line; and, if the signal needs to be retarded the signal LTE gates the second delay of data to the WXS line.

The signals of UCWDTA, ELY, OT, and LTE are syncronous and only one of ELY,OT, or LTE can be true at one time.

The following timing diagram might help clear it up a little.



NOTE: this diagram is for training ppurposes and does not necessarily conform to to the rules of precomp.....OK?

The leading edge of the WXS singal is the working edge that toggles the WXS FF the very sight shift of timing between the precomp and unprecompensated signal

21

202

is enough to increase read data recovery by several magnitudes.

MISC Circuits on sheet 1

The little buffers shape the TRACK 0 sensor signal and the INDEX pulse from the HDA. The Track 0 circuit forms a little noise latching function (the source signal is pretty ragged).

WRITE CURRENT GATING AND HEAD SELECTION (SHEET 2)

The WXS signal developed in the precomp gating circuit toggles the WXS FF,U2, with each pulse that is developed. So, the result is a divide by two function.

The outputs of the WXS FF are level shifted by the first set of transitors in U1. The remaining transistors comprise a constant current switching network. And they follow the command of the WXS FF. The WXS FF is clamped reset during the hardware format (Write Sector Mark) routine. This will cause the heads to write a field (10 bytes long) with no intelligence. Remember a string of zeroes would result in a 2.5 MHZ signal at the heads due to the MFM encoder.

WTGT will allow the current switching circuit to turn on when WTGT is true. WTGT (Write Gate) ,when inactive, will kill both sides of the level shifter and therfore turn off both sides of the current source effectively killing all write current.

The signal /PRECOMP*LOCUR will cause a reduction of write current to the heads when writing in the inner tracks. The normal write current is about 25 mils but in the inner tracks it is reduced to slightly less than 20 mils.

Head selection is done by U4 (7445, one of ten decoder). The signals HS0 and HS1 are binary selections of the head from the Z8. Only one head can be selected at a time. The heads are center tapped and the center is connected to the 7445. The selected head now has a current path for reading or writing.

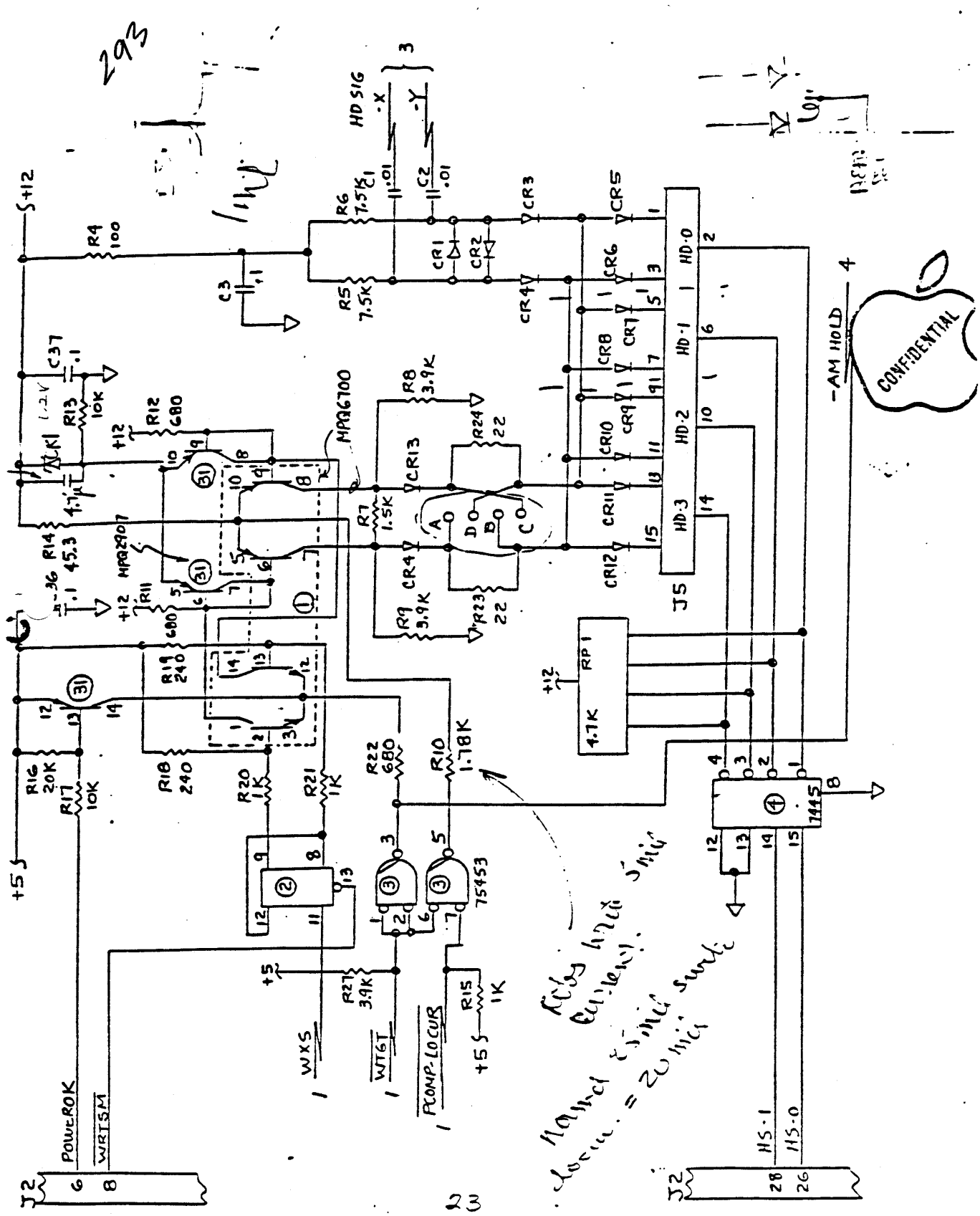
The read circuitry is always on...however it goes into saturation during the write operation and does not reliably recover the data.

During reading (not writing) the selected head develops a signal on the -x/-y lines. This signal is about 1 mv and usally cannot be seen with conventional techiques. Probing dampens the signal and induces so much noise it is obiterated. So don't expect to see data right off the heads.

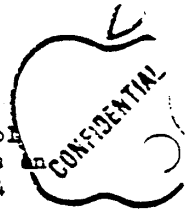
AUTOMATIC GAIN CONTROLLED AMPLIFIER (SHEET 3)

The read signals from the head are brought to the first stage preamplifier. The devices used here are ECL, but are not used in their intended application or environment. (Don't worry it works fine). Each stage of the preamplifier has a gain of 12. This means that the approximate 1 mv head signal leaves the first stage at about 18 mv. The signal is amplified again, and leaves this stage at about 200 mv.

Thats nice because the filter network attenuates it about 60% just before it



294



gets to the next amplifier. This amplifier has a circuit which will control its gain. Q4 (FET resistor) is the feedback port. By controlling the bias on the gate of Q4 the gain is controlled shunting effect the FET has on pins 4 and 11 of U6.

The output of this amplifier is reference shifted by the transistors of U7. The rest of U7 and the stages of U8 are the gain control for the FET. Simply if the amplitude of the incoming signal has peaks less than previous peaks the gain will increase. If the peaks are higher than previous peaks then the gain will be reduced. A normal occurance is that as the heads traverse the sector mark the AGC will cause the amplification to go to maximum. So as the data immediately returns after the sector mark the amplification is usually far too high and one can see very strong signals just after the sector mark but they quickly regain the normal gain. Conversely, during the write operation when there is a very strong signal present at the x/y lines the gain is reduced to the minimun, and the circuit requires a little time to recover normal gain when writing stops.

READ DETECTOR AND SECTOR DETECTOR (SHEET 4)

The Read Detector simply cleans and shapes the signal and by delaying itself in one path clocks itself through U11 and the action of U10 and C20 decern transitions of U11 and develop 50 ns pulses at each transition.

The RDPLS go on to be seperated but perform another function on this sheet. The RDPLS is gated down through U12 trannys which pump a little charge to C23 at the input of U13. If there are no read pulses (as during a sector mark) R71 slowly discharges C23 and causes U13-12 to switch. When read pulses reappear (as in the sync field) C23 quickly recharges and U13-12 flips back. This direction of flip is intergrated to U13-10 and an approximate 2usec pulse is developed, SECTOR. This is directly connected to the Z8, so it can do its thing.

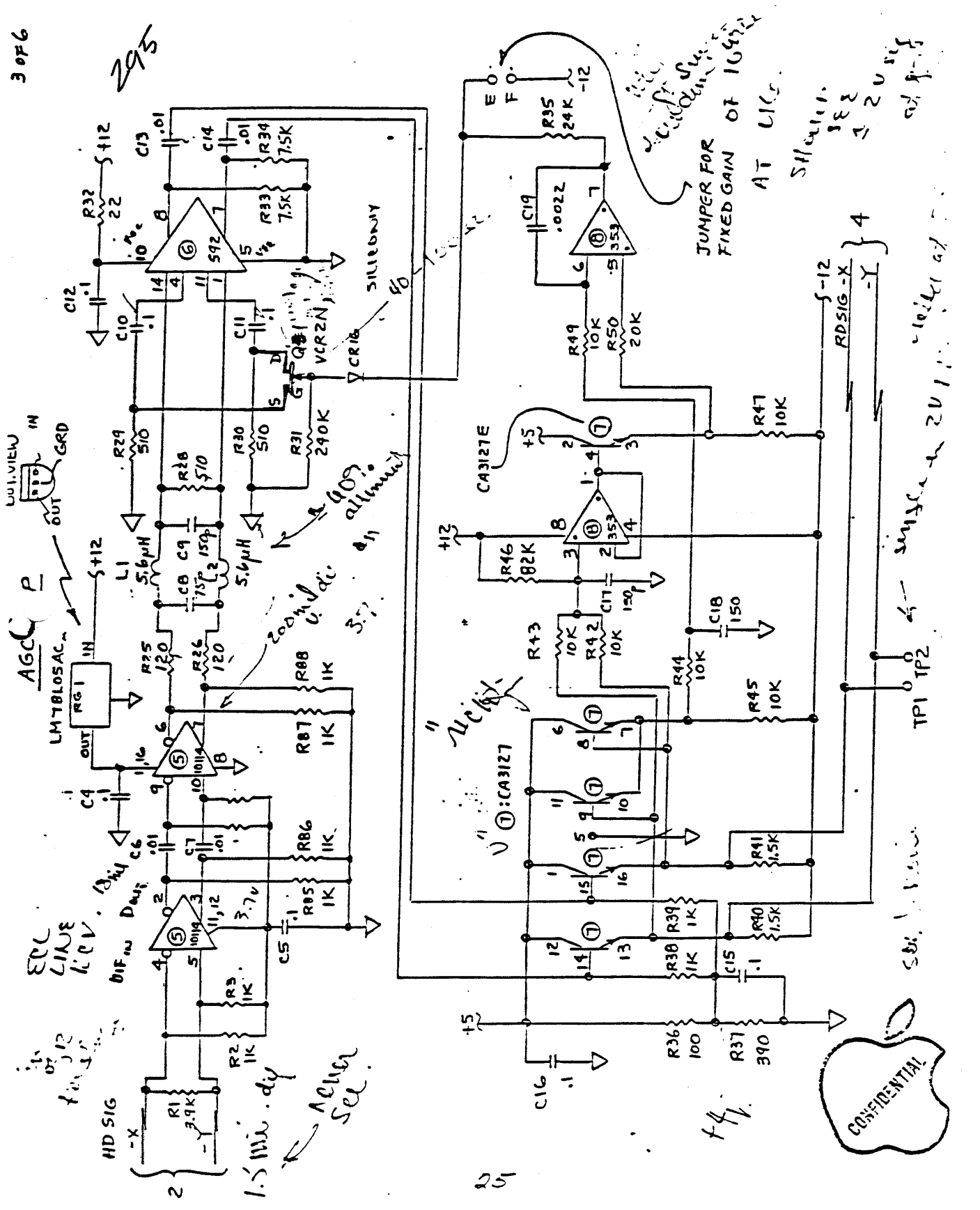
SOURCE PULSE GATING
PHASE COMPARATOR
DATA SEPERATOR (SHEET 5)

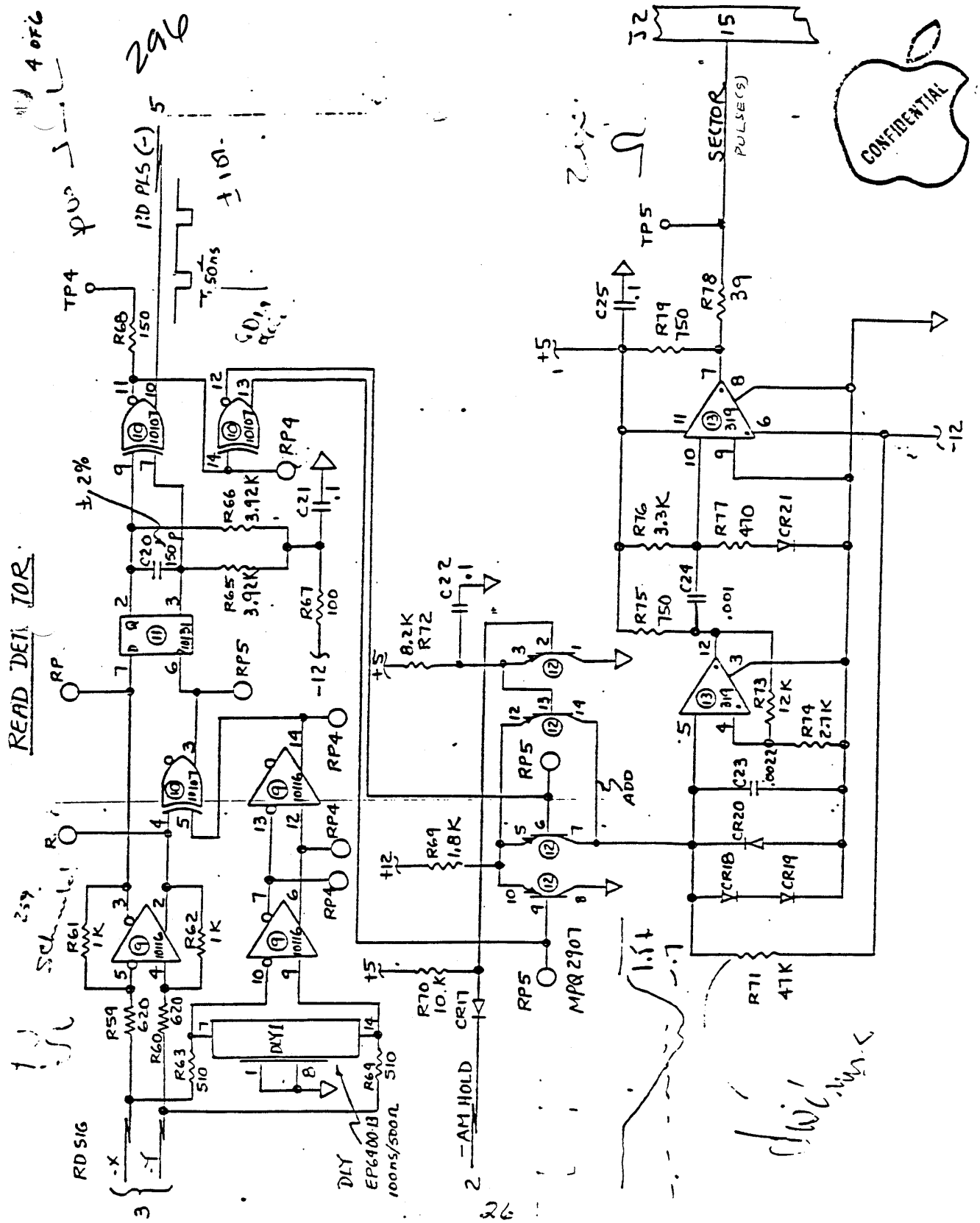
When we are not reading (RDGT false) a lot of the circuitry in this section is disabled. The major activity is to keep the VCO at near optimum frequency and in phase with the master oscillator clock. (the fact that it is in phase with the master clock is a irrelevant fact, it just ends up in phase.)

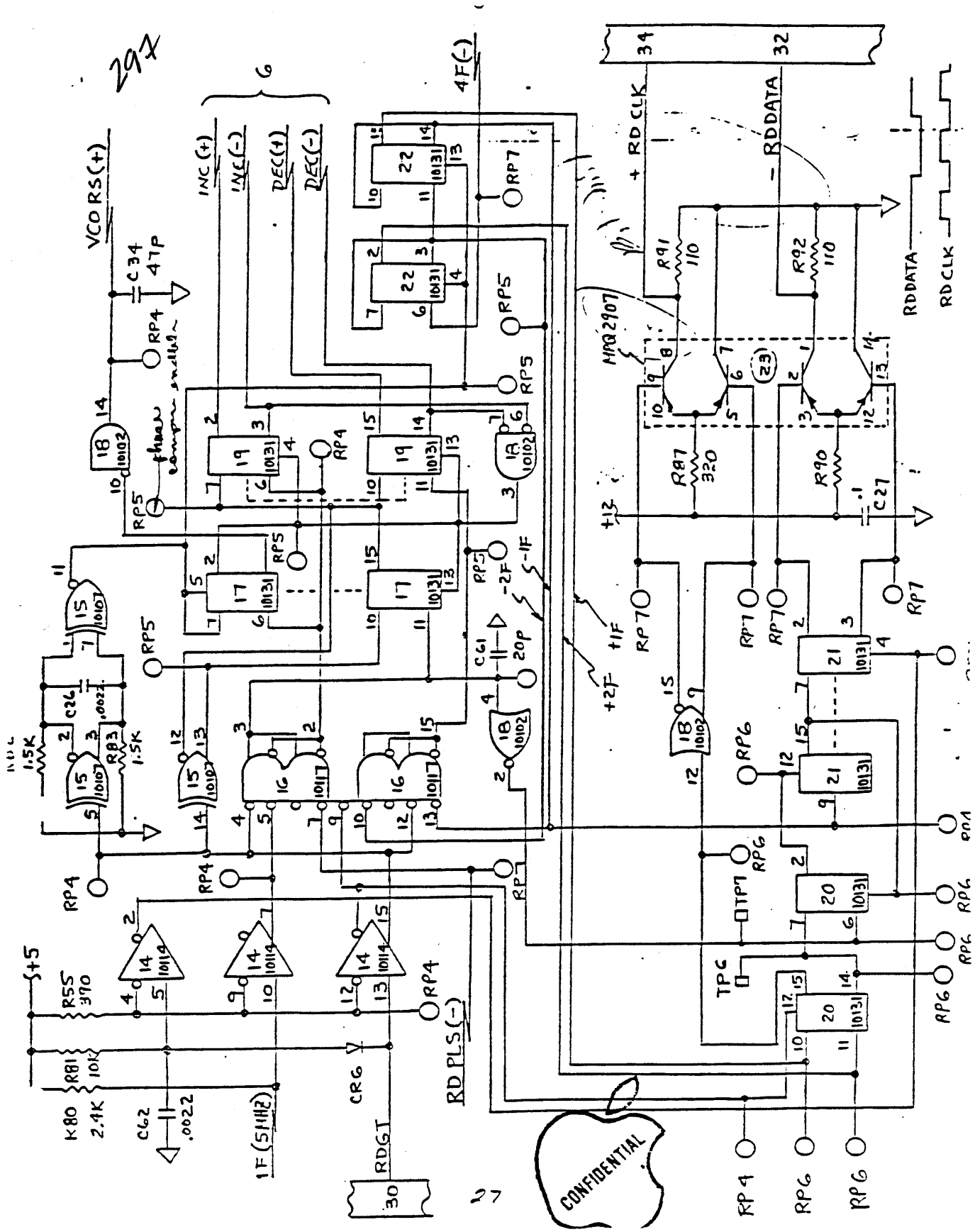
When not reading, the master osc derived 1F (5MHZ) signal is gated through to the phase lock loop detector. The actions of U17,U19 in relation to VCO's derived 1F clock will cause the VCO to lock on the edges of the master clock and keep at frequency. Also, when not reading, the data seperator is turned off.

When reading and RDGT is true, the phase comparator is now looking at the edges of the data pulses and locks on their edges. If the pulses are coming a little faster than the VCO frequency the VCO is told to INC. If the VCO is faster than the frequency of the pulses it is told to DEC. An INC/DEC decision is made at each data pulse. If the VCO is right on no change is commanded.

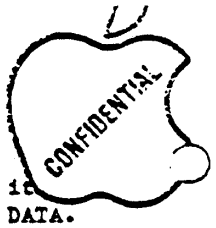
24







296



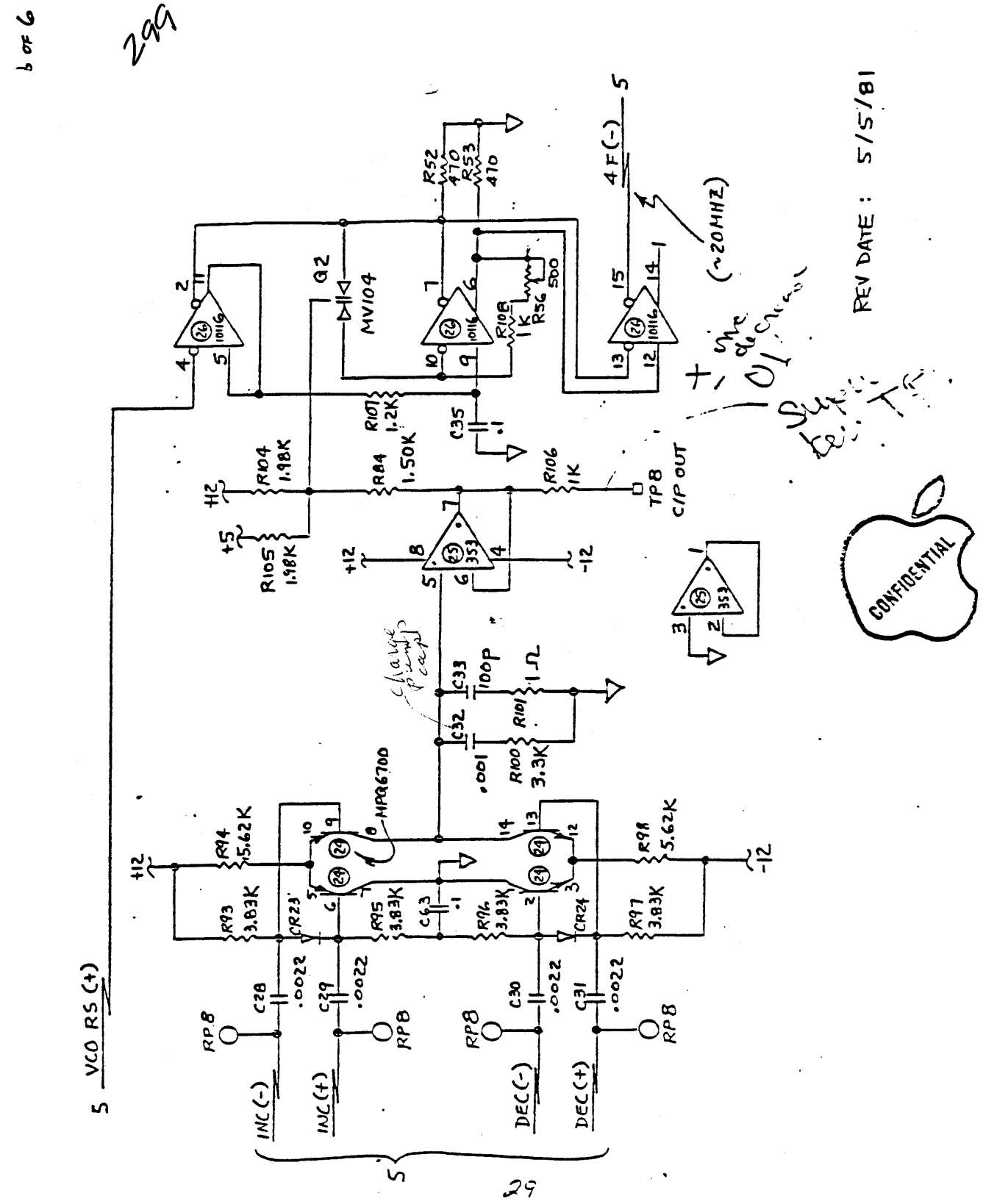
The RDPLS's are gated down to U20 which takes the pulse stream and beats it against the functions of 1F and 2F (VCO). The results are RD CLK and RD DATA.

VCO (SHEET 6)

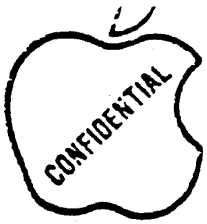
The VCO is comprised of three elements. The RESYNC GATE, the INC/DEC control, and the Oscillator itself.

U26 pulses very quickly at the beginning of each read gate on the first read pulse. This action syncronizes the VCO to the phase, if the VCO starts getting out of phase it is resync'd.

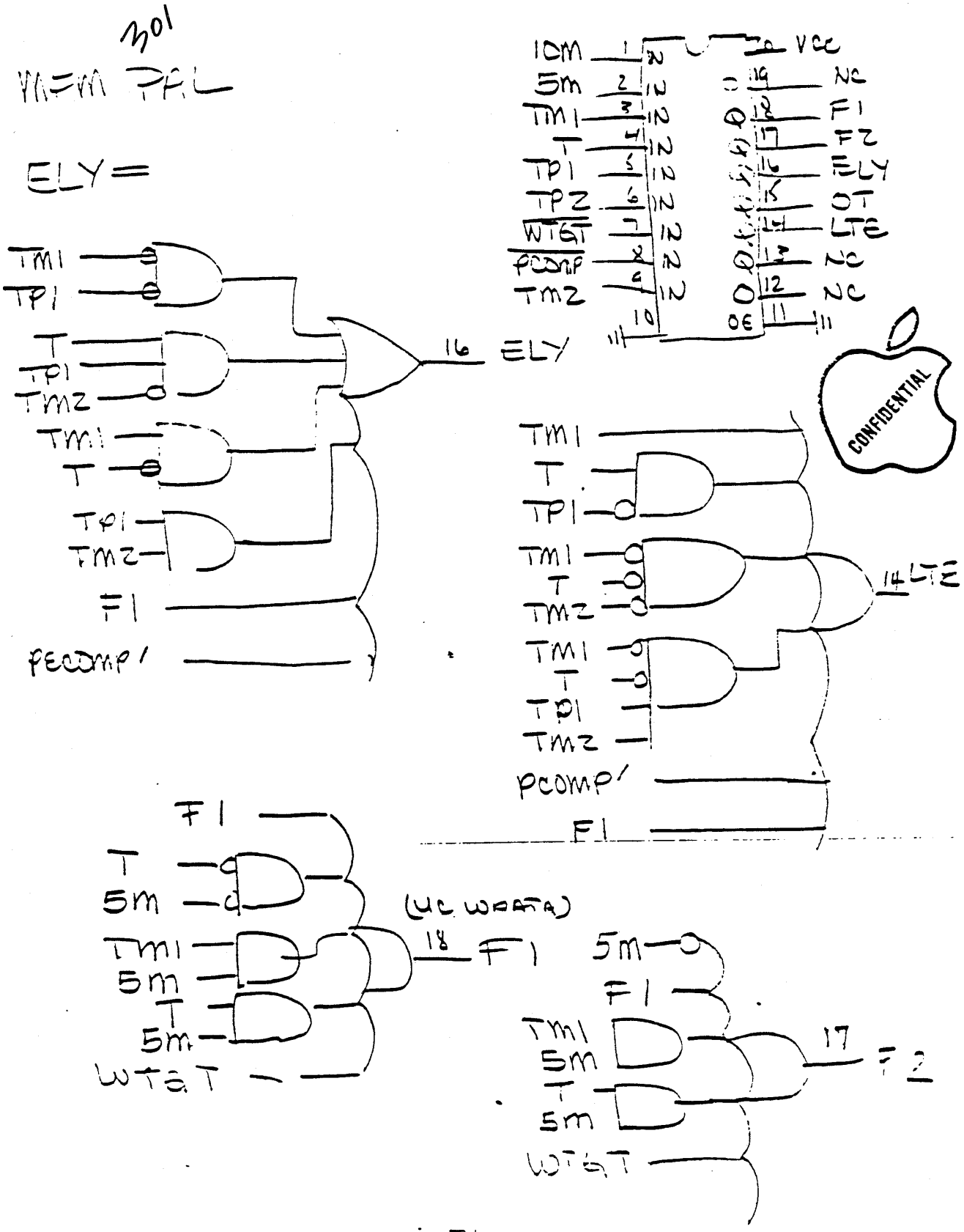
The INC/DEC pumps more charge onto C32 or decreases the charge on C32. The amplifier reads this and alters the potential at the gate of Q2. This bias shift will affect the resultant frequency of the oscillator. Swings in the bias are directy proportional to swings in frequency. The oscillator centers on 20 MHZ.



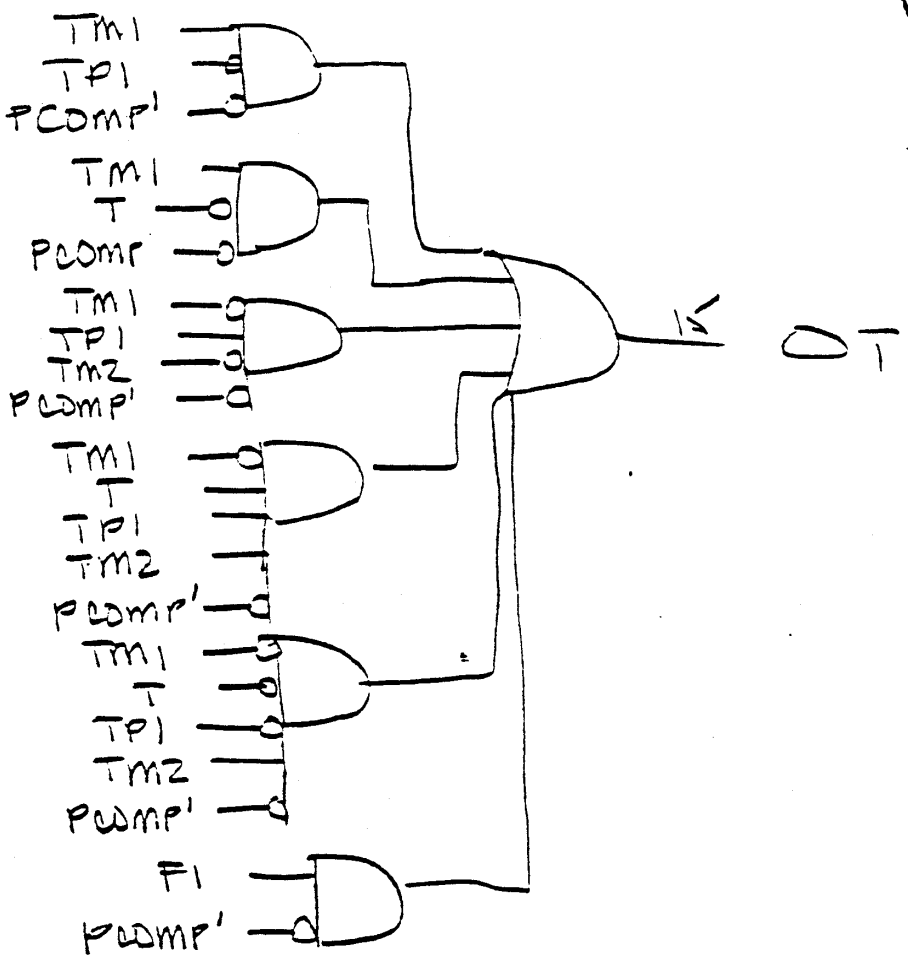
300 1/28/81




```
[illegible] 5M TM1 T TP1 TP2 WTGT' PCOMP' TM2
[illegible]D OE 12 13 LTE OT ELY F2 F1 19 VCC

ELY:= /TM1*/TP1 + T*TP1*/TM2 +
TM1*/T + TP1*TM2 + PCOMP' + F1

LTE:= TM1 + T*/TP1 + /TM1*/T*/TM2 +
/TM1*/T*TP1*TM2 + PCOMP' + F1

OT:= TM1*/TP1*/PCOMP' +
TM1*/T*/PCOMP' +
/TM1*TP1*/TM2*/PCOMP' +
/TM1*T*TP1*TM2*/PCOMP' +
/TM1*/T*/TP1*TM2*/PCOMP' +
F1*/PCOMP

F1:= F1 + /T*/5M + TM1*5M +
 *5M + WTGT'

F2:= /5M + F1 + TM1*5M +
T*5M + WTGT'


DESCRIPTION:

THIS PAL PERFORMS NRZ TO MFM ENCODING
[illegible]US THE EARLY, ON TIME, AND LATE
SIGNALS FOR WRITE PRECOMPENSATION.
NOTE THAT ALL OUTPUTS ARE ACTIVE HIGH,
SO THE ZEROS OF THE FUNCTIONS ARE
IMPLEMENTED IN THE EQUATIONS.
```



CT- 302



303

CONTROLLER CARD

The Profile controller card consists of four main sections; the MPU (Z8), 2K byte RAM, serializing/deserializing and error detectin logic, and the read/write control logic.

Functionally, the controller communicates with the Apple ///, provides signals to read and write serial data on the disk, moves the heads to the proper track, and monitors error conditions.

The Z8 MPU provides an intelligent interface to the Apple. High level commands (e.g. read, write, status and associated data) are communicated the the MPU via the RAM, then interpreted and executed. the results of the operation (completion status bytes, disk data) are passed back to the Apple via the RAM. Access to the RAM is controlled by the MPU as a function of its current state and the CMD input from the Apple interface card. The MPU responds to the Apple with the BSY line. The CMD/BSY handshake establishes directin of data transfer between the controller and the Apple, and determines when the Apple begins and ends its data transfer. Once the data transfer has been completed, the MPU regains control of the memory and either executes the command or returns to an idle condition if a command has just been completed.

CONTROLLER/DISK INTERFACE

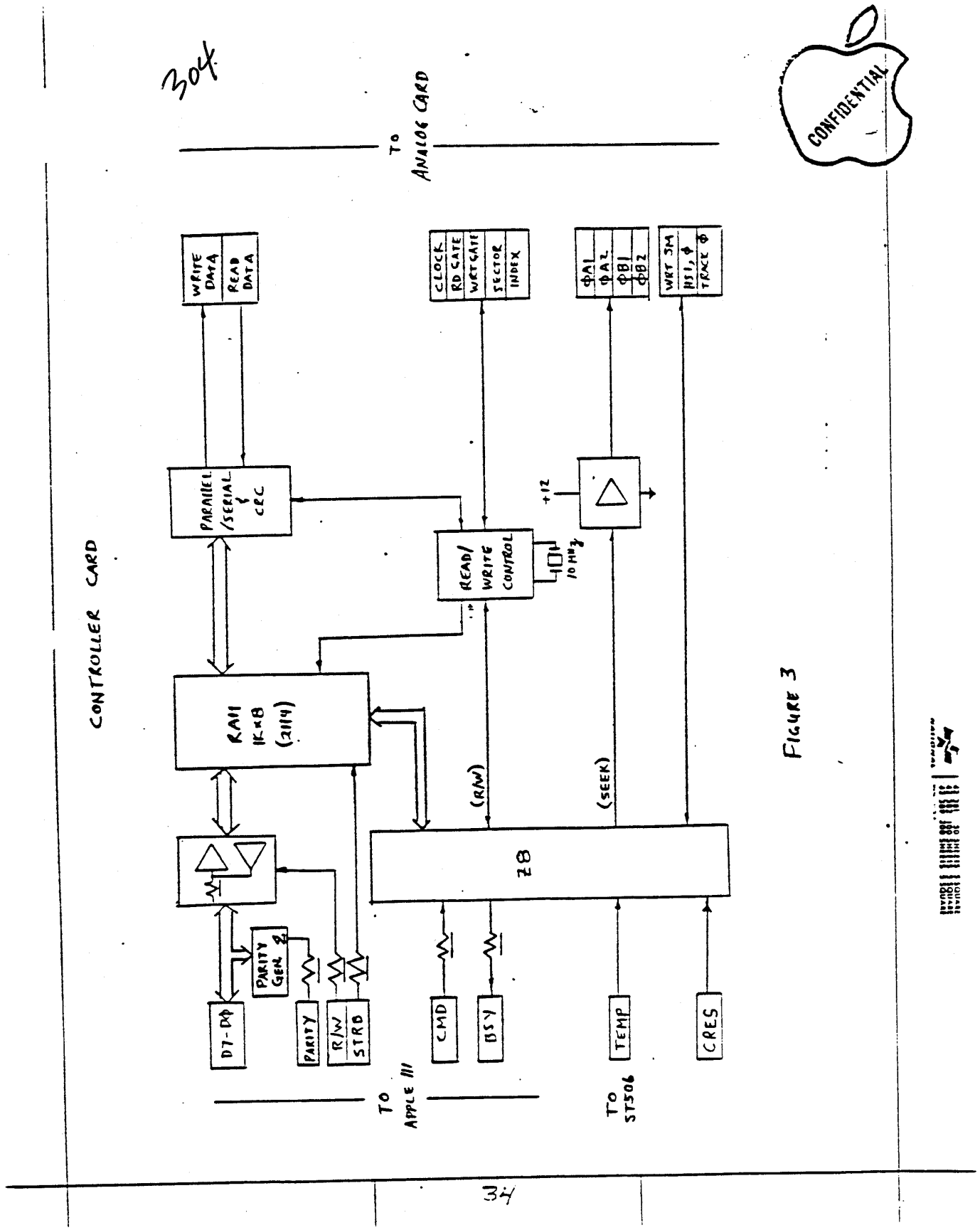
The controller card outputs signals to the analog card over a 25-pin conductor flat cable. The function of the analog card is to translate between the digital data stream at the controller and the analog signals compatible with the heads and disk media.

Signals at the interface are as follows:

Write Data	(output)	Serial Data to be written on the disk.
Read Data	(input)	Serial Data recovered from the disk.
Clock	(i & o)	Defines when Write and Read Data are valid.
Read Gate	(output)	Enables read circuitry on analog card.
Write Gate	(output)	Enables write circuitry on analog card.
Sector	(input)	Defines sector boundaries.
Index	(input)	A pulse which occurs once per disk revolution.
Phaes A1,2 B1,2	(output)	Activates stepper motor coils to position heads.
Write Sector Mark	(output)	Writes DC sector boundaries.
Head Select 1,0	(output)	Causes selection of one of four read/write heads.
Track 0	(input)	Indicates that the heads are positioned over the outermost tracks.

READ/WRITE SEEK

The MPU supervises the remaining hardware on the controller via signals on its I/O ports and through control information stored in RAM. The MPU directly controls the stepper motor phases to move the actuator and heads from track to



track (seek control), selects one of four read/write heads, and writes sector marks on the disk during the virgin formatting operation (at the factory).

Read/Write functions are performed by the Read/Wrtie Control Logic on command from the MPU. This logic in turn controls the parallel to serial conversion when writing, and the serial to parallel conversion when reading; data transfer to and from RAM; and the CRC (Cyclic Redundancy Check); and outputs an error signal to the MPU. If an error occurs, the MPU will automatically perform a data recovery procedure if the number of CRC errors on a given sector exceeds a predetermined level. If the error rate persists, the sector will be spared (relocated to an other area of the disk).

OTHER CONTROLLER FUNCTIONS

To prevent heat buildup in the drive, the MPU removes power from the stepper motor if no commands have been received in .75 seconds. After 3 seconds it will move the head to a non data area and shut down again.

The BSY signal controls the activity LED which will turn on whenever the controller is idle (not busy).

306

CONTROLLER DETAILED BLOCK DIAGRAMS

The first thing we need to do is to get familiar with where the data goes in each one of the various modes. Once we understand where it is supposed to go it is a lot easier to understand how the control logic gets the job done. By not bothering with the super detail at this point, or using the assumption that it does it as if it were a mystery of life. Trust us!

CONTROLLER/INTERFACE COMMAND HANDSHAKE

The MPU having done its previous command is sitting idle and is waiting for the Apple to tell it to do something. The interface raises CMD and the MPU starts its thing.

Upon seeing CMD going high the MPU raises its BSY line and places 01 on the interface bus. The apple must ack the Profiles response with a 55 when it lowers the CMD line or the profile will abort the operation and go back to idle. When the interface lowers CMD (and has acked the response with 55 the MPU conditions the bus to receive the command bytes.

The command bytes are not read by the MPU at this time but are stored in the RAM. When CMD goes high again the MPU interprets the command bytes and responds with the result of its command interpretation (For example: if the apple had said to read a block, the MPU will respond with "02" which means "I'm going to read a block"). The apple must confirm the response with a 55 on the bus again, if it disagrees or has changed its mind any other byte will cause the Profile to abort the operation.

It takes two command handshakes to complete a Read operation, and three handshakes for both a write, and a write/verify operatin.

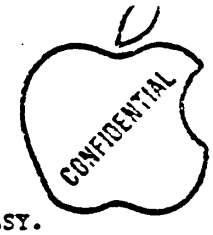
Let's now look at each operation's handshake routine:

Read Operation

1. The Apple raises CMD.
2. Profile places 01 on the bus and signals the apple by raising BSY.
3. The Apple places 55 on the bus and signals Profile by lowering CMD.
4. The apple then transfers the read command bytes to profile ram.
5. The apple the raises CMD again.
6 Profile looks at the command bytes and reponds with 02 and raises BSY.
7. Apple checks response, acks with 55 and lowers CMD.
8. Profile then goes and reads the desired block, keeping BSY high.
9. When finished reading block from disk to RAM, Profile lowers BSY.
10. Apple then transfers data from Profile Ram to its own.
11. Operation complete.

Write Operation

1. The Apple raises CMD.
2. Profile places 01 on the bus and signals the apple by raising BSY.
3. The Apple places 55 on the bus and signals Profile by lowering CMD.



4. The apple then transfers the write command bytes to profile ram.
5. The apple the raises CMD again.
6 Profile looks at the command bytes and reponds with 03 and raises BSY.
7. Apple checks response, acks with 55 and lowers CMD.
8. Apple then transfers block to Profile RAM
9. When finished transferring block to RAM Apple raises CMD.
10.Profile places 06 on bus and raises BSY.
11.Apple checks response puts 55 on bus and lowers CMD.
12.Profile writes data on disk and updates status bytes in RAM.
13 When done writing Profile lowers BSY.
14.Apple transfers status bytes from Profile RAM and sees if OK.
15.Operation complete.

Write/Verify Operation

1. The Apple raises CMD.
2. Profile places 01 on the bus and signals the apple by raising BSY.
3. The Apple places 55 on the bus and signals Profile by lowering CMD.
4. The apple then transfers the write/verify command bytes to profile ram.
5. The apple the raises CMD again.
6 Profile looks at the command bytes and reponds with 04 and raises BSY.
7. Apple checks response, acks with 55 and lowers CMD.
8. Apple then transfers block to Profile RAM
9. When finished transferring block to RAM Apple raises CMD.
10.Profile places 06 on bus and raises BSY.
11.Apple checks response puts 55 on bus and lowers CMD.
12.Profile writes and verifies data on disk and updates status bytes in RAM.
13 When done writing Profile lowers BSY.
14.Apple transfers status bytes from Profile RAM and sees if OK.
15.Operation complete.

This simple handshaking protocol, seems a bit cumbersome but it allows very complete control and acknowledgement of every action before it is allowed to go ahead.

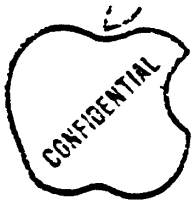
We have gone through system level data movement but what goes on inside Profile? Thought you'd never ask.

The simplest operation is the read.

First one very simple statement, the MPU is used to condition the logic but is not really actively involved with data transfers to/from disk or the Apple, that is done by the Read/Write Control logic. Now let's go.

The Read Operation Command Handshake is complete and the MPU has seen both command responses ack'd. It then conditions the logic to start a read. First it interprets the block number requested and selects the proper head, and alters the stepper phase control lines to match the proper track. It then places the head,track,sector infromation in RAM for comparison with infomatin returning from the disks headers. After proper timeouts, if there was eiter a head and/or track change it starts the Read/Wrtie control Logic (hereafter referred to as RWCL).

306



In the overview in the introduction it was stated that there were three or four main circuit groups in the controller. Well that's true, and they are again:

1. MPU
2. RAM
3. Serializer/Deserializer
4. Read/Write Control

The simplifed block diagram shows the overall relationship of these elements. But it certainly doesn't help understand operation. So Lets get a little more detailed in the block diagram department. The next level of block breaks the controller into 17 elements.

1. Data buffers
2. Data In Mux
3. Deserialized Data Register
4. Stepper Motor Drivers
5. MPU
6. RAM Address Counter/Regeister
7. RAM
8. Programmable Counter Timer
9. Serial/Deserial Shift Register
10. PAL (Programmed Array Logic)
11. BYTC State Machine
12. RWTC State Machine
13. Read Data Mux
14. Write Data Mux
15. CRC Generator,Checker
16. System Clock Selector
17. Bus Parity Checker

A very brief functional description of each of these elements will help give a foundation for later discussions.

DATA BUFFERS

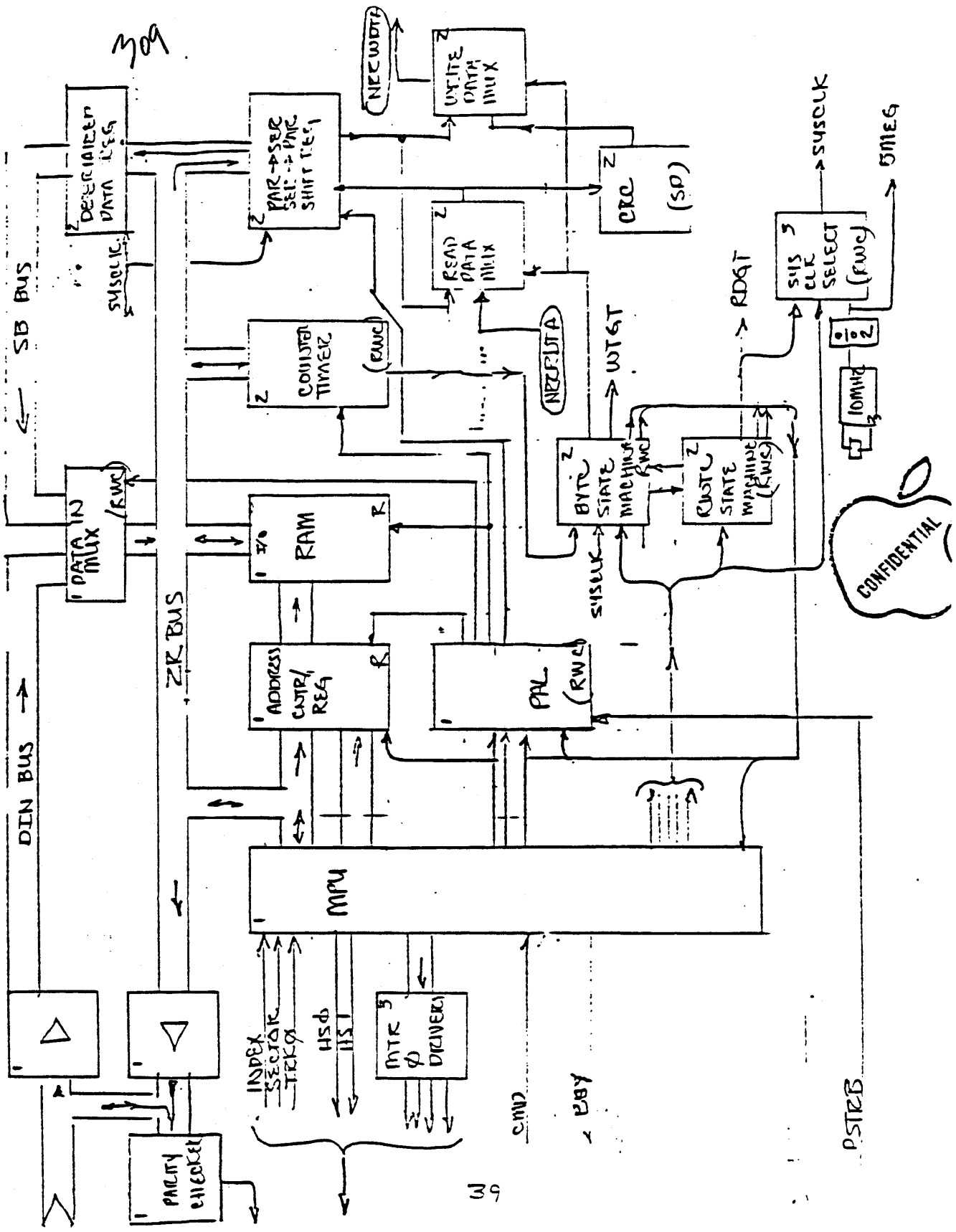
Simple octal buffers which gate data to and from the controller and interface cards.

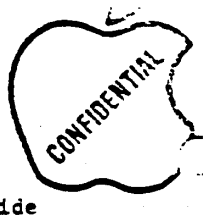
DATA IN MUX

This mux is used to direct data coming in from the interface card or coming "in" from the disk. The usual destination of its outputs is the RAM.

DESERIALIZED DATA REGISTER

This 8 bit register temporarily holds the deserialized data from the disk so that the shift register can recieve the next byte. When the logic is ready it will direct the registers contents to RAM though the DATA IN MUX.





310

STEPPER MOTOR DRIVERS

Using direct outputs of the MPU as inputs the stepper motor drivers provide the level shifting and current source for driving the head positioning stepper motor on the HDA.

MPU

The MPU is a Z8 microprocessor. It has a self contained rom program. It controlls by setting modes, and directly controls the stepper motor, head selction. It provides the intelligence to control the machine and to interface to the host computer.

RAM ADDRESS COUNTER/REGISTER

This element can be preset to a certain point and counted up through a sequential range of address for RAM access. It is also used as a Address register where it is loaded with a value for a specific single access. The method of use depends on what operation is currently going on.

RAM

The RAM is a 2k byte ram array which is used to hold data to and from the host and disk. Various locations are used to hold status information, and the current sparing tables.

PROGRAMMABLE COUNTER/TIMER

This element contains three programmable counters which yeild byte time information to the Read Write Control logic, and basically keeps track of where in the particular sector the current information is.

SERIAL/DESERIAL SHIFT REGISTER

This register is quite busy. It is used to take the parallel data from RAM and shift it out serially to the analog card. It also takes the serial data from disk and shifts into a parrallel format for transfer back into ram.

PAL

This single chip is a logic array specifically programmed for this application. It performs many very complex and/or combintational logic functions. Its primary function is to control the function of the address counter, direction of data to/from ram, loading the counter/timer and controlling the serial/deserial register.

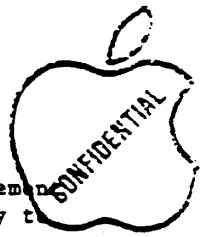
BYTC STATE MACHINE

The BYTC (BYTe Control) State Machine one of the two major elements of the Read Write Control. It steps through the control states for each byte at a bit time rate.

RWTC STATE MACHINE

40

311



The RWTC (Read Write Timing Control) State Machine is the second major element of the Read Write Control. Its steps through the control states necessary to control the timing for all the operations associated with block/sector reading and writing.

READ DATA MUX

The Read Data Mux is used to select one of two sources of data (either NRZ Read Data or Serialized Data from the SER/DESER reg. It is used in both the read and write modes. In reading it gates NRZ Read Data through. During the write operation it gates serialized data through for use by the CRC generator.

WRITE DATA MUX

The WDM is used to select either serialized data from the shift register or the generated CRC characters to the disk. It is really only used in the write operation, though its ouput is flailing away during reads.

CRC GENERATOR/CHECKER

The CRC (cyclic redundancy check) circuit is used to compute crc check characters that are written at the end of each data block on disk during write operations, and used to compute CRC for read data and compare the result with the CRC characters that were read at the end of each data block.

SYSTEM CLOCK SELECTOR

The SSC switches from the crystal oscillator used during idle and write operations, to the read clock generated by the analog card during the read operation. This keeps the logic in "perfect" sync with the data.

BUS PARITY CHECKER

The BPC forms the other half of the bus parity checking circuit on the interface card. It constantly monitors the bus and check the parity at the controller end and sends its sum to the interface to be compared with the sum on the interfac end. There should hardly ever be a parity error unless there is a cable fault.

41

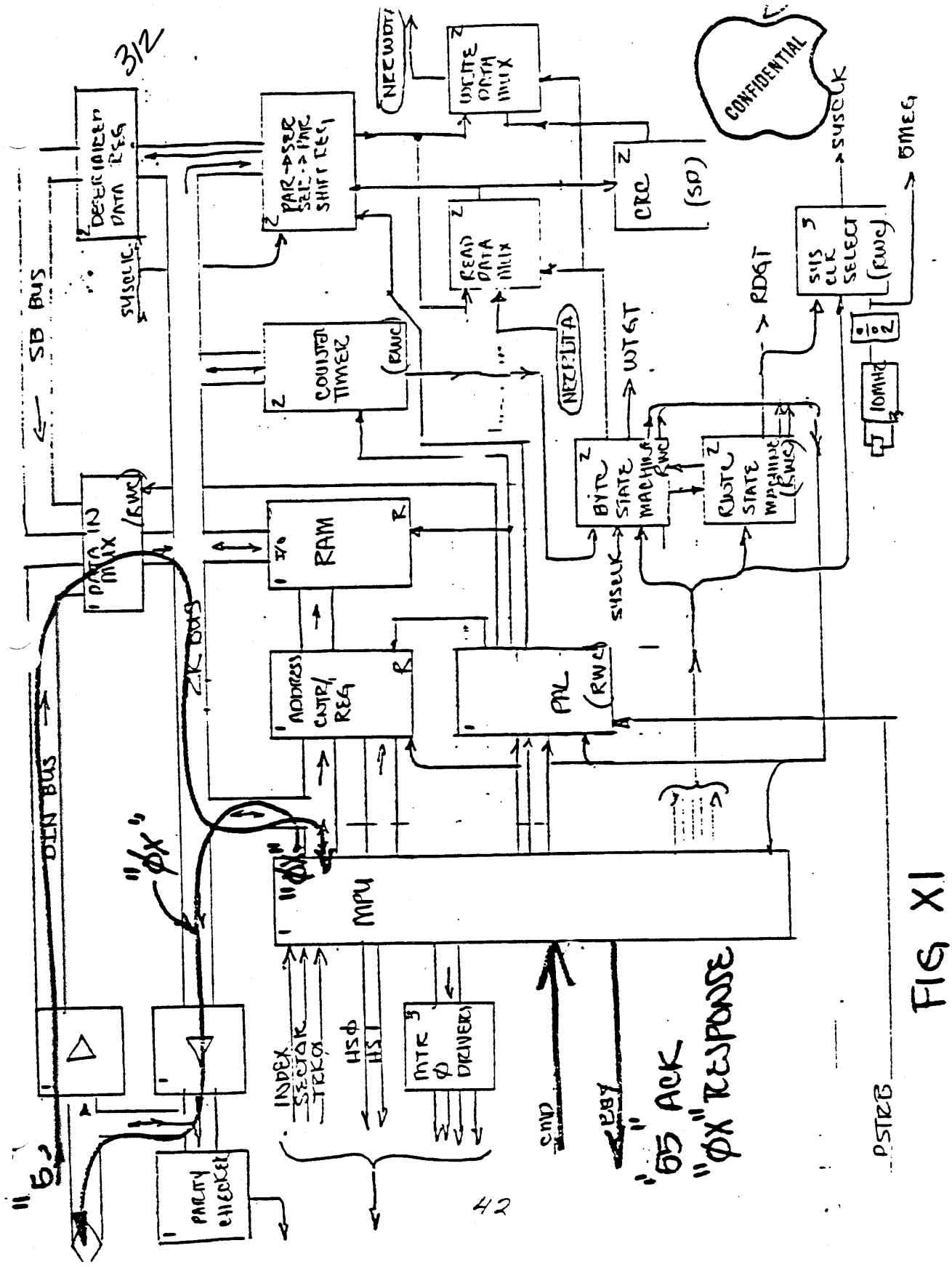
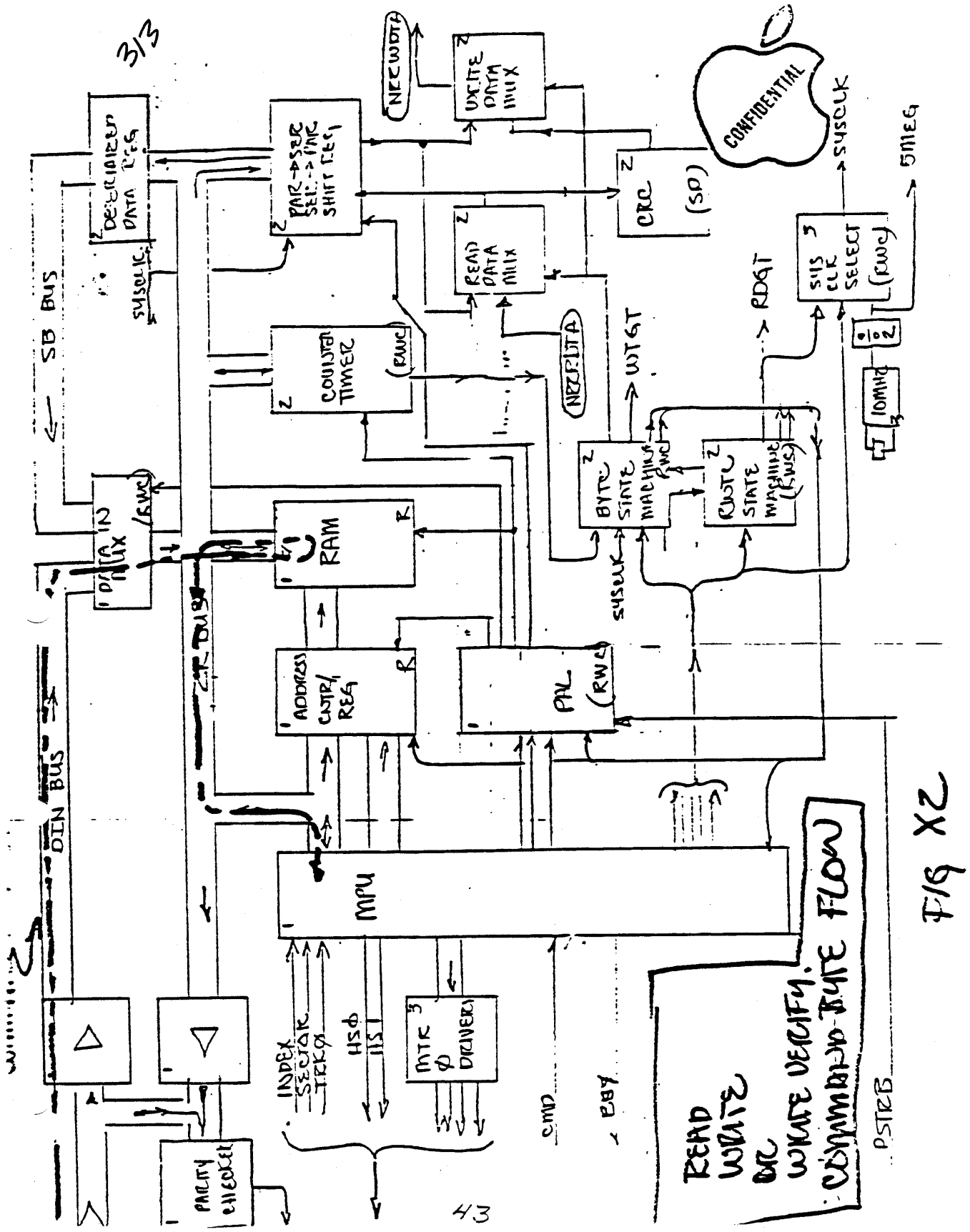
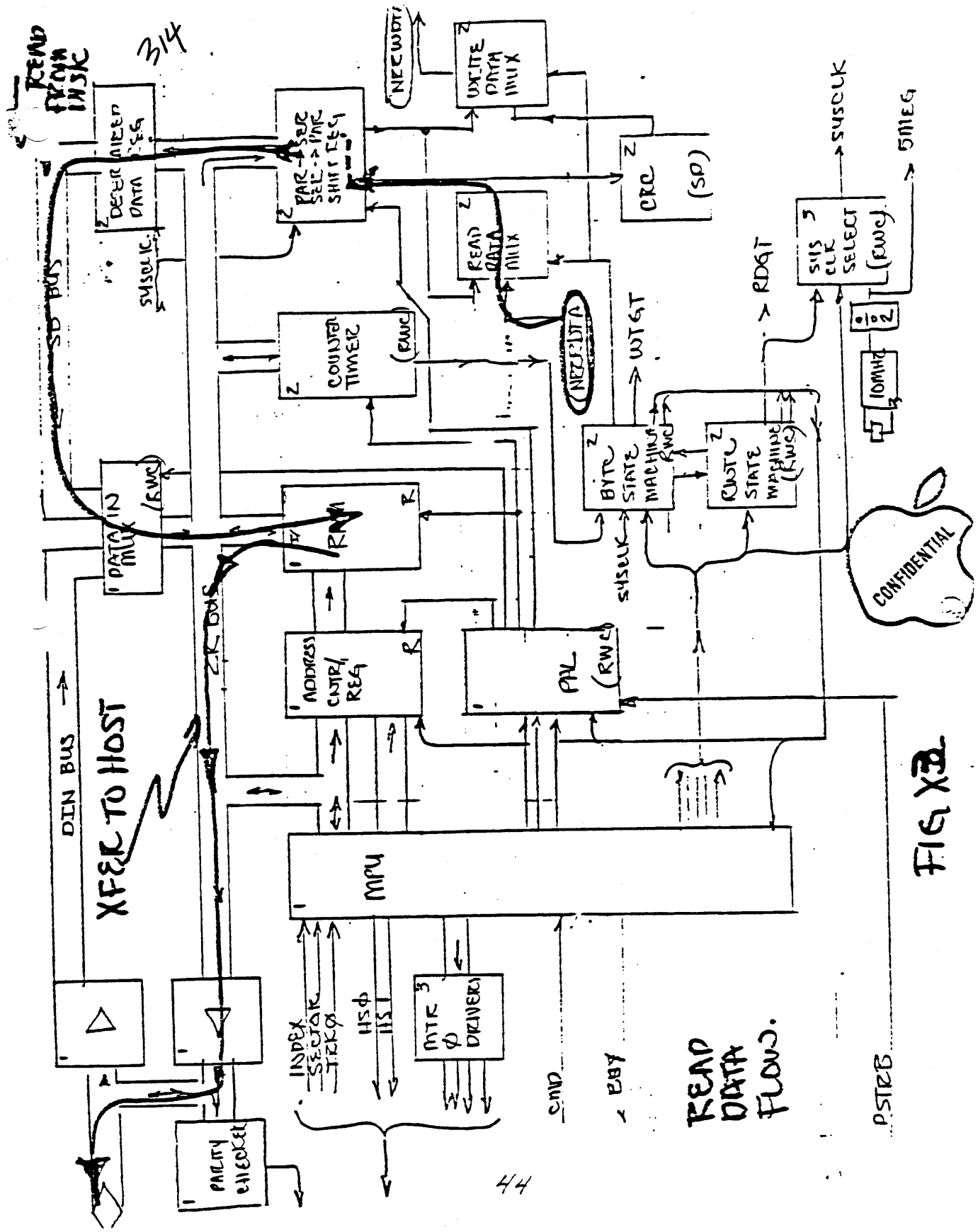
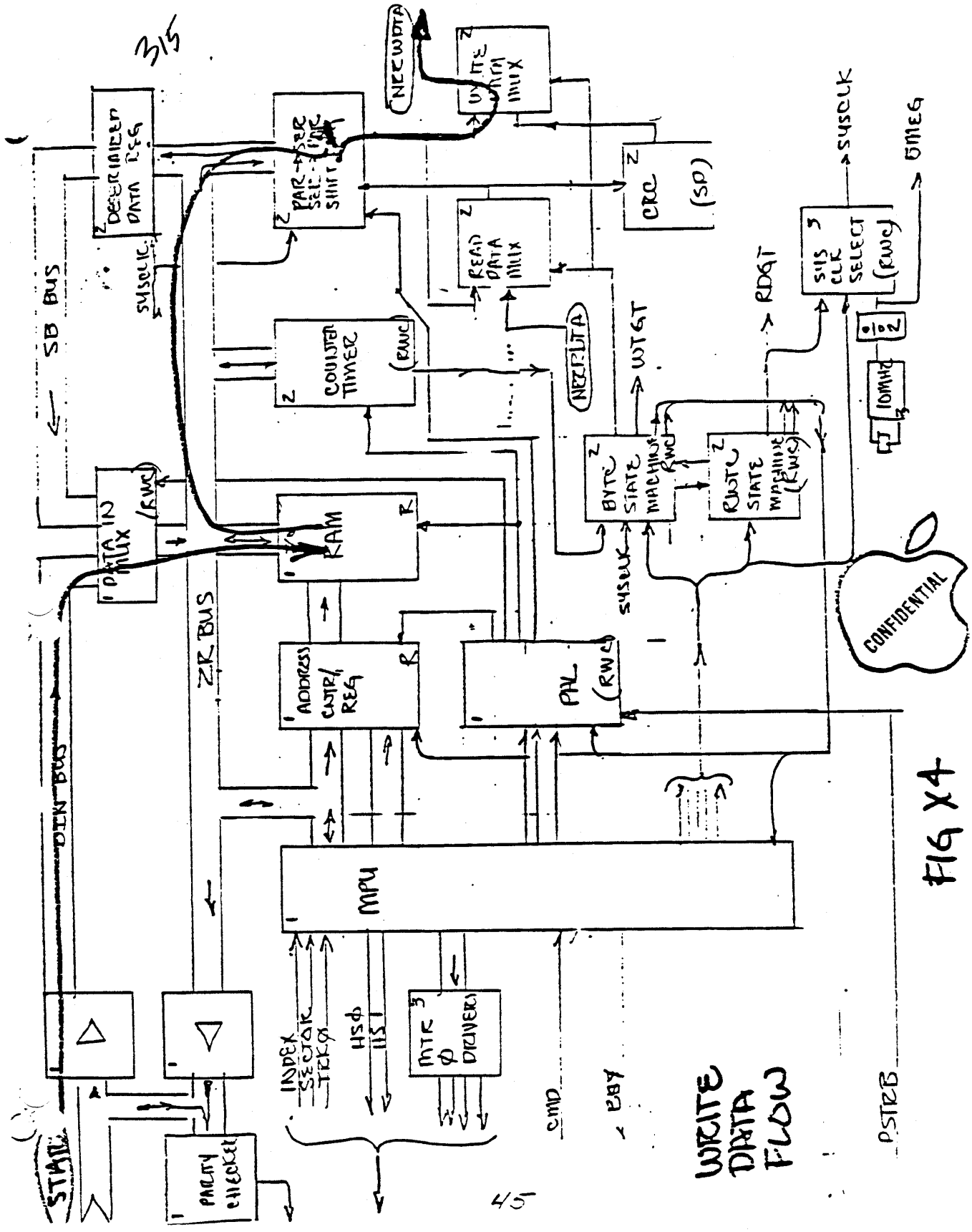


FIG XI

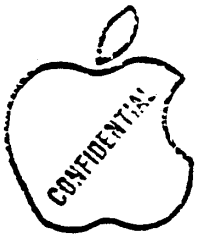


F/G X2





316



COMMAND ACK' AND RESPONSES

Looking at figure X1, we see the MPU's response to the interfaces CMD going high of "0X" coming out of the processor up to the ZR bus and off to the left through the buffer to the interface. When the interface drops CMD the processor will (should) see the apples's "55" acknowledgement coming in through the buffer to the Data In Mux and directly to the processor. This ack will trigger the processor to get the RAM ready to accept the command bytes.

COMMAND BYTES

After the first handshake of CMD and BSY, the Apple transferrs the actual command and delimiting bytes. They come in as does all data from the Apple through the DATA IN MUX, but gets stored in the RAM. There is another CMD handshake, during which the MPU evaluates the command bytes by bringing them in from RAM. The MPU then gives its response byte, which is ack'd or nack'd.

READ DATA FLOW

The control logic conditions the circuitry to accept the NRZRDTA through the READ DATA MUX into the DESERIALIZER where it is converted to parallel bytes. It is then transferred to the DESERIALIZED DATA REGISTER, held momentarily then gated through the DATA IN MUX to the RAM.

When the sector(block) is complete the control logic tells the MPU to lower the BSY line. It does and then the Apple strobes the data out of RAM through the output buffer to the interface card. The RAM address counter has been set to the beginning of the data, and each strobe sent by the Apple increments the counter to the next address.

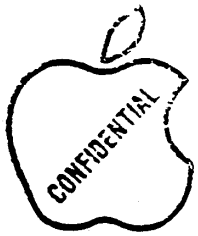
WRITE DATA FLOW

After completing the proper handshake the logic is conditions to accept a block of data from the Apple. The Apple strobes in the data into the RAM and when complete will go through another handshake. This handshake conditions the logic to take the data from the RAM to the SERIALIZER, where it is converted to a serial data stream. Then onto the analog card through the WRITE DATA MUX.

WRITE/VERIFY

The W/V operation is a combined write and read compare. The data comes from the Apple in the same way as it does for a write. When the write operation is complete the MPU goes through a verification of the data written, much like a read operation.

3X



DISK FORMAT

The Profile has two disks, each having two sides. Their is a read/write head for each surface. Each surface is divided into 152 concentric tracks. Each Track has 16 sectors. The user has access to 9,728 blocks (sectors). Each sector has 532 bytes of user data. This means that the formatted drive contains 5.17 million bytes of user data. (the equivilant of over 40 DISK II diskettes)

SECTOR FORMAT

Each sector is formatted into two fields, the header and the data block. There is a preamble of 22 bytetimes of zeros before each field.

The header contains 16 bytes; two start header bytes, three bytes for track, sector, and head ID's, and three bytes for a redundancy of the compliments of the track, sector head bytes. The rest of the field is filled with zeroes.

The data field contains two synce bytes, 532 data bytes, and ***bytes for the CRC charaters. After the CRC charaters the disk writes zeros until WTGT is lowered.

CHECK HEADER OPERATION

For each read or write operation, the specific sector must be located and checked. This is accomplished by the action of the MPU, RWCL and most all of the logic on the controller. It however is a relatively simple operation.

The MPU sets a complete replica of the desired header (exclusive of the first "01" start byte) into a specific area of RAM. Then it waits for the sector pulse. When it sees the sector pulse it starts the State Machines. The state machines in combination with the PAL move each successive byte of the header replica into the SERIALIZER which is then serialized and shifted out in sync with the incoming NRZRDTA. It is compared and if there is even a single bit difference in comparison the STATE MACHINES abort the attempt and reset to wait for the next incoming sector then the process is repeated. This will go on until the header matches the image in RAM or there is a timeout error (inside the MPU...its waiting for the "sector done" from the state machine if it doesn't see it in a reasonable amount of time the MPU takes over and goes through an Error routine.)

If the desired operation was to read a block, the logic then accepts the data in from disk and moves it into RAM. If the operation was to write a block the logic is conditioned to move the data from RAM to the disk.

318

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

319

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 FFFFFE will read or write Profile's RAM buffer, while a block number of FFFFFF 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

320

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

32

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

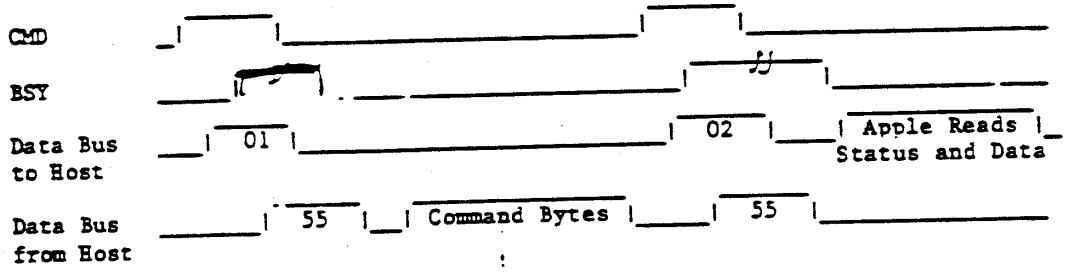
7 - 0 = the number of errors encountered when rereading a block after any read error

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

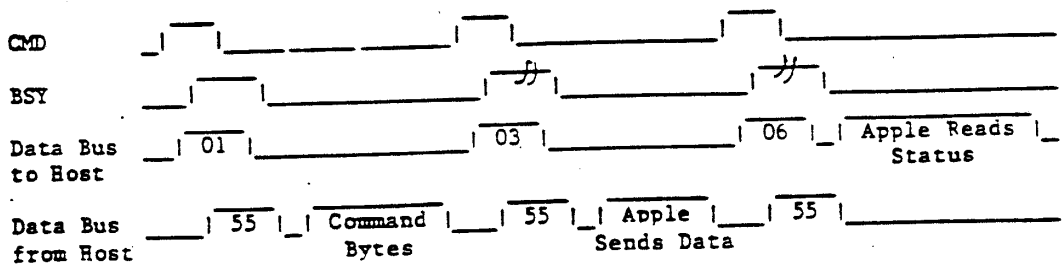
As mentioned previously, reading block FFFFFF gets Profile's spare table. The first 13 bytes are allocated for the device name, which is PROFILE followed by 6 blanks. The next 3 bytes are allocated for the device number, which is 00 00 00. The next 2 bytes are used for the program revision number, which 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



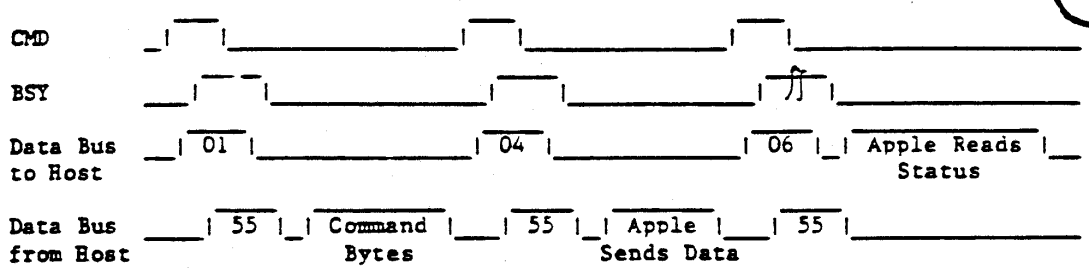
Write Operation



32M

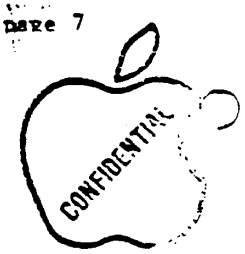
PROFILE COMMUNICATION PROTOCOL

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. After BSY goes high, CMD can be lowered as long as anything but 55 (hexadecimal) is on the data bus.

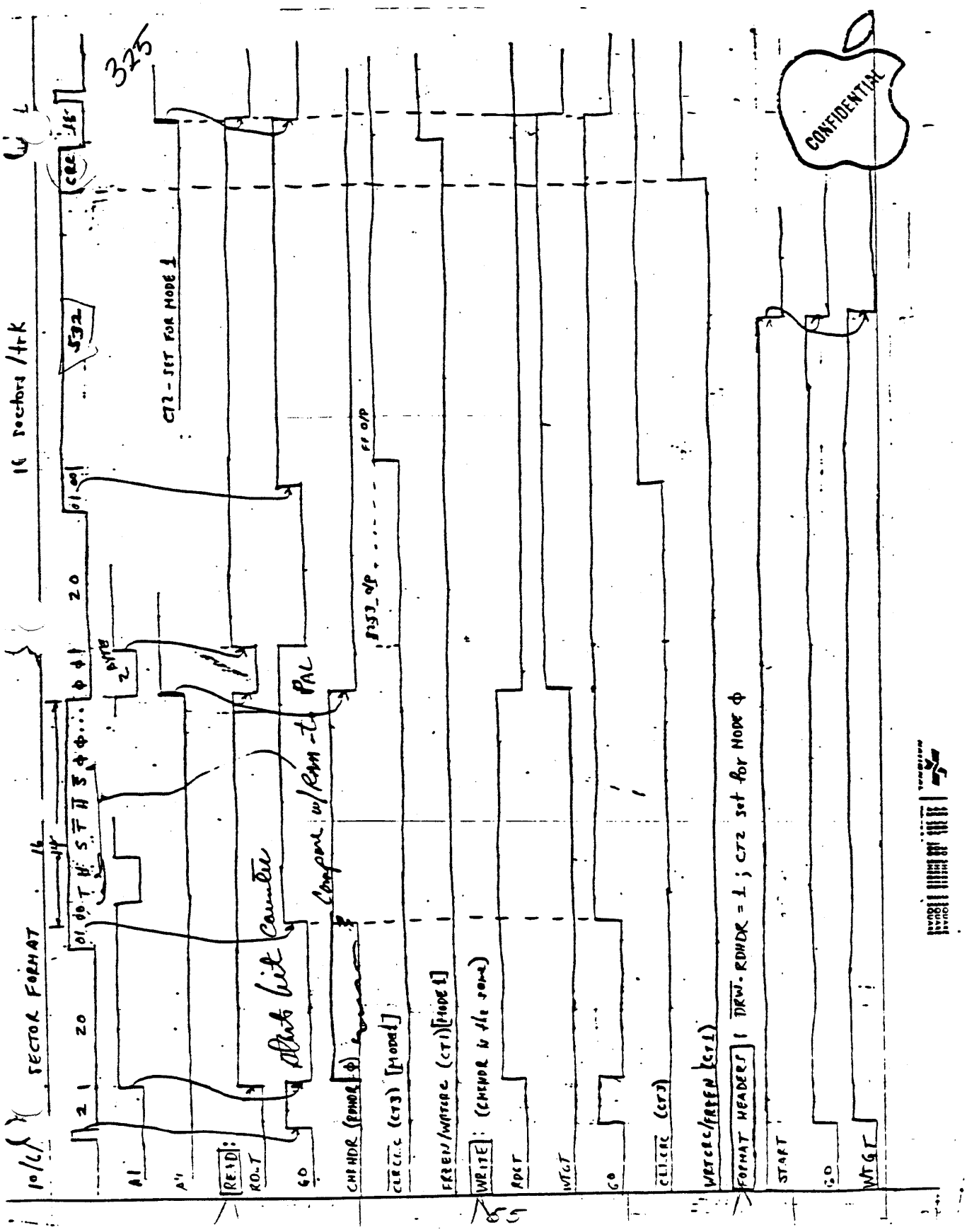
324

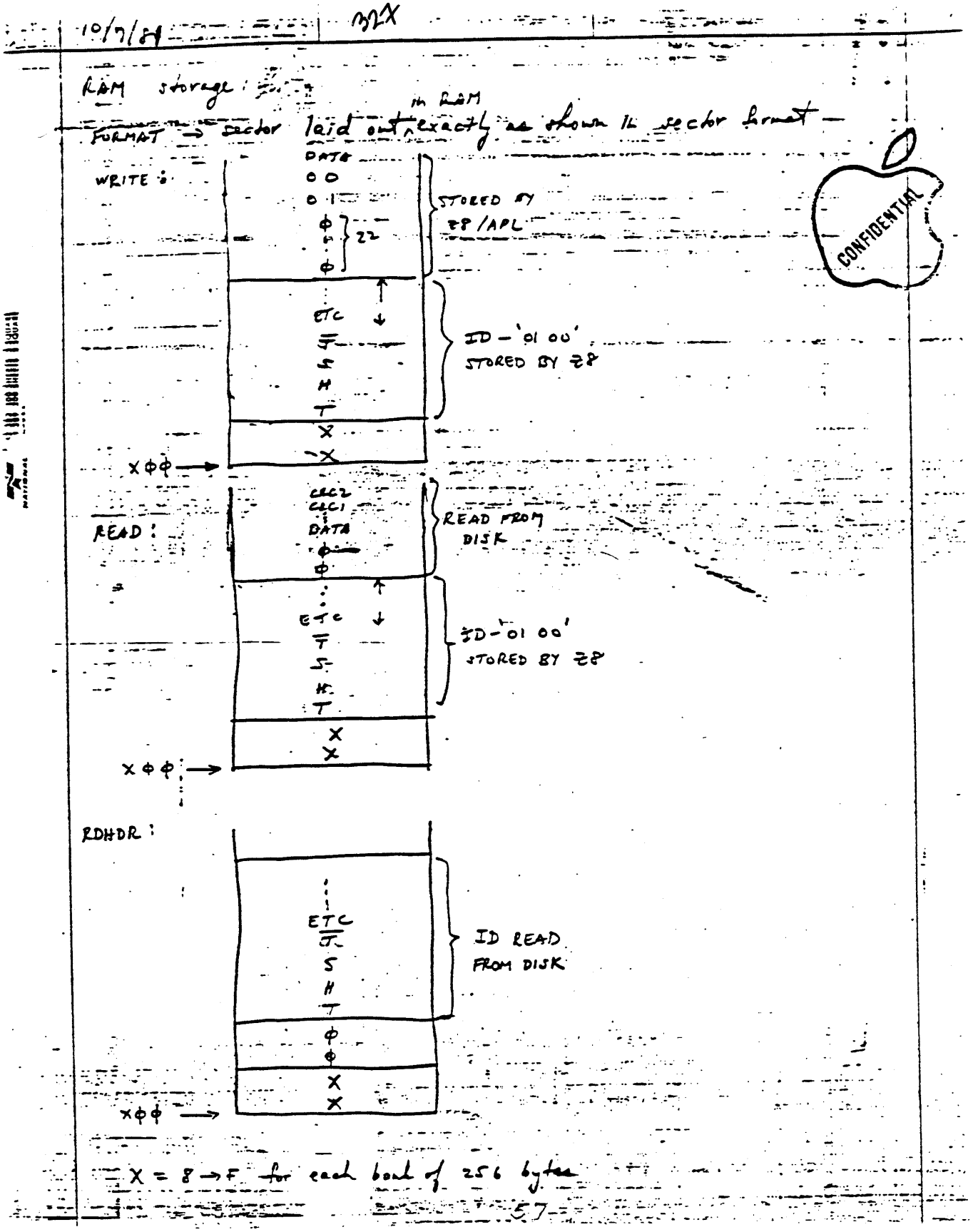


Addendum for Controller Versions 3.96 and 3.97

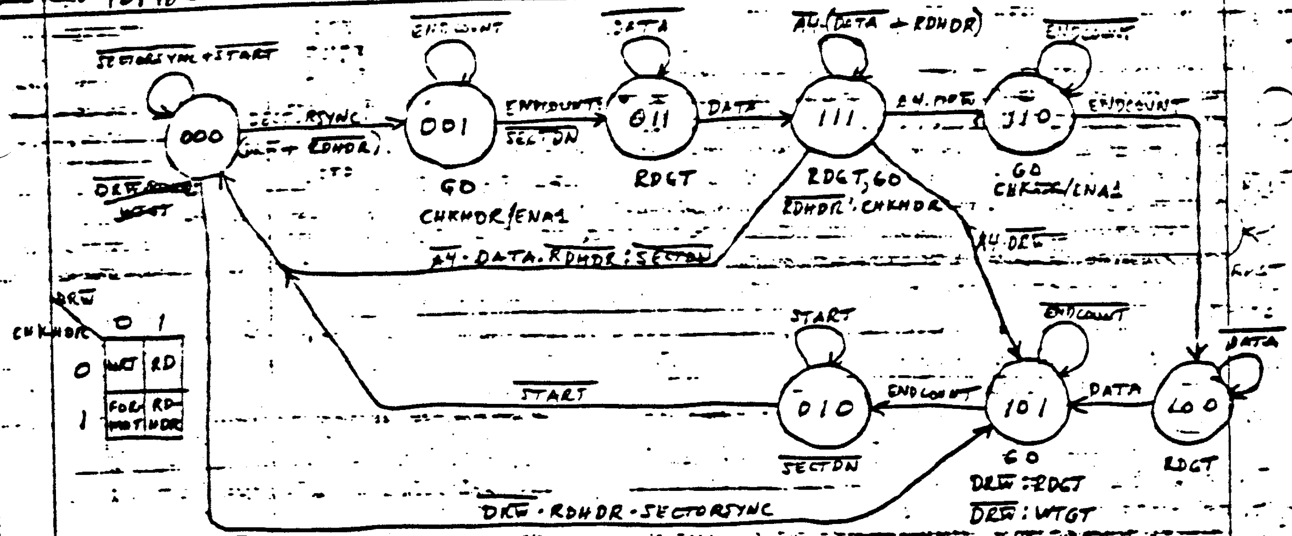
Revision 3.96 of the Profile controller program has several improvements over revisions 3.90 and 3.92 (3.90 and 3.92 are identical except that 3.92 moves the stepper at half the 1.5ms per track rate used by 3.90). Instead of the Z8 falling directly into the write/verify/sparing routine if the number 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. Revision 3.97 waits 3.28 seconds instead of 1.5 seconds before moving the head off the data area of the disk.

WILL BE FIRST MARKED VERSION





10/27/80 328 STATE MACHINE



EQUATIONS: W71B [WIG11B.TEXT]

$D\phi = Y\phi = \overline{Y2}\cdot\overline{Y1}\cdot\overline{Y\phi}\cdot SECTORSYNC + \overline{Y2}\cdot Y\phi + Y2\cdot Y1\cdot Y\phi\cdot\overline{A4}\cdot\overline{DATA}$
$+ Y2\cdot Y1\cdot Y\phi\cdot\overline{A4}\cdot RDHDR + Y2\cdot Y1\cdot Y\phi\cdot A4\cdot\overline{DRW} + Y2\cdot\overline{Y1}\cdot\overline{Y\phi}\cdot DATA$
$+ Y2\cdot\overline{Y1}\cdot Y\phi\cdot\overline{ENDCOUNT}$

$Y\phi = \bar c\cdot\bar b\cdot\bar a\cdot i + \bar c\cdot a + c\cdot b\cdot a\cdot\bar e\cdot\bar g + c\cdot b\cdot a\cdot\bar e\cdot d$
$+ c\cdot b\cdot a\cdot e\cdot\bar f + c\cdot\bar b\cdot\bar a\cdot g + c\cdot\bar b\cdot a\cdot\bar h$

INPUTS

a = $Y\phi$
b = Y1
c = Y2
d = RDHDR
e = A4
f = $\overline{DRW}$
g = DATA
h = ENDCOUNT
i = SECTORSYNC

CONFIDENTIAL

$D1 = Y1 = \overline{Y1}\cdot Y\phi\cdot ENDCOUNT + \overline{Y2}\cdot Y1 + Y2\cdot Y1\cdot Y\phi\cdot\overline{A4}\cdot\overline{DATA}$
$+ Y2\cdot Y1\cdot Y\phi\cdot RDHDR + Y2\cdot Y1\cdot Y\phi\cdot A4\cdot\overline{DRW} + Y2\cdot Y1\cdot\overline{Y\phi}\cdot\overline{ENDCOUNT}$

$Y1 = \bar b\cdot a\cdot h + \bar c\cdot b + c\cdot b\cdot a\cdot\bar e\cdot\bar g + c\cdot b\cdot a\cdot d + c\cdot b\cdot a\cdot e\cdot f$
$+ c\cdot b\cdot\bar a\cdot\bar h$

$D2 = Y2 = \overline{Y2}\cdot\overline{Y1}\cdot\overline{Y\phi}\cdot\overline{DRW}\cdot RDHDR\cdot SECTORSYNC + \overline{Y2}\cdot Y1\cdot Y\phi\cdot DATA$
$+ Y2\cdot Y1\cdot Y\phi\cdot\overline{A4}\cdot\overline{DATA} + Y2\cdot Y1\cdot Y\phi\cdot\overline{A4}\cdot RDHDR + Y2\cdot Y1\cdot Y\phi\cdot A4$
$+ Y2\cdot\overline{Y\phi} + Y2\cdot\overline{Y1}\cdot Y\phi\cdot\overline{ENDCOUNT}$

$Y2 = \bar c\cdot\bar b\cdot\bar a\cdot\bar f\cdot d\cdot i + \bar c\cdot b\cdot a\cdot g + c\cdot b\cdot a\cdot\bar e\cdot\bar g + c\cdot b\cdot a\cdot\bar e\cdot d + c\cdot b\cdot a\cdot e$
$+ c\cdot\bar a + c\cdot\bar b\cdot a\cdot\bar h$

$D3 = RDGTA = Y1\cdot Y\phi + Y2\cdot\overline{Y1}\cdot\overline{Y\phi} + Y2\cdot\overline{Y1}\cdot Y\phi\cdot\overline{DRW} = b\cdot a + c\cdot\bar b\cdot\bar a + c\cdot\bar b\cdot a\cdot f$

$D4 = CHKHDR/ENA1(A) = \overline{Y2}\cdot\overline{Y1}\cdot Y\phi + Y2\cdot Y1\cdot Y\phi\cdot\overline{RDHDR} + Y2\cdot Y1\cdot\overline{Y\phi}$
$= \bar c\cdot\bar b\cdot a + c\cdot b\cdot a\cdot\bar d + c\cdot b\cdot\bar a$

$D5 = \overline{SECTDNA}$ = {$\overline{Y2}\cdot\overline{Y1}\cdot Y\phi\cdot\overline{ENDCOUNT}$ ~~Y2·Y1·Yφ·A4·DATA·RDHDR~~ + $\overline{Y2}\cdot Y1\cdot\overline{Y\phi}$} = {$\bar c\cdot\bar b\cdot a\cdot h$ ~~c·b·a·e·g·d~~ + $\bar c\cdot b\cdot\bar a$}

$D6 = WTGTA$ = ~~Y2·Y1·Y0·DRW·RDHDR~~ + $Y2\cdot\overline{Y1}\cdot Y\phi\cdot\overline{DRW}$ = ~~c·b·a·f·d~~ + $c\cdot\bar b\cdot a\cdot f$

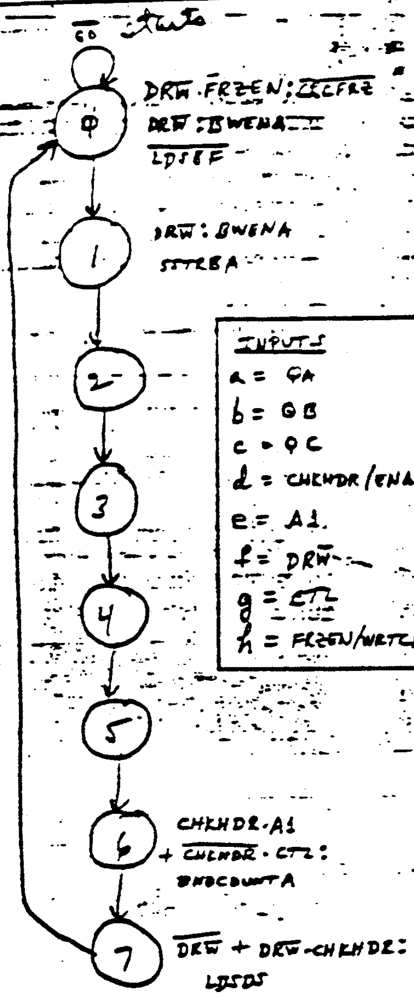
D7 = GO = ~~Y1·Yφ + Y2·Y1 → b·a + c·b~~
$= \overline{Y1}\cdot Y\phi\cdot\overline{ENDCOUNT} + Y2\cdot Y1\cdot Y\phi\cdot ENDCOUNT + Y2\cdot Y1\cdot Y\phi = \bar b\cdot a\cdot\bar h + c\cdot b\cdot\bar a\cdot\bar h + c\cdot b\cdot a$

58

10/28/80 329

U22 U29 U30 STATE MACHINE 2

W13B [W13B.TEXT]



INPUTS

- a = QA
- b = QB
- c = QC
- d = CHKHDR/ENA1
- e = A1
- f = $\overline{DRW}$
- g = CT2
- h = FRZEN/WRTCRC

$D\phi = ENDCOUNTA = c\cdot b\cdot\bar{a}\cdot CHKHDR\cdot A1 + c\cdot b\cdot\bar{a}\cdot\overline{CHKHDR}\cdot CT2$
$= c\cdot b\cdot\bar{a}\cdot d\cdot e + c\cdot b\cdot\bar{a}\cdot\bar{d}\cdot g$

$D1 = BWENA = \bar{c}\cdot\bar{b}\cdot DR\bar{W} = \bar{c}\cdot\bar{b}\cdot f$

$D2 = SSTRBA = \bar{c}\cdot\bar{b}\cdot a$

$D3 = \overline{CRCFRZ} = (\bar{c}\cdot\bar{b}\cdot\bar{a}\cdot DR\bar{W}\cdot FRZEN)' = (\bar{c}\cdot\bar{b}\cdot\bar{a}\cdot f\cdot h)'$

$D4 = \overline{LDSBF} = (\bar{c}\cdot\bar{b}\cdot\bar{a})'$

$D5 = LDSDS = c\cdot b\cdot a\cdot\overline{DR\bar{W}} + c\cdot b\cdot a\cdot DR\bar{W}\cdot CHKHDR$
$= c\cdot b\cdot a\cdot\bar{f} + c\cdot b\cdot a\cdot f\cdot d$

$D6 = \bar{c}\cdot\bar{b}\cdot a$ = state 1

$D7 = c\cdot b\cdot\bar{a}$ = state 6

CONFIDENTIAL

59

PAL16L8 330

1/28/81

CONFIDENTIAL

RAMPAL4 Same as 'RAMPAL'

```
AS DS ZRW DRW MSEL1 MSEL0 SSTRB PSTRB
BWEN GND CHKHDR CTRCS SDSOE MUXEN
RAMOUTEN RAMWRITE RAMSEL ADLOAD ADCLK
VCC

ADCLK = MSEL0*/AS +
MSEL1*/MSEL0*DRW*/CHKHDR*/RAMWRITE +
MSEL1*/MSEL0*DRW*CHKHDR*SSTRB +
MSEL1*/MSEL0*/DRW*SSTRB +
/MSEL1*/MSEL0*/DRW*/RAMWRITE +
/MSEL1*/MSEL0*DRW*PSTRB + /ZRW*/AS

ADLOAD = MSEL0 + /ZRW

RAMSEL = MSEL1*MSEL0*/DS +
MSEL1*/MSEL0*DRW*SSTRB +
MSEL1*/MSEL0*/DRW*ZRW +
MSEL1*/MSEL0*CHKHDR +
/MSEL1*/MSEL0*/DRW*PSTRB +
/MSEL1*/MSEL0*DRW*ZRW

RAMWRITE = MSEL1*MSEL0*/ZRW +
MSEL1*/MSEL0*DRW*/CHKHDR*SSTRB +
/MSEL1*/MSEL0*/DRW*PSTRB +
/MSEL1*MSEL0*/ZRW*/DS

RAMOUTEN = MSEL1*MSEL0*ZRW +
MSEL1*/MSEL0*DRW*CHKHDR +
MSEL1*/MSEL0*/DRW + /MSEL1*/MSEL0*DRW
+ /MSEL1*MSEL0*ZRW*/DS

MUXEN =
MSEL1*/MSEL0*DRW*BWEN*/CHKHDR*ZRW
+ /MSEL1*/MSEL0*/DRW*ZRW

SDSOE = MSEL1*/MSEL0*DRW*/DWEN*/CHKHDR

CTRCS = /MSEL1*MSEL0


DESCRIPTION:

THIS PAL CONTROLS ACCESS TO THE RAM
BUFFER MEMORY FOR THE LS240-LS257
IMPLEMENTATION.
```

WIG BROM.TEXT

WIGSROM.TEXT

60

331

10/13/84

This page not too clear

Val

CONFIDENTIAL

E
CL
RES
$\overline{AS}$
$\overline{DS}$

2MHz ÷ 2

line OR $\overline{CRES}$

latch data from Port 1

LS161 counters (RAM address counters)

& valid time during write or read to

counter.

ZRW : Z8 read/write line. Defines external memory or I/O operation as a read or write.

NOTE: $\overline{AS}$, $\overline{DS}$, & ZRW are tri-stated when Port 1 is placed in the high impedance state.

PORT Φ:

PΦ0-PΦ2 (A8-A10): High order address bits, A8-A10, for RAM accesses. They are latched into the LS161 address counter during $\overline{AS}$.

PΦ3 (Pcomp): Precomp / Reduce Write Current command to disk with electronics (State is 'don't care' if not writing). ~~Also used as 'Transmit Enable' for serial data transmission when latched into the high order bit of the LS161 address counter. (Not currently used)~~

PΦ4-PΦ7 (Φ1-Φ4): ΦB2, ΦB1, ΦA2, and ΦA1 commands, respectively, to the stepper motor drive circuitry.

Port 1 (Z87-Z80):

a. Low order address bits and data, multiplexed by $\overline{AS}$ and $\overline{DS}$, for Z8 RAM accesses.

b. Data bits for loading the 8253 counters (LDCTR, derived from $\overline{DS}$ when MSEL1 is low) is the write input to the 8253.

PORT 1 (cont) — 3 3 2

c. Data to/from the Apple. ZP program must set Port 0 to the high impedance state.

d. Data to/from the disk. ZP program sets Port 1 to hi-z.

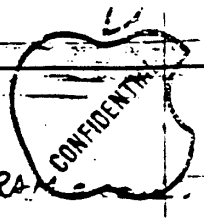
PORT 2: Bits programmed as inputs or outputs via ZP mode register. All bits set to hi-z (input mode) after a reset.

P36 P28 WRITE SECTOR MARK: output. This signal, in conjunction with Format-Enable (FMTEN), which causes WTGT to be true, writes sector marks on the disk. This should only be done during the format operation.

P21 TRK0: Track 0 (TRK0) input. Indicates that R/W heads are positioned over outermost track.

P22 CMD: Command (CMD) input. This line is set true by the Apple via the interface card, and indicates that the Apple is requesting access to the controller RAM. The corresponding handshake line, BSY/INT, is a ZP output described below.

P23 BSY/INT: Busy/Interrupt (BSY/INT). This signal acknowledges the CMD input, via the CMD+BSY protocol described elsewhere. BSY may also be used to interrupt the Apple if enabled on the interface card. (This is not implemented on the first release of the SOR Driver.)



PORT 2 (cont) 333

P24: (MSEL0) ~~The low order select bit to control RAM~~ ~~Also used as the low order address bit during the~~ ~~and to registers~~

P25: (MSEL1 / FJEL): High order RAM accen select bit. Also controls data input to RAM from the Apple or the disk; hi = disk, low = Apple.

P26: Start / Reset Error (START/RSTERR). Enables read/write control hardware to begin a read or write operation at the next sector pulse. Also resets the ECC error flip-flop and the sector timing register.

P27: Disk Read Write / ~~counter~~ (DRW ~~test~~). Controls the read/write input to the RAM for Apple or disk accesses. Command input to read/write hardware. ~~High order address bit, input to the counter registers.~~

PORT 3: P33 - P36 are inputs, and/or interrupts; P37 - P34 are outputs.

P30: ~~Track 0 (TRK0). Indicates that the read/write heads are positioned over the outermost track.~~ SERIAL IN. Serial input port. Not currently used.

P31: Sector. Pulses resulting from asserting the 'write Sector Mark' line during formatting. Note that P31 is also input to the T1 counter under program control, so that T1 can be used as a sector counter. (It is not currently used in that mode)

P32: Sector Done (SECTDN). (Input) Resets the ~~P253~~ sector counter and the lower 4-bits of the RAM address counter when the read/write hardware is ~~searching for the target~~ sector (200 ns pulse). Goes low

PORT 3 (cont) 334

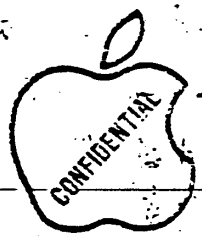
when the read/write operation has been completed. Stays low until

P37: CRC Error (CRCERR). (Gnout) Goes low if a CRC error is
CRCERR detected during a disk read.

P34, P35: Head Select 0, 1 (HS0, HS1). Binary coded bits
HS0, HS1 to select head 0, 1, 2 or 3.

P36: Write Sector Mark (WRTSM). Output. Causes sector
WRTSM marks to be written on the disk if FMTEN is
high and the Format jumper is installed.

P37: Read Header/Transmit (RDHDR/TX). Command input to the
RDHDR/TX read/write control circuits. If DRW (P27) is hi
and START/RATE (P26) is low, RDHDR hi will cause the header field on the
next sector to be stored in RAM.



335

CONFIDENTIAL

Introduction

1.1 General Description:

The ST506 disc drive is a random access storage device with two non-removable 5¼ inch discs as storage media. Each disc surface employs one movable head to service 153 data tracks. The total formatted capacity of the four heads and surfaces is 5 megabytes. (32 sectors per track, 256 bytes per sector, 612 tracks). 10 532

Low cost and unit reliability are achieved through the use of a band actuator and open loop stepper head positioning mechanism. The inherent simplicity of mechanical construction and electronic controls allows maintenance free operation throughout the life of the drive. Both Electronic PCB's are mounted outside HDA for field servicability.

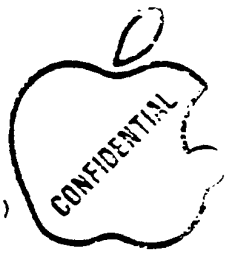
Mechanical and contamination protection for the heads, actuator and discs are provided by an impact resistant aluminum enclosure. A self contained recirculating system supplies clean air through a 0.3 micronfilter. A second port in the filter assembly allows pressure equalization with ambient air without chance of contamination. A patented spindle pump assures adequate air flow and uniform temperature distribution throughout the head and disc area. Thermal isolation of the stepper and spindle motor assemblies from the disc enclosure yields a very low temperature rise within the enclosure, providing significantly greater off track margin and the ability to immediately perform read and write operations after power up with no thermal stabilization delay.

The ST506 electrical interface is similar to the Shugart Associates SA1000 family of 8 inch fixed disc drives. The ST506 size and mounting is identical to the industry standard minifloppy disc drives and uses the same DC voltage and connector. No AC power is required.

Key Features:

- Storage Capacity of 6.38 megabytes unformatted, 5.0 megabytes formatted as shipped.
- Winchester design reliability, 9.5 gram head load force, 19 micro-inch flying height.
- Same physical size and mounting as the minifloppy.
- Same DC voltages as the minifloppy.
- Band actuator and stepper motor head positioning.
- 5.0 megabit/second transfer rate.
- Simple floppy like interface.
- Same track capacity as a double density 8 inch floppy.

336



- 170 millisecond random average access time, reducible to 95 MS via a simple software algorithm (See Section 4.5.3)

1.2 Specification Summary:

1.2.1 Physical Specifications:

Environmental Limits:

Ambient Temperature = 40° to 122°F (4° to 50°C)
Relative Humidity = 8 to 80%
Maximum Wet Bulb = 78° non-condensing

DC Power Requirements

+12 volts ±5%, 1.8 amps typical

4.5 amps maximum during power on

+5 volts ±5%, 0.7 amps typical, 1.0 amp maximum

Mechanical Dimensions:

Height = 3.25 inches
Width = 5.75 inches
Depth = 8.00 inches
Weight = 4.2 lbs. (1.9 kg)

Heat Dissipation = 25 watts typical
= 29 watts maximum

1.2.2 Reliability Specifications:

MTBF = 11,000 POH, typical usage
PM = Not required
MTTR = 30 minutes
Component Design Life = 5 years

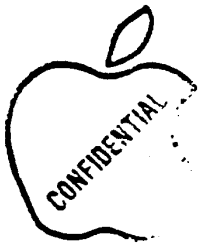
Error Rates:

Soft read errors = 1 per 10^{10} bits read
*Hard read errors = 1 per 10^{12} bits read
Seek errors = 1 per 10^{6} seek

*Not recoverable within 16 re-tries.

1.2.3 Performance Specifications:

337



Capacity

Unformatted		
	Per Drive	6.38 Megabytes
	Per Surface	1.59 Megabytes
	Per Track	10416 Bytes
Formatted		
	Per Drive	5.0 Megabytes
	Per Surface	1.25 Megabytes
	Per Track	8192 Bytes
	Per Sector	~~532~~ ~~356~~ Bytes 512
	Sectors per Track	~~32~~ 16
Transfer Rate		5.0 Megabits per second
Access Time		
	Track to Track	3 ms
	Average	170 ms, reducible to 95 ms. (See Section 4.5.3)
	Maximum	500 ms, reducible to 245 ms. "
	Setting Time	20 ms
Average Latency		8.33 msec

1.2.4 Functional Specifications

Rotational speed	3600 ~~rpm~~ ± 1%
Recording density	7690 ~~bpi~~ max
Flux density	7690 fci
Track density	255 tpi
Cylinders	~~153~~ 152
Tracks	~~612~~ 608
R/W Heads	4
Disks	2

2.0 Functional Characteristics

2.1 General Operation

The ST506 disc drive consists of ~~Read/Write and control electronics~~, read/write heads, track positioning actuator, media, and air filtration system. The components perform the following functions:

1. Interpret and generate control signals.
2. Position the heads over the desired track.
3. Read and write data.
4. Provide a contamination free environment.

CONFIDENTIAL

2.2 R[redacted]d Control Electronics

Electronics are packaged on two printed circuit boards. The primary board to which power, control and data signals are connected includes:

1. Index detection circuit.

2. Head position/actuator circuit.

3. Read/write circuits.

4. Drive up to speed circuit.

5. Head select circuit.

6. Write fault detection circuit.

7. Step motor drive circuit.

8. Drive select circuit.

9. Track zero detector circuit.

The [redacted] PCB, mounted to the baseplate under the ~~primary~~ board derives its power from the primary board and provides power and speed control to the spindle drive motor.

2.3 Drive Mechanism

A Brushless DC drive motor rotates the spindle at 3600 rpm. The spindle is driven directly with no belt or pulley being used. The motor is thermally isolated from the baseplate to minimize temperature rise in the sealed chamber containing the heads and discs. The motor and spindle are dynamically balanced to insure a low vibration level. A brake is used to provide a fast stop to the spindle motor when power is removed. The baseplate is shock mounted to minimize transmissibility of vibration to the chassis or frame.

2.4 Air Filtration System (Figure 1)

The discs and read/write heads are fully enclosed in a module using a integral recirculation air system with an absolute filter which maintains a clean environment. Integral to the filter is a port which also permits pressure equalization with the ambient air without contaminate entry.

2.5 Positioning Mechanism (Figure 2)

The read/write heads are mounted on a ball bearing supported carriage which is positioned by a band actuator connected to the stepper motor shaft. The stepper motor is thermally isolated from the baseplate to minimize temperature rise in the sealed chamber containing the heads and discs.

339

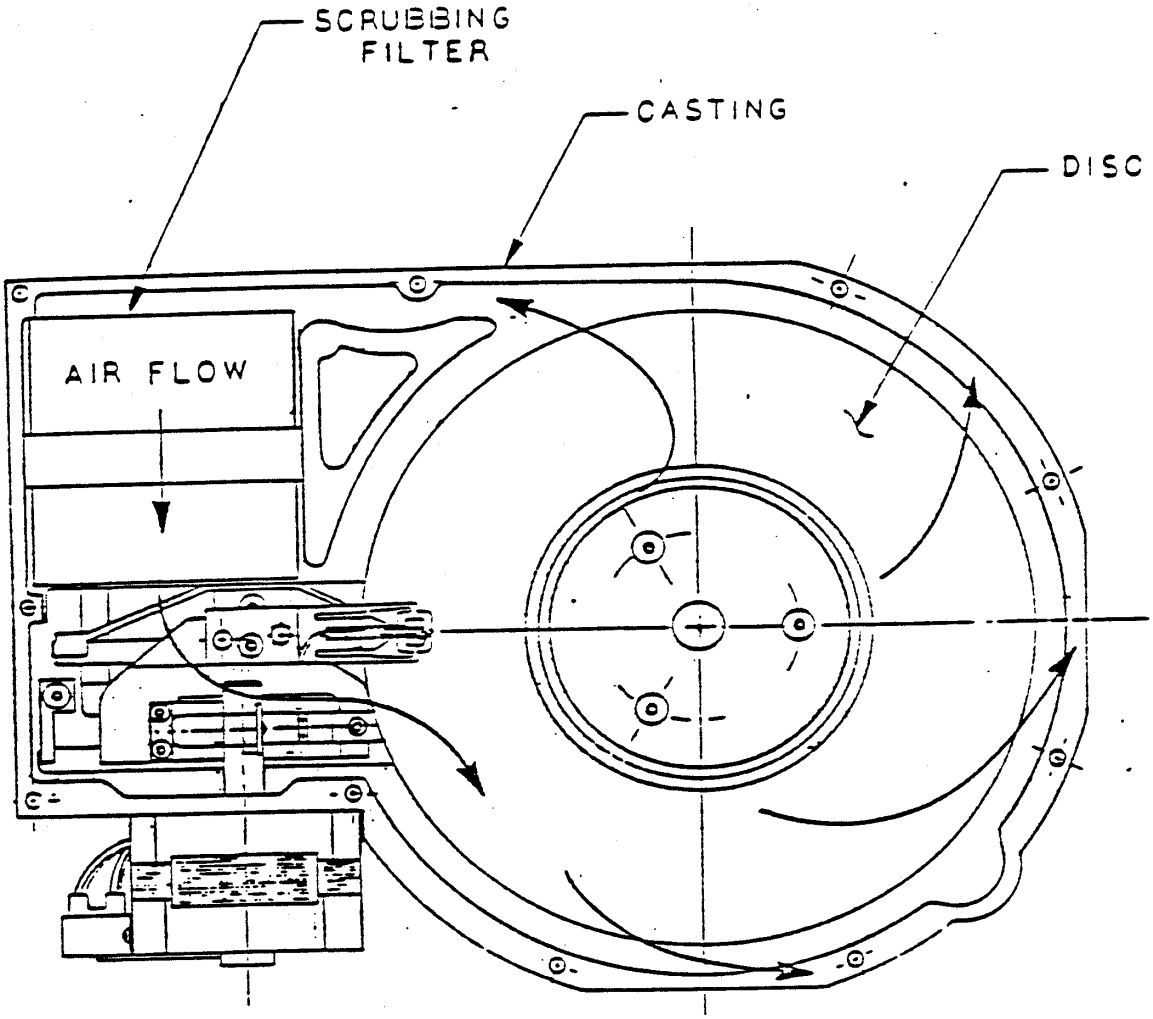


FIG 1A

69

340

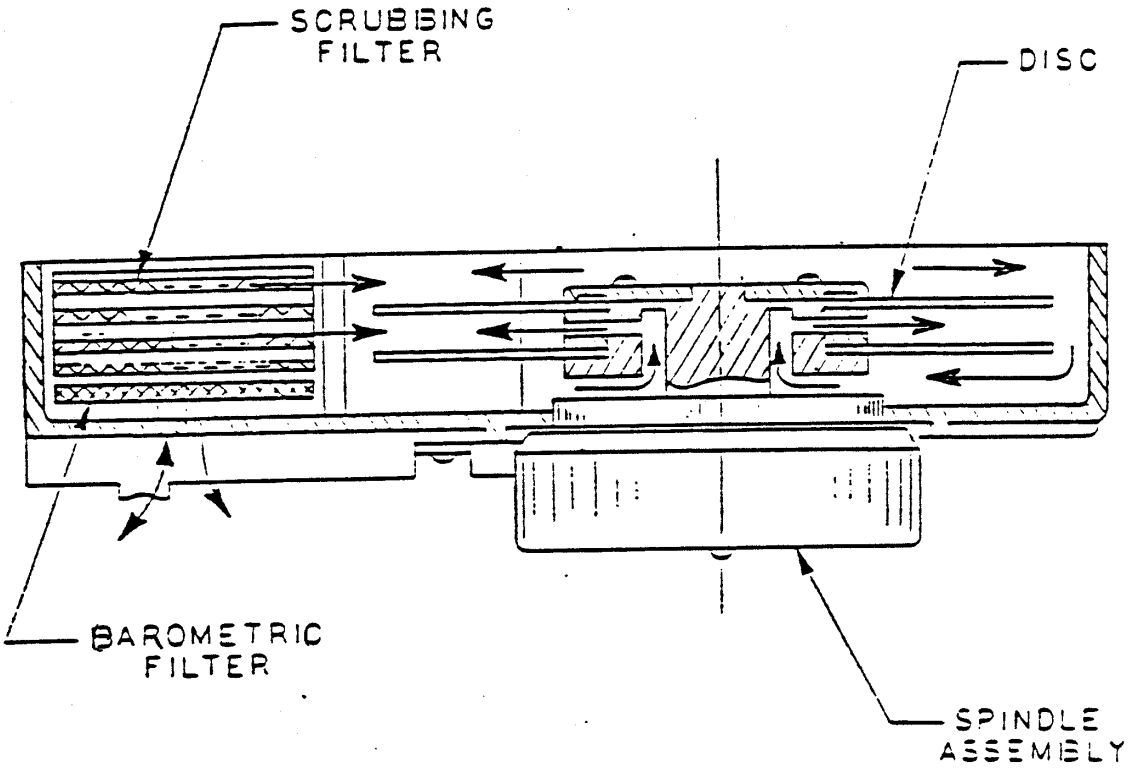


FIG 1B

341

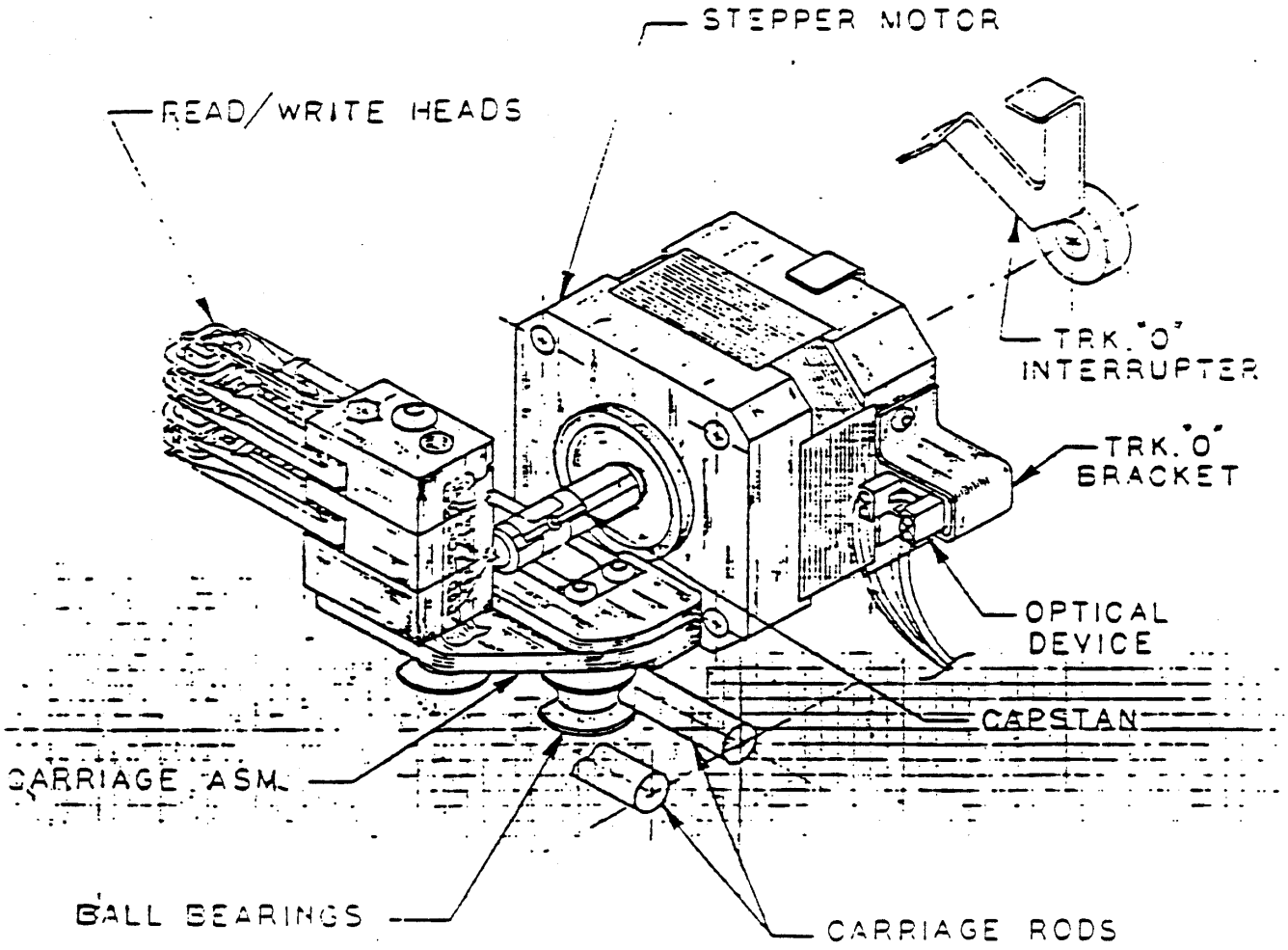
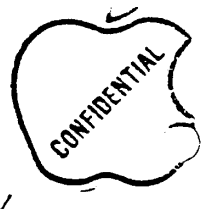


FIG 2

71

342



2.6 Read/Write Heads and Discs

The recording media consists of a lubricated thin magnetic oxide coating on a 130 mm diameter aluminum substrate. This coating formulation, together with the low load force/ low mass Winchester type flying heads, permit reliable contact start/stop operation.

Data on each of the four disc surfaces is read by one read/write head, each of which accesses 154 tracks.

3.0 Functional Operations

3.1 Power Sequencing (Figure 3)

+5 and +12 volts may be applied in any order. +12 volts must be applied to start the spindle drive motor. ~~A speed sense circuit counts 1000 disc revolutions.~~ After 1024 ~~revolutions, the heads will automatically recalibrate to track ØØ. (See Section 4.5.2 for exception). For this recalibration to occur, step input signal must be inactive. Track ØØ, seek COMPLETE and READY signals on the interface will become true simultaneously. The drive will not perform read & write or seek functions until READY becomes true.~~

3.2 Drive Selection

Drive selection occurs when one of the drive select lines are activated. Only the disc selected will respond to the input signals, and only that drive's output signals are then gated to the controller interface. (See Section 4.5.1 for exception).

3.3 Track Accessing

Read/write head positioning is accomplished by:

a) Deactivating write gate.

b) Activating the appropriate Drive Select Line.

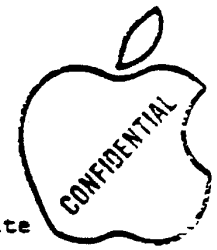
c) Being in the READY condition with SEEK COMPLETE true.

d) Selecting the appropriate direction.

e) Pulsing the step line.

Each step pulse will cause the heads to move either 1 track in or 1 track out depending on the level of the

343



4.1.6 DRIVE SELECT 1 - 4

DRIVE SELECT, when logically true, connects the drive to the control lines. Cutting the appropriate shunts at IC position 6C will determine which select line on the interface will activate that drive.

4.2 CONTROL OUTPUT LINES

The output control signals are driven with an open collector output stage capable of sinking a maximum of 40mA at logical zero or true state with maximum voltage of 0.4V measured at the driver. When the line driver is in the logical one or false state, the driver transistor is off and the collector cutoff current is a maximum of 250µA.

All J1 output lines are enabled by the respective DRIVE SELECT line.

Figure 7 shows the recommended circuit.

4.2.1 SEEK COMPLETE

This line will go true when the R/W heads have settled on the final track at the end of a seek. Reading or writing should not be attempted when SEEK COMPLETE is false.

SEEK COMPLETE will go false in three cases:

1) A recalibration sequence is initiated (by drive logic) at power on because R/W heads are not over track zero.

2) 500mS (typical) after the leading edge of a step pulse or series of step pulses.

3) If +5 volts or +12 volts are lost momentarily but restored.

4.2.2 TRACK ØØ

This interface signal indicates a true state or logical zero only when the drive's R/W heads are positioned at track zero (the outermost data track).

4.2.3 WRITE FAULT

This signal is used to indicate a condition exists at the drive that causes improper writing on the disk. When this line is true, further writing and stepping is inhibited at the drive until the condition is corrected. It cannot be reset via the interface.

344

There are three conditions detected:

a) Write current in a head without WRITE GATE active or no write current in the head with WRITE GATE active and DRIVE SELECTED.

b) Multiple heads selected, no head selected, or improperly selected.

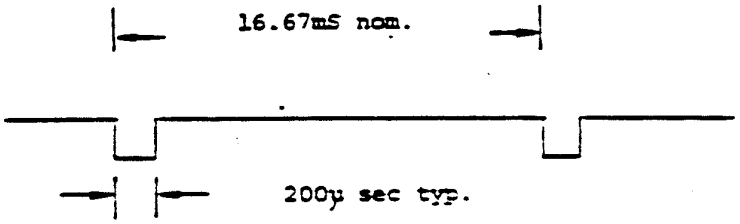
c) DC voltages are grossly out of tolerance.

4.2.4 INDEX

This interface signal is provided by the drive once each revolution (16.67mS nom.) to indicate the beginning of the track. Normally, this signal is a logical one level and makes the transition to the logical zero level to indicate INDEX. Only the transition from one to zero is valid. See Figure 9.

FIGURE 9

INDEX TIMING



4.2.5 READY

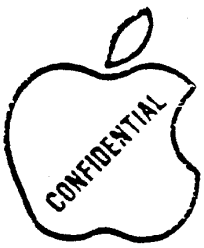
This interface signal when true together with SEEK COMPLETE, indicates that the drive is ready to read, write or seek, and that the I/O signals are valid. When this line is false, all writing and seeking are inhibited.

The typical time after power on for READY to be true is 15 seconds.

4.3 DATA TRANSFER LINES

All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. These lines are provided at the J2/P2 connectors on all drives.

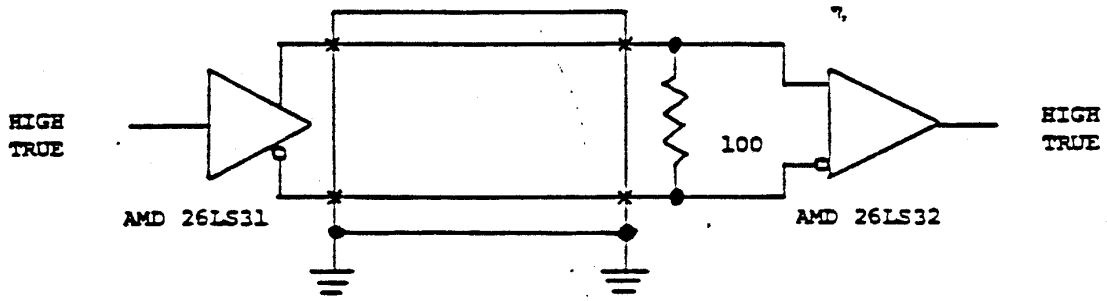
345



Two pairs of balanced signals are used for the transfer of data: WRITE DATA and READ DATA. Figure 10 illustrates the driver/receiver combination used in the ST506 drive for DATA TRANSFER signals.

FIGURE 10

DATA LINE DRIVER/RECEIVER COMBINATION



Z = 105Ω
FLAT RIBBON OR TWISTED PAIR MAX 20 FT.

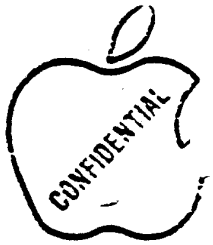
NOTE: ANY RS 422 DRIVER/RECEIVER PAIR WILL INTERFACE.

4.3.1 MFM WRITE DATA

This is a differential pair that defines the transitions to be written on the track. The transition of +MFM WRITE DATA line going more positive than the -MFM WRITE DATA will cause flux reversal on the track provided WRITE GATE in active. This signal must be driven to an inactive state (+MFM WRITE DATA more negative than -MFM WRITE DATA) by the host system when in a read mode.

To insure data integrity at the error rate specified, the write data presented by the host must be pre-compensated on tracks 64 through 152.

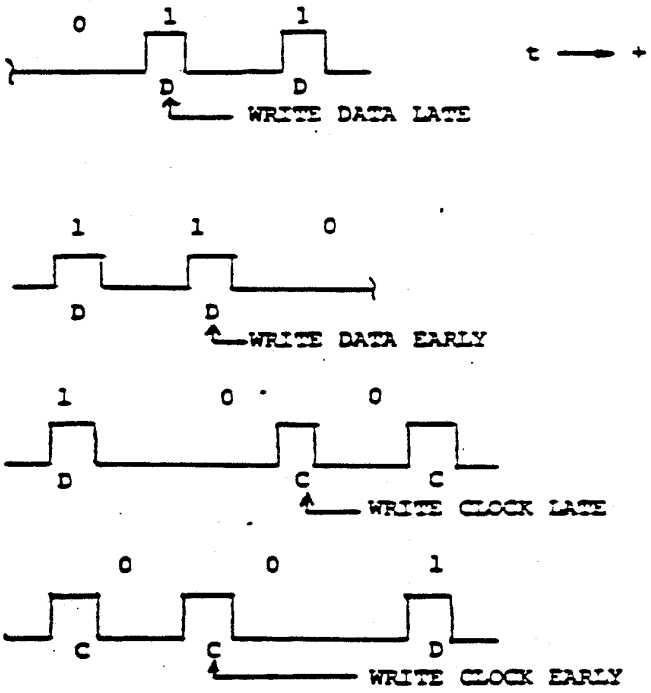
346



The optimum amount of pre-compensation is 12nS for both early and late written bits. Figure 11 shows the bit patterns to be compensated. All other patterns are written "on time."

FIGURE 11

WRITE PRE-COMPENSATION PATTERNS



Writing should occur out of a shift register which is used to observe the pattern. "On time" represents a nominal delay. Early and late represent less or more delay respectively.

347

5.4 J4/P4 Frame Ground Connector

Faston AMP P/N 61761-2

Recommended mating connector AMP 62187-1

Note that DC logic ground in the Printed Circuit Board is not connected to frame ground. Use of the frame ground connection is very important to reduce ground loops and noise problems which may cause excess errors.

If used, the hole in J4 will accomodate wire size of 18 AWG max.

6.0 Physical Specifications

This section describes the mechanical dimensions and mounting recommendations for the ST506.

6.1 Mounting Orientation

Recommended orientation is either vertical on either side or horizontal with PCB down. The only prohibited orientation is horizontal with PCB up (disk surface down, parallel and next to mounting surface). In the final mounting configuration, insure that operation of the four shock mounts which isolate the aluminum base casting from the frame is not restricted.

6.2 Mounting Holes

Eight mounting holes, four on bottom and two on each side are provided for mounting the drive into an enclosure. The size and location of these holes, shown in Figure 17, are identical to the industry standard minifloppy drive.

6.3 Physical Dimensions

Overall height/width/depth and other key dimensions are shown in Figures 17 and 18. As in the case of the mounting holes, the dimensions are identical to the minifloppy, allowing a direct physical replacement.

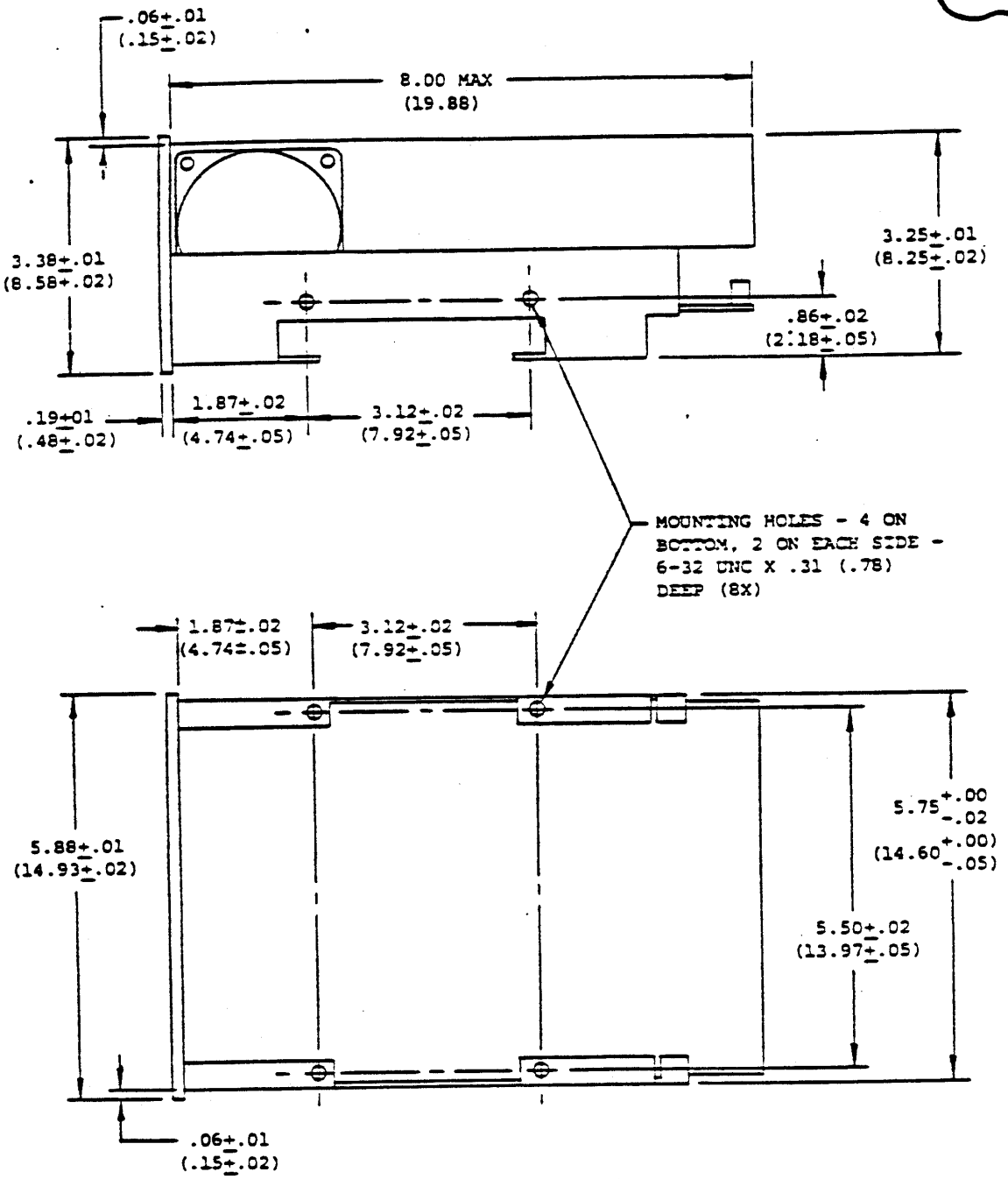
6.4 Shipping Requirement

During shipping the heads shall be positioned on track 152 to eliminate the possibility of damage. This shall be done while the discs are spinning.

348

FIGURE 17

MOUNTING PHYSICAL DIMENSIONS



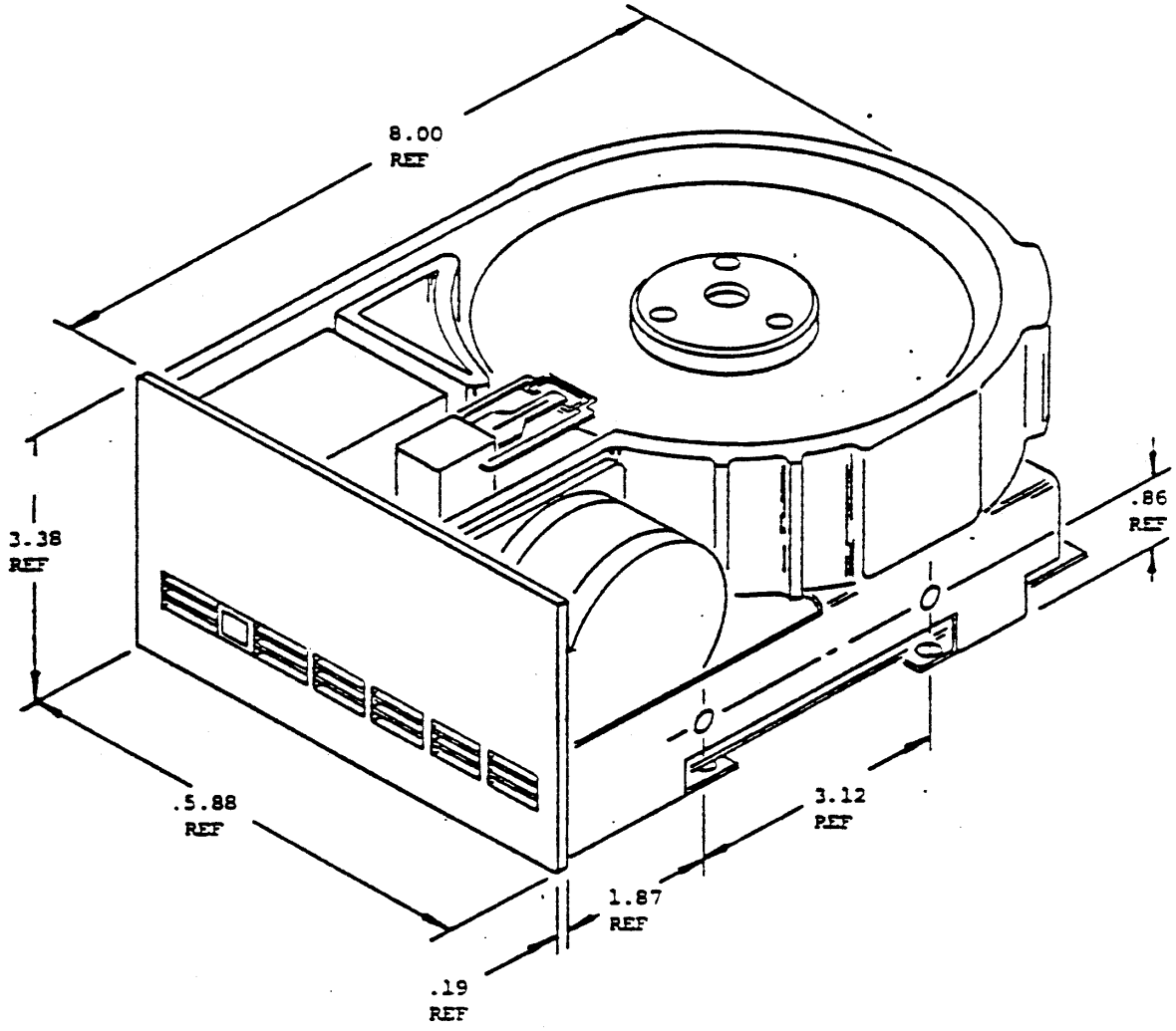
76

349

CONFIDENTIAL

FIGURE 18

OVERALL PHYSICAL DIMENSIONS



79

MLC		DESCRIPTION	INC BY	APPROVAL & DATE
		350		

APPLICABLE DOCUMENTS

Motor Control PCB Assembly 20003-001

Engineering Specifications:

The HDA components shall meet all requirements and specifications set forth in the following documents unless separately specified herein.

Main Control PCB (Fab drawing)	20019-001
Motor Control PCB (Fab drawing)	20004-001
HDA Assembly drawing	
HDS Electrical interconnect drawing	
Magnetic Disc	30126-001
Read/Write Heads	30125-001
Interface	30001-001
Head Connector	10420-016
Spindle Motor Connector	10417-006
Stepper Motor Connector	10417-005
Index Connector	10417-005
Brake Connector	10417-002

DWG. NO.

MODEL NO. FIRST USE	ST506	NEXT ASSY FIRST USE FINAL ASSY

DRAWN		
CHECK		
APPD(E)		
APPD(M)	Stig	1.19.81

UNLESS OTHERWISE SPECIFIED

DIMENSIONS ARE IN INCHES
TOLERANCES ON
DECIMALS .XX .XXX
ANGLES ±

SEAGATE TECHNOLOGY

HEAD DISC ASSEMBLY

SCALE	SIZE	DWG NO.	REV. EC
	A	30124-001	0011

DO NOT SCALE DRAWING | SHEET 1 OF 14

P 4 9 F. 11
EC

SHUGART TECHNOLOGY

Engineering Specification

	30124-001	0011

ST-506 HEAD DISC ASM SPECIFICATION

TITLE HEAD DISC ASSEMBLY

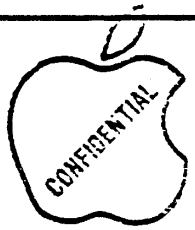
SHEET 2 OF

APPLICABLE DOCUMENTS (continued)

TK Ø Connector	10417-005
Spin Motor PCB Connector	10419-002
10416-XXX (Ref) Mates with	10417-XXX
10415-XXX (Ref) " "	10420-XXX
10418-XXX (Ref) " "	10419-XXX

CONFIDENTIAL

Engineering Specification	30124-001	0011
TITLE ST-506 HEAD DISC ASM SPECIFICATION 30124-001	SHEET 3	OF



CONTENTS

I. SCOPE

The Head Disc Assembly (HDA) is a sub-assembly of the ST-506 Final Disc Drive Assembly. It will undergo all the drive testing before being shipped to the customer. The Head Disc Assembly will be a complete ST-506 disc drive less the following hardware:

1) Main control PCB
2) Motor control PCB
3) Front panel

II. CAPACITY

Unformatted

Mbytes/Drive	6.38
Bytes/Track	10417

Formatted 32 Sectors/Track (Soft)

Mbytes/Drive	5.0
Bytes/Track	8192

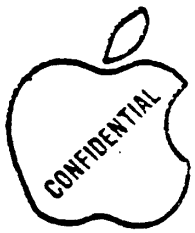
SEAGATE TECHNOLOGY

Engineering Specification

	30124-001	0011

TITLE ST-506 HEAD DISC ASM SPECIFICATION 30124-001

SHEET 4 OF



	Max recording frequency (MHZ)	2.5
	Transfer Rate	
	Mbits/Second	5.0
	Density	
	Flux Changes/Inch	7690

III. MEDIA

Tracks	
Per Inch	255
Per Surface	153
Per Drive	612

Defects (Hard Errors)

Any flaw defined as ≤ 2 bytes in length. Sequential, radial, and circumfrential defects to be counted separately.

Per Drive	32
Per Surface	8
Per Cylinder Zero	0

IV. ENVIRONMENTAL

	Operating	Shipping	Storage
Ambient Temperature*	39° - 135°F 4° - 57°C	25° - 144°F -4° - 62°C	-8° - 176°F -22°C - 80°C
Max Temperature Gradient/Hour	18°F	18°F	18°F
Relative Humidity	20 - 80%	1 - 95%	1 - 95%
Maximum Wet Bulb	78°	No Condensation	
Stray Magnetic Field	(1" from casting)	20 Gauss Max	
Altitude		10,000 ft. Max	

* Ambient Temperature is defined as the HDA casting temperature.

354 Engineering Specification | 30124-001 | 0011

TITLE ST-506 HEAD DISC ASM SPECIFICATION 30124-001 | SHEET 5 OF

CONFIDENTIAL

V. RELIABILITY

Error Rate (Excluding defects) - When used with Seagate Technology PCB P/N 20019-001

Soft read errors (16 retries min.)	1 per 10^{10} bits transferred
Hard read errors	1 per 10^{12} bits transferred
Seek errors	1 per 10^{6} seeks
MTBF (Meantime before failure)	
Typical usage	15,000 hours
PM (Preventive Maintenance)	None
MTTR (Meantime to Repair)	30 minutes
Component Life	5 years
Media Life	10,000 starts/stops

SEE NEXT PAGE →

VI. POWER

DC Voltages

+12 VDC ± 10%, 1.7 amps typical (3.3 amps during motor start)
(More detail on current - see ST-506 OEM Manual 30001-001)

BTU/Hr. (1 watt = 3.413 BTU/Hr.)

Watts	25	± 40%
BTU/Hr.	85.3	∓ 40%

VII. INTERFACE

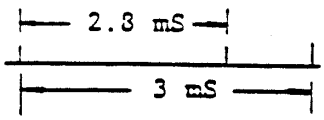
See ST-506 OEM Manual 30001-001

VIII. DRIVE

A. Mechanical

Seek Settle Time*
Uncontrolled 100mS
Controlled 20 mS using this step sequence

Track to Track (Seek = 3 ms)



* Using Seagate recommended PCB

E4

SEAGATE TECHNOLOGY

355 Engineering Specification

ST-506 HEAD DISC ASM SPECIFICATION
TITLE 30124-001

30124-001 | 0011

SHEET 6 OF

CONFIDENTIAL

Drive Motor Start Time* 15 sec.

Spindle Speed** 3600 rpm ± 1%

HDA Cleanliness

The air inside the HDA shall be class 100 (0.5 micron particles)

Mounting

See Figures 1 & 2

Vibration and Shock***

Operational 1g
Shipping 5g

B. Electrical

For purposes of normalized test data, and unless otherwise stated, all test measurements shall be taken at the ambient conditions of 68 + 5°F (20 + 3°C) and 40 to 60% relative humidity after the HDA has been stabilized at the test environment for at least 1 hour.

Not withstanding this, the HDA shall be capable of meeting all the requirements of this specification over the full operating environment of Paragraph IV, Page 4.

Warm Up Time

The HDA shall meet all operating specifications within 15 seconds after spindle motor has reached 3600 RPM at all operational environmental conditions.

1. Spindle motor speed regulation 3600 ± 1% RPM

2. Spindle motor current

Start	3.6 amps max.
Running	1.0 amps max.

3. Stepper motor/phase 250 ± 50 Ma Typical

* Using Seagate recommended PCB
** Adjustments are customer responsibility
*** For shipment, the heads shall be moved to ID while the spindle is rotating

E5

CONFIDENTIAL

4. Performance (Head Disc Asm)

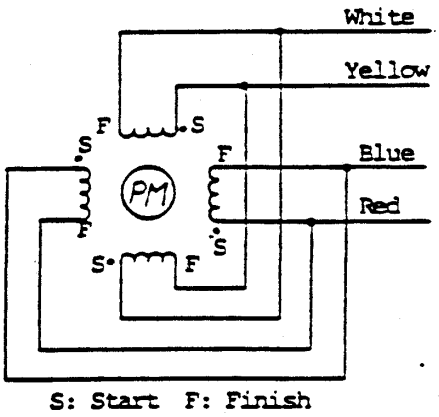
Average 2F amplitude	1.0 mv p-p min.
Inside window margin (without write precomp)	30 ns total
Minimum resolution on any track (without precomp.)	60%

Wide Gap Measurement

1. DC erase tracks 150, 151, 152
2. Write 2F on track 151
3. Write 1F on tracks 150 & 152
4. Measure 1F amplitude relative to 2F amplitude on track 151 using H.P. spectrum analyzer Model No. 3585A or equivalent.
5. Failure is defined as a measurement $\geq$ - 26dB.

EG

Engineering Specification	30124-001	0011
ST-506	SHEET 8	OF
TITLE HEAD DISC ASSEMBLY		



S: Start F: Finish

ELECTRICAL REQUIREMENTS

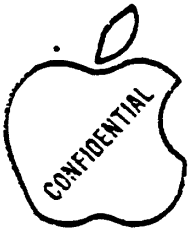
STEPPER MOTOR

Switching Sequence for CCW Rotation
Facing Mounting End

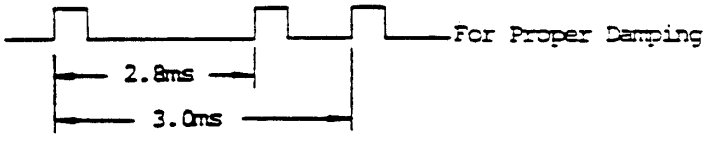
Step	Pin 4 Blue	Pin 5 Red	Pin 2 White	Pin 1 Yellow	Pin 3 Not Used
1	-	+	+	-	
2	+	-	+	-	
3	+	-	-	+	
4	-	+	-	+	
5	-	+	+	-	

SCHEMATIC

SPECIFICATIONS
Step per revolution: 200 (1.8° per step)
Step to step accuracy (Notes 1,2,): ± 5%
Positional Accuracy (1,3): ± 5%
Rotor Inertia: 20 Gcm² (.28 MOISS)
DC Phase Resistance: 38 ± 3.8 at 25° C
Phase Inductance: 52mH ± 20% at 1 KHZ
Phase Voltage: 9.2
Phase Current (Steady State): 240ma
Holding Torque: 864 gcm (12 oz. - IN) Min.
Detent Torque: 36 gcm (0.5 oz. - IN) Min.
Pull-out Torque: 504 gcm (7 oz. - IN) NOM.



NOTES:

1. Measurements made at rated current on each phase.
2. Between any two adjacent step positions.
3. Maximum error in 360°.
4. Motor to be driven bipolar.
   The above specifications is of a motor in bipolar mode.
5. Leads: 4, No. 26 AWG PVC insulation UL & CSA approved.
6. 

EE

SHUGART TECHNOLOGY

368 Engineering Specification

30124-001 | 0011

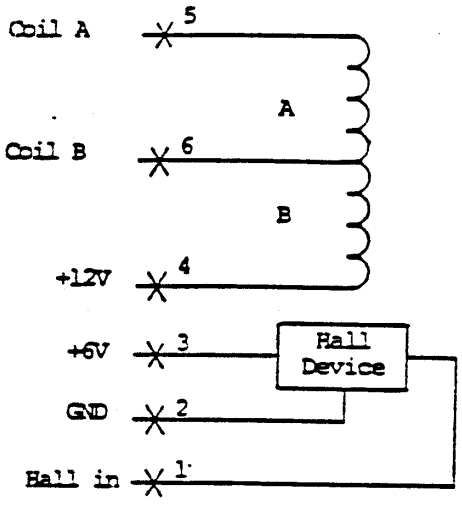
SHEET 9 OF

ST-506

TITLE HEAD DISC ASSEMBLY

ELECTRICAL REQUIREMENTS

SPINDLE MOTOR



Resistance of Winding:
A Or B = 2Ω NOM

Inductance of Winding:
A Or B = 1.6 mH NOM

Hall Device Asymmetry
≤ - 30 db.

NOTES:

All measurements shall be done with two discs.

1. Rated Speed : 3600 RPM
2. Input Voltage : 12 VDC ± 10%
3. Start Time : 15 Sec (Max) at Low Voltage
4. Stop Time : 15 Sec (Max)
5. Inertia of Discs: .04 Oz. - In - Sec^2
6. The horizontal magnetic field in a plane parallel to the plane of the disc shall be less than 0.5 gauss.

359 Engineering Specification

ST-506 HEAD DISC ASM SPECIFICATION

TITLE 30124-001

30124-001 | 0011

SHEET 10 OF

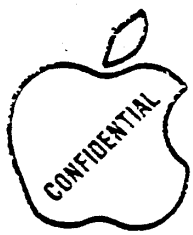
IX. PHYSICAL REQUIREMENTS

The HDA can be mounted on any side except upside down or front and back. (See Figures 1 & 2)

The HDA is shock mounted for vibration isolation. In the final mounting configuration, care shall be taken to insure that the operation of the four shock mounts is not restricted.

A shipping bracket shall be provided to hold the two side frames together to minimize shipping damage.

During operation, the side frames shall be provided with a tie bracket to simulate the front cover, in applications where the ST-506 front cover is removed.



SHUGART TECHNOLOGY

300

30124-001 | 0011

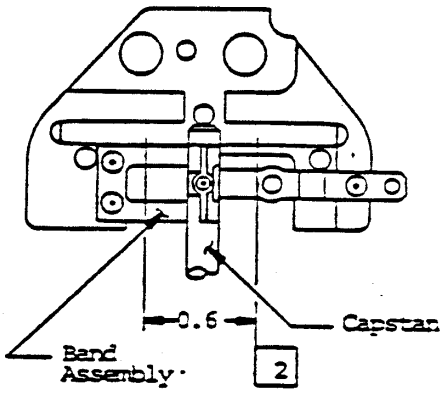
ST-506 Head Disc ASM Specification

TITLE 30124-001

SHEET 11 OF

CONFIDENTIAL

BAND ASSEMBLY



BAND SPECIFICATIONS

1. Band shall be capable of withstanding 2.5 lbs. Max. tension.

2. There shall be no creases on the active portion of the band.

3. Band shall be free of contaminants that would decrease it's life, such as finger prints and edge nicks.

C-7

360 Engineering Specification	30124-001	0011
ST-506 TITLE HEAD DISC ASSEMBLY	SHEET 12 OF	

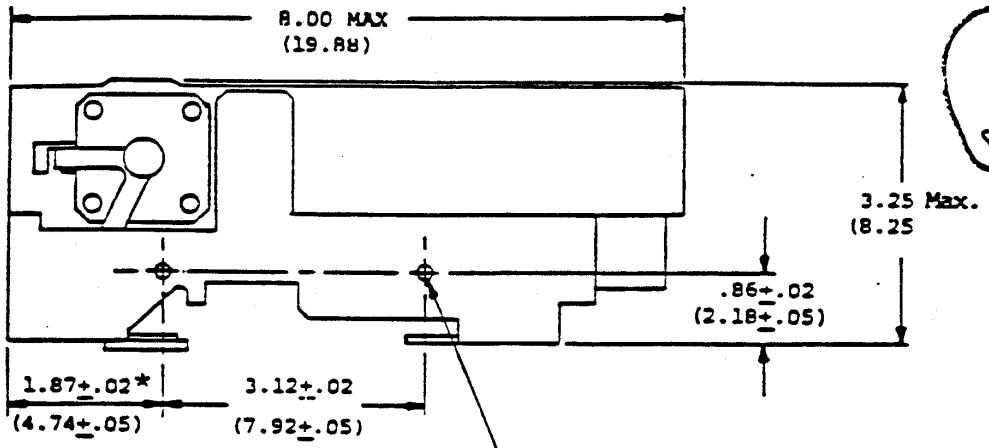


Fig. 1

MOUNTING HOLES - 4 ON BOTTOM, 2 ON EACH SIDE - 6-32 UNC X .25 (.78) DEEP (8X)

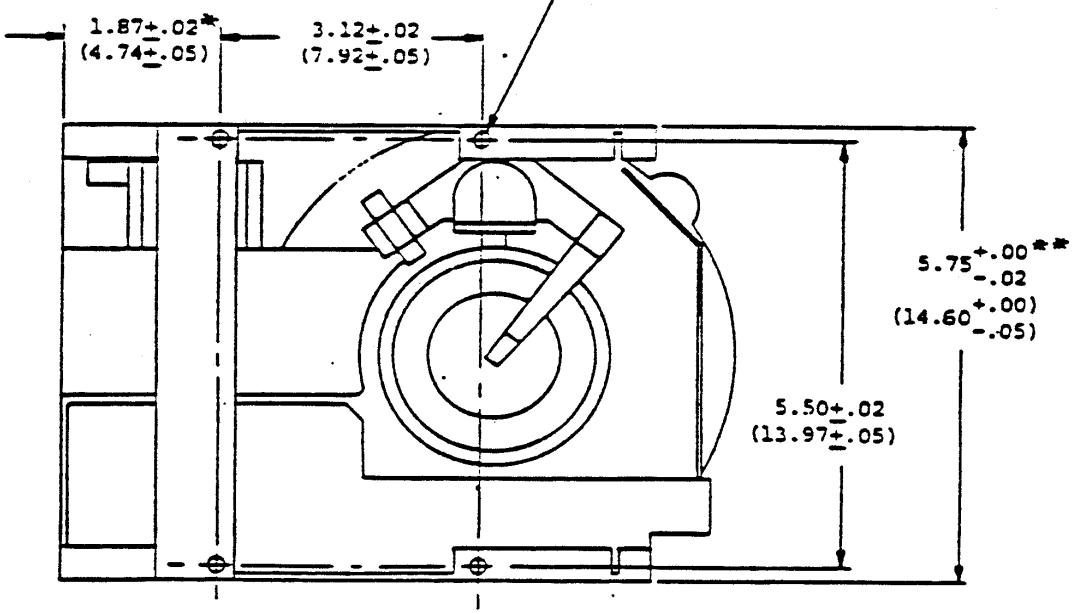


Fig. 2

* This Dimension would eventually become 1.75 inches. (4.44)

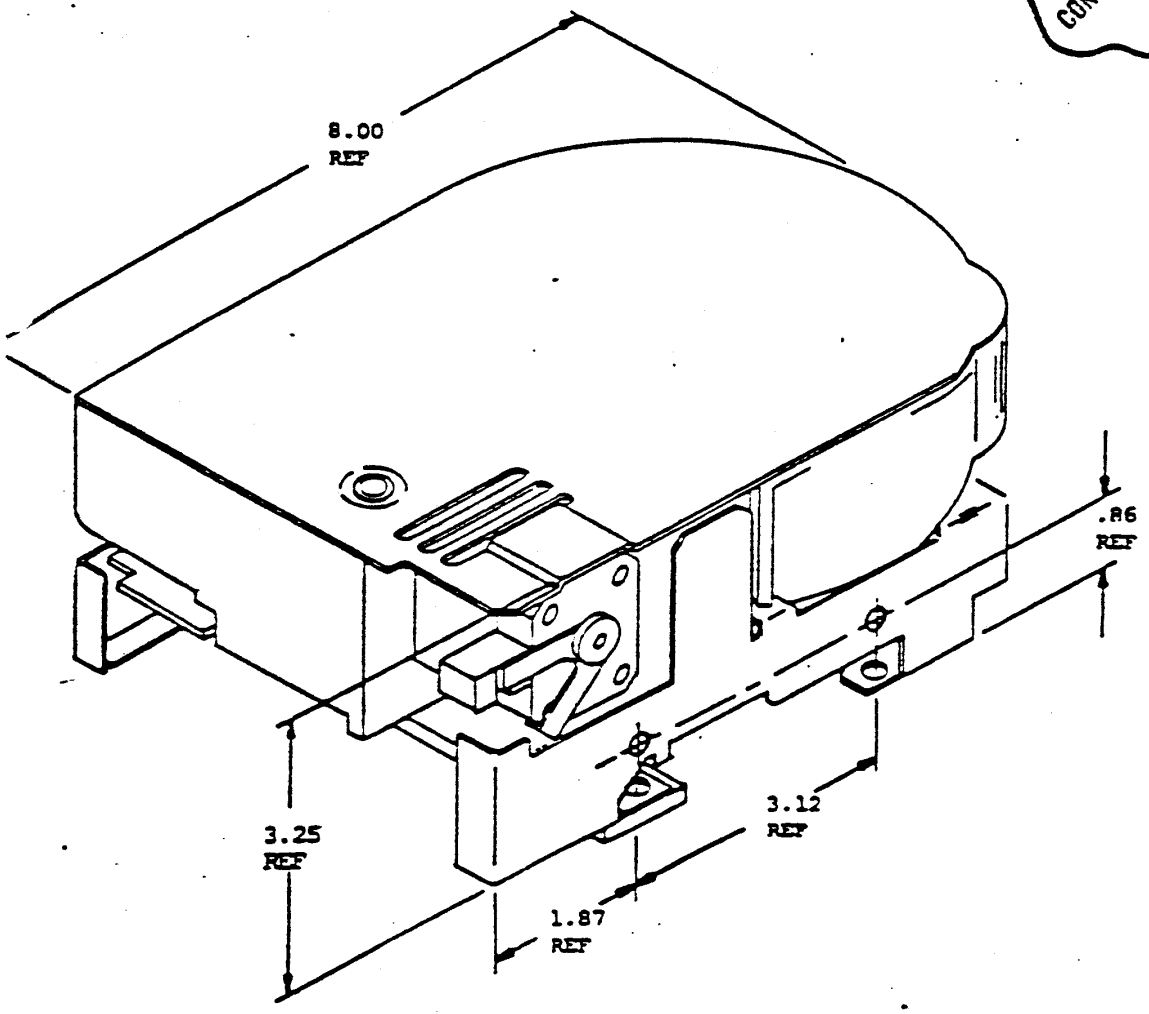
** This dimension has been off speced to a maximum of 5.78

MOUNTING DIMENSIONS

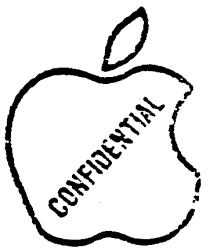
91

Engineering Specification	30124-001	0011
360" ST-506 HEAD DISC ASSEMBLY	SHEET 13 OF	

OVERALL PHYSICAL DIMENSIONS



92



363

APPENDIX A

I/O PIN ASSIGNMENTS OF PROFILE SYSTEM

SECTION 1—INTERFACE CARD

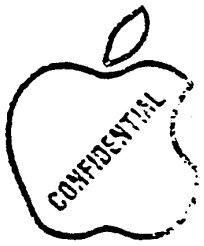
SLOT EDGE CONNECTOR

PIN NUMBER	SIGNAL NMEMONIC	DIRECTION
1	/I/O SELECT	INPUT
2	A0	INPUT
3	A1	INPUT
4	A2	INPUT
5	A3	INPUT
6-17	N.C.	
18	R/W	BI
19	PH0	INPUT
20-24	N.C.	
25	+5VDC	INPUT
26	GROUND	INPUT
27	/DMAOK	INPUT
28	/DMA1	OUTPUT
29	N.C.	
30	/IRQ	OUTPUT
31	/RESET	INPUT
32	N.C.	
33	-12VDC	INPUT
34	-5VDC	INPUT
35-36	N.C.	
37	Q3	INPUT
38	PH1	INPUT
39	/C02X	INPUT
40	N.C.	
41	/DEVSEL	INPUT
42-49	D7-D0 RESECTIVELY	BI
50	+12VDC	INPUT

INTERFACE/CONTROLLER CONNECTOR "P1"

PIN NUMBER	NMEMONIC	DIRECTION
1	PH0	OUTPUT
2	SIGNAL GROUND	OUTPUT
3	TR/W	OUTPUT
4	SIGNAL GROUND	OUTPUT
5	XD0	BI
6	XD1	BI
7	N.C.	
8	XD2	BI
9	SIGNAL GROUND	OUTPUT

364



10	"	"
11	XD5	BI
12	XD6	BI
13	XD7	BI
14	SIGNAL GROUND	OUTPUT
15	/PSTRB	OUTPUT
16	/BSY	INPUT
17	/CMD	OUTPUT
18	/RPARITY	INPUT
19	SIGNAL GROUND	OUTPUT
20	"	"
21	/CRES	OUTPUT
22	XD3	BI
23	XD4	BI
24	SIGNAL GROUND	OUTPUT
25	/CDET	INPUT

NOTE: ONE MAY USE THIS TABLE FOR THE CONTROLLER CARD P1 JUST REVERSE OUTS/INS

SECTION 2

CONTROLLER CARD

INTERFACE/CONTROLLER "P1"

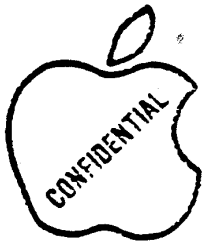
SEE P1 IN SECTION 1

CONTROLLER TO ANALOG CARD "P2"

PIN NUMBER	NMEMONIC	DIRECTION
1	SIGNAL GROUND	OUTPUT
2	/TRK0	INPUT
3	SIGNAL GROUND	OUTPUT
4	INDEX	INPUT
5	SIGNAL GROUND	OUTPUT
6	POWEROK	INPUT
7	SIGNAL GROUND	OUTPUT
8	/WRTSM	OUTPUT
9	SIGNAL GROUND	OUTPUT
10	10MHZ	OUTPUT
11	SIGNAL GROUND	OUTPUT
12	NRZWDTA	OUTPUT
13	SIGNAL GROUND	OUTPUT
14	MTR DRIVE 0A1	OUTPUT
15	SECTOR	INPUT
16	MTR DRIVE 0A2	OUTPUT
17	+12VDC	OUTPUT
18	MTR DRIVE 0B2	OUTPUT
19	-12VDC	INPUT
20	MTR DRIVE 0B1	OUTPUT
21	+5VDC	OUTPUT
22	/WTGT	OUTPUT

94

365



23	+5VDC	OUTPUT
24	/PRECOMP*LOCUR	OUTPUT
25	+5VDC	OUTPUT
26	HS0	OUTPUT
27	+5VDC	OUTPUT
28	HS1	OUTPUT
29	SIGNAL GROUND	OUTPUT
30	RDGT	OUTPUT
31	SIGNAL GROUND	OUTPUT
32	/NRZDATA	INPUT
33	SIGNAL GROUND	OUTPUT
34	/RDCLK	INPUT

NOTE: ONE MAY USE THIS TABLE FOR ANALOG CARD J15, REVERSE INS AND OUTS

POWER SUPPLY TO CONTROLLER "P3"

PIN NUMBER	NMEMONIC	DIRECTION
1	+5VDC	INPUT
2	+12VDC	INPUT
3	POWEROK	INPUT
4	-12VDC	INPUT
5	GROUND	INPUT

CONTROLLER TO LED "P4"

PIN NUMBER	NMEMONIC	DIRECTION
1	/READY	OUTPUT
2	R17 SOFT FIVE	OUTPUT

CONTROLLER TO TEST "P5"

PIN NUMBER	NMEMONIC	DIRECTION
1	+5VDC	OUTPUT
2	5MHZ	OUTPUT
3	+5VDC	OUTPUT
4	10MHZ	OUTPUT
5	GROUND	OUTPUT
6	SYSCLK	OUTPUT
7	GROUND	OUTPUT
8	/PH0	OUTPUT
9	GROUND	OUTPUT

SECTION 3

ANALOG CARD

36P ANALOG TO HEADS

J5

1	HD0Y	BI
2	HD-0	OUT
3	HD0X	BI
4		
5	HD1Y	BI
6	HD-1	OUT
7	HD1X	BI
8		
9	HD2Y	BI
10	HD-2	OUT
11	HD2X	BI
12		
13	HD3X	BI
14	HD-3	OUT
15	HD3Y	BI

RNA TO TRACK 0
J8

1	GROUND	
2	TRK0*	IN
3	R114 SOFT	OUT
4	GROUND	OUT

CONFIDENTIAL

J6 RNA TO INDEX

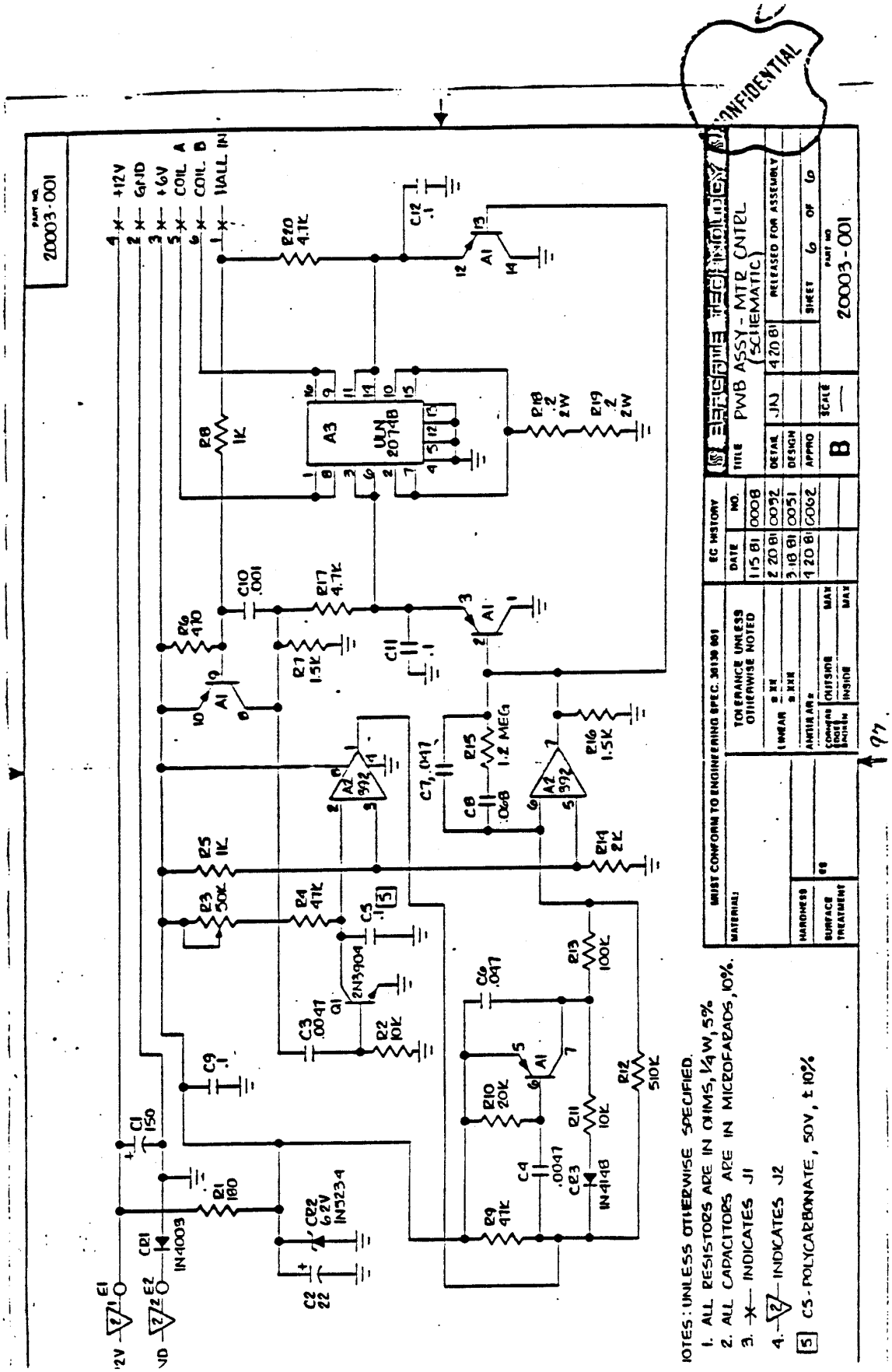
1		
2	GND	OUT
3	[redacted] INDEX*	IN
4		
5	R117 SOFT FIVE	OUT

J7 RNA TO STEPPER

1	-Ø3	OUT
2	+Ø3	OUT
3	GROUND	OUT
4	-ØA	OUT
5	+ØA	OUT

96

367



3.2 Removals/Adjustments

3.2.1 Main Control P.C.B.

Tools Required: 5/64" Hex Driver
Loctite #242

1) Remove (4) 6-32 x 1/4" button head Screws

2) Disconnect J4 through J8, noting their positions

3) Slide the P.C.B. toward the rear of the drive and remove

4) To re-install, reverse the above procedure using Loctite on the first 3 threads of all 6-32 screws.

3.4.2 Motor Control P.C.B

Tools Required: 5/64" Hex Driver
Loctite #242

1) Remove (4) 6-32 x 1/4" Main Control P.C.B. Mounting Screws

2) Disconnect J4 through J8, noting their positions

3) Disconnect J1 and J2, noting their operation

4) Remove the Motor Control P.C.B.

5) To re-install, reverse the above procedure using Loctite on the first 3 threads of all 6-32 screws.

Note: Whenever the Motor Control P.C.B. is replaced or any of its components are changed, the Spindle speed must be adjusted. Connect a frequency counter to Test Point 4 (Index) and adjust R3 on the Motor Control P.C.B. (accessible through the sideframe) to a period of 16.67 msec. Additionally, if C5 is changed, the Motor Control board should be burned in for at least 12 hours settle the capacitor before adjusting the speed.

3.2.3 Spindle Brake

Tools Required: 5/64" Hex Driver
Loctite #242
10 Mil shim (.010")

1) Remove the Main Control P.C.B. as outlined in Section 3.2.1

2) Disconnect J2 at the Motor Control P.C.B.; free the brake wires from their retaining clips

3) Remove (1) 6-32 x 1" button head brake mounting screw, washer, and spacer.

4) Remove the brake solenoid

369

5) To re-install, reverse the above procedure using Loctite on the first 3 threads of the 6-32 screws. Do not tighten the brake mounting screw

6) Insert the 10 mil shim between the brake pad and the spindle motor housing

7) Move the brake toward the spindle motor housing until a slight resistance is felt on the shim

8) Tighten the 6-32 brake mounting screw and verify that the contour of the brake pad aligns with the spindle motor housing

9) Continue the re-installation procedure

3.2.4 Index Sensor

Tools Required: 5/64" Hex Driver
Loctite #242
30 mil shim

1) Remove the Main Control P.C.B. as outlined in Section 3.2.1

2) Remove the connector from the front panel LED, noting the orientation of the connector

3) Free the Index wires from their retaining clips.

4) Remove (2) 6-32 x 1/8" button head Index mounting screws and remove the Index mounting bracket

5) Remove the Index sensor

6) To re-install, reverse the above procedure using Loctite on the first 3 threads of all 6-32 screws.

7) Check the gap between the index sensor and the motor hub using the 30 mil shim. If the gap is not 30 mils, loosen the two index screws and adjust it.

3.2.5 Ground Spring

Tools Required: 5/64" Hex Driver
Loctite #242

1) Remove the Main Control P.C.B. as outlined in Section 3.2.1

2) Remove (1) 6-32 x 1/8" button head ground spring mounting screw

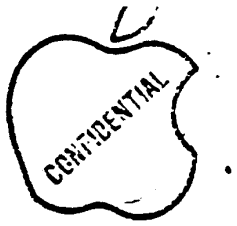
3) Remove the ground spring

4) To re-install, reverse the above procedure using Loctite on the first 3 thread of all 6-32 screws

5) Verify that the ground spring/spindle motor contacts are clean and making good contact. Also center the ground spring over the spindle motor ball.

99

370



3.2.6 Front Cover

Tools Required: 5/64" Hex Driver
Loctite #242

1) Remove the Main Control P.C.B. as outlined in Section 3.2.1

2) Remove the connector from the front panel LED, noting the orientation of the connector

3) Remove (2) 6-32 x 1/8" button head fron panel mounting screws

4) Remove the front panel

5) Remove the LED and grommet from the front panel, noting the orientation of the LED by the small dots on the connector end of the LED

6) To re-install, reverse the above procedures using Loctite on the first 3 threads of all 6-32 screws

3.2.7 Front Cover LED

Tools Required: 5/64" Hex Driver
Loctite #242

1) Remove the Main Control P.C.B. as outlined in Section 3.2.1

2) Remove the connector from the front panel LED, noting the orientation of the connector.

3) Remove the LED and grommet from the front panel, noting the orientation of the LED by the small dots on the connector end of the LED

4) To re-install, reverse the above procedure using Loctite on the first 3 threads of all 6-32 screws

3.2.8 Side Frames

Tools Required: 5/64" Hex Driver
Loctite #242

1) Remove the Main and Motor Control P.C.B.'s as outlined in Section 3.2.2

2) Remove (2) 6-32 x 1/8" button head front cover mounting screws. Do not disconnect the LED connector

3) Remove (4) 6-32 x 5/16" button head side frame mounting screws, grommets, and spacers

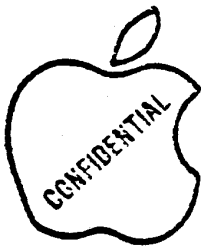
4) Remove the right and left side frames

100

371

5) To re-install, reverse the above procedure using Loctite on the first 3 threads of all 6-32 screws

Note: Removal of any assembly not covered in this section is not possible without special clean room facilities and tools.



FINIS

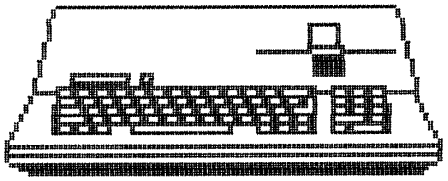
101

372

6

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

DskDbg – Block Device Debugger
Version E00.17 Service Manual

Author: Apple

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

BROMAGEM 4/85

373 DAVID BROMAGEM (Saratoga, CA) 1991?

Pages: 26 (#ed 1-24)

DSKDBG - BLOCK DEVICE DEBUGGER

VERSION E00.17

SERVICE MANUAL

RT4D — STATUS (DEBUGGER V11)
H 2(3)

(i)

374

TABLE OF CONTENTS

375

Block Device Debugger

1. INTRODUCTION

The Formatter/Debugger program is provided as a service aid for troubleshooting block I/O devices such as the ProFile disk drive. It is a program that allows you to directly communicate with a block I/O device and exercise it through a series of unique commands or build simple test sequences that will assist in debugging the device.

It is a particularly helpful tool for servicing devices that have been returned because of a malfunction, etc. For example, the program can be used to detect an error and then by using the loop on error function, you can use an oscilloscope to analyse the circuit that caused the error.

You can use the program to service a drive containing either the Formatter/Debugger ROM or the Standard ROM. (The Standard ROM is the operational ROM that is shipped in the disk drive.) The command summary, section 3, specifies which type of ROM must be in the drive to use a given command.

1.1 HOW TO USE THIS PROGRAM

To use this debugger properly, you should be aware of some of the design concepts. In designing the user interface (command structure) it was decided to use single character commands. This permits you to type in the command very rapidly and use options (which are not always required) to modify certain test variables. The most used test variables are the three byte logical block that is treated as either a 24-bit number or as three 8-bit numbers. As a 24-bit number, the variable represents a standard logical block with a decimal range of 0-16 million blocks (Hex 000000- FFFFFF). As three 8-bit numbers, each variable has a range of 0-255 (Hex 00-FF).

The program makes no assumptions about the use of the test variable; it just gives you two ways to talk to it. You can consider it as a large (24 bit) number or as three smaller (8 bit) numbers. This variable is sent to the disk exactly as a 24-bit number which the firmware decides how to interpret. You, the user, must know how the firmware will react to understand what the firmware is doing with the block number.

1.2 TYPICAL COMMAND EXAMPLES

To help make the concept of this program more clear, here are some examples that show what the more common commands do.

376

R (Read)

This command requests data to be read from the unit being tested. It passes the command and the BLOCK variable to the unit and transfers the data from the unit to the input buffer (see the display command, section 2.2.3, for information on displaying the input buffer).

R by itself does not change the BLOCK variable; it uses its current value. However, you can change the block variable by adding a modifier to the command in the form of a number. This number is normally treated such that leading zeros are assumed, thus 0, 00, 0000, etc. all produce the 24-bit value 000000. For example:

R0 or R00 or R000000, etc. (read block 000000)
R13 or R013 or R 0013, etc. (read block 000013)

In the above example, the BLOCK variable sent is either 000000 or 000013. The number 13 is the same as 013 or 000013 and produces the value 000013 in the BLOCK variable.

NOTE

The number 13 in the example is a HEX number, not a decimal number.

13 Hex = 19 Decimal

Remember that the firmware in the disk drive being tested determines how the BLOCK variable will be treated. Normally it is treated as a logical block.

Next, let's change the command as shown in the following example:

RT1H2S3 or RT01H02S03, etc. (read block 010203)

Notice that the command now has 3 modifiers in the form of T and a number, H and a number, and S and a number. The modifier T references the first 8-bits of the BLOCK variable, the H references the second 8-bits, and the S references the third 8-bits. Note that they are separate modifiers and can be used independently. For example:

If BLOCK = 000000, then the command RT5 will set BLOCK = 050000
If BLOCK = 050000, then the command RH3 will set BLOCK = 050300
If BLOCK = 050300, then the command RT16S9 will set BLOCK = 160309

Note that the last command in the example happened to affect the first and third bytes. Had the last command been RS9 then the BLOCK would have been 050309.

REMEMBER! The numbers used are always Hexidecimal numbers.

T=0-97
S=0-15
H=0-3

H = 0-3
T = 0-97
S = 0-15
B = 0- ----97740

377

W (write)

This command requests data to to be written to the unit being tested. It passes the command and the BLOCK variable to the unit and transfers the data to the unit from the output buffer (see the buffer fill command, section 2.2.1, and the display command, section 2.2.3).

W by itself does not change the BLOCK variable; it uses its current value. However, as was the case with a read command, the BLOCK variable can be changed by adding a modifier. For example, the commands W0, WT3H2S1, WH9, etc. have the same effect on the BLOCK variable that they do in the Read command.

+ (plus)

This command increments the BLOCK variable. + by itself increments the variable by 1. For example, if the BLOCK variable was 00001F, then after + it will be 000020 (Note the hex numbers). +3 will increment the BLOCK by 3 each time. +3T will increment the first 8-bit variable by 3 each time. For example, if the BLOCK was 050311, then +3T will change it to 080311. H and S work the same way to modify the second and third groups of 8-bit variables.

To provide wraparound and carry, the + command will wrap a 24-bit number at FFFFFF. It will wrap T at 97 (97 increments to 0) and will set H an S to 0 to provide full wrap. H will wrap at 3 (3 goes to 0) creating a carry to T and S will wrap at F (F goes to 0) creating a carry to H. Future enhancements will be to find out what the maximum block count for a device is and then wrap the 24-bit number at that count.

D (Display)

This command displays the contents of the input or output buffers.

D by itself will display the input buffer.
DI is the same as D.
DO will display the output buffer.
DS will decode the contents of the input buffer as extended status information.

The form of the display is the same as that for an APPLE monitor memory dump. It will display the first 256 bytes of the buffer and pause so that you can study the values. At the bottom of the screen it asks for ESCAPE to terminate, RETURN to quit, ANY other key to continue. ESCAPE stops further display and cancels the rest of the command line effectively stopping processing and returning

S - SCAN

378

control to the user. RETURN quits the display command and causes the program to go on to the next command. Pressing any of the other keys, except the space bar, will allow the program to display the next 256 bytes of the buffer. Pressing the space bar will allow you to display the buffer line by line.

CAUTION

If you just keep hitting a key to see the next 256 bytes you will see first the buffer you requested and then memory above it. There are areas of ram above the buffers that will, when displayed cause your screen to get VERY sick and you will have to either re-boot the program or be very familiar with APPLE III hardware and monitor.

Just incase you are not familiar with the monitor memory display, here is an example. (The monitor doesn't display the descriptive information above the dashed line.)

```
  Hex
Address  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
-------------------------------------------------------
 8000:  00 1A 23 92 FF FE A5 2F 33 34 00 00 00 00 00 00
 8010:  12 11 34 87 FE 44 00 00 3E 4C 00 20 A9 0F C5 00
 etc.
```

329

2. COMMAND STRUCTURE

Each command consists of one or more characters and modifiers in command groups. Each group is separated by one or more spaces. The command line is one or more command groups. There is enough space in the command buffer for 255 characters.

Multiple command lines are possible by use of the Macro Add command.

2.1 COMMAND PROMPT

The program prompt consists of the following two lines:

```
S1 D1
COMMAND (B,C,D,F,G,H,I,L,M,N,O,P,Q,R,S,T,V,W,X,+,-,/,   ?) =>
```

The first line contains two flags, S1 D1, that tell which slot and drive the program is set to test. Two other flags can also appear on this line to indicate when pause on error is active and when the Macro table is full. They are identified by the letters P (pause on error active) and M (macro table full) on a white background.

The second line of the prompt consists of the first letter of each valid command. A list of the commands is always available by typing H or ? and pressing RETURN. In this manual you will see this written as H<nl> or ?<nl>. (The <nl> is the symbol for newline, often called RETURN.)

NOTE

After typing a command and pressing RETURN to execute it, you can press ESCAPE to terminate the command . However, if you wish to review the command that you just executed for the purpose of changing it, etc., you can type CONTROL A instead of ESCAPE.

380

2.2 DETAILED COMMAND DESCRIPTIONS

Following are descriptions of some of the commands that require more detailed explanations than others. In depth examples of some of the more commonly used commands such as Read, Write, Increment, and Display were provided in section 1.2. For an overall summary of all commands, refer to section 3.

2.2.1 BUFFER FILL

Syntax B[num] ---- where num is an optional 16 bit hex number
(eg.,B0123)

The buffer fill command is used to fill the output buffer with a specific data pattern. It can be used with either the Formatter/Debugger ROM or the Standard ROM installed in the drive. To use the command, type B followed by a number which will be treated as a 16 bit (2 byte) pattern. For example, the command

R0 B1234 W0 B45 W1<nl>

would read logical block-0, fill the output buffer with 12341234..., write block-0, fill the output buffer with 00450045..., and finally write this data to block-1.

2.2.2 CREATE SPARE

Syntax C<nl> ---- the C-command must be the first command on the line.

The create spare command is used to force a peripheral device to transfer a logical block to a new (spare) physical location on the device. The command will then request the block number to be spared and will then request confirmation from the user. If confirmed, the logical block will be spared by the device controller. This command

should only be used when the Standard ROM is installed in the drive.

Using this command will allow you to fix flaky blocks found during testing.

The user will be asked for confirmation of this command. Respond with Y (Yes) or N (no).

NOTE

Do not attempt to use the create spare command when the Formatter/Debugger ROM, version D3.11 or earlier, is installed in your drive.

381

2.2.3 DISPLAY

Syntax D[I,O,S] ---- where I is input buffer, O is output buffer, and S is the extended status information in the input buffer.

The display command can be used with either the Formatter/Debugger ROM or the Standard ROM installed in the drive. It is used to display the contents of the input or output buffers. These buffers occupy the following Hex locations in the computer's memory:

input buffer ----8000H - 8213H
output buffer ---8300H - 8523H

The display will show 256 bytes at a time, scrolling to the next 'page' after each key press. To end the display and continue executing the command line, press RETURN. To abort the command line press ESCAPE. With the display on the screen, you can press the spacebar to cause a scroll to the first line of the next 'page'. Then, each time you press the space bar, the display will scroll one line forward.

The Display Status command works only slightly differently in that it decodes the input buffer into Extended status info about the spares and bad blocks and it will not stop until it is finished or until the space bar, RETURN key, or the ESCAPE key are pressed.

REMEMBER! The Display Status command will decode anything found in the input buffer so be sure to use the Get Status command first to make sure that the information is valid.

Before using the Display Status command with the Formatter/Debugger ROM installed, you must initialize the spare tables (see section 2.2.6) and then use the Get Status command, otherwise the display will contain garbage. With the Standard ROM installed, you only have to issue the Get Status command before displaying the extended status of the input buffer.

Here are some additional tips about using the Display Status command.

- After typing D[S] and pressing RETURN, press the space bar to stop the listing. Press the space bar again each time you wish to step through another line of the listing. Pressing any other key will cause a fast scan of the listing.

- Pressing ESCAPE will cancel the rest of the display function.

- Pressing RETURN will terminate the spare sector listing and start the bad block listing.

382

2.2.4 FORMAT

Syntax F<nl> ---- this command must be the first command in the command line.

The Format command is used with the Formatter/Debugger ROM installed in the drive. It is used to erase all old data from memory and lay down a new pattern of address and data.

After formatting a drive with the Formatter/Debugger ROM installed, you can type D[I] to get a list of the defective blocks. The list will end with FF FF FF FF. Refer to the documentation provided for the F/D ROM for further details.

WARNING

This command is very dangerous and should only be used if damage to the address headers has occurred and only after every reasonable attempt has been made to recover other data from the device. The user will be asked to confirm this command. Respond with Y (Yes) or N (No).

Remember! This command will erase all previously recorded data.

Following is an example of the error message that will be displayed if you attempt to format a drive with a Standard ROM installed.

			Block	I/O R1	R2	R3	R4	ProFile S1	S2	S3	S4	S5	S6
FORMAT	S1	D1	00 03 5F	00	00	00	00	55	55	55	55	00	00

The 55's in this example are the error indicators.

2.2.5 GET STATUS

Syntax G<nl>

The Get Status command is used to obtain extended status information from the drive and place it in the input buffer. (You will need to execute the Display command to view the contents.) The Get Status command makes sure that the status you are going to view is valid.

When using a ProFile you can also get the status by reading block FFFFFF.

3863

2.2.6 INITIALIZE SPARE TABLE

Syntax I<nl> ---- this command must be the first command in the command line.

The Initialize command is used by the ProFile Formatter/Debugger ROM to setup the spare tables in the ProFile drive. After the Debugger ROM has formatted the disk, the entire disk is available to read from or write on so that certification of the entire disk is possible. After the spare table sectors have been certified, the tables need to be initialized to allow the controller ROM to work properly. The Initialize command is not used when the Standard ROM is installed in the drive.

WARNING

This command is potentially dangerous as it effectively erases any old spared table data and if used incorrectly will cause the loss of valuable data on the ProFile.

The user will be asked to confirm this command. In response, type Y (Yes) or N (No).

384

3. COMMAND SUMMARY

Following is a summary of all the commands that can be used with this program:

NOTE

Data shown in [] or () is optional.

BUFFER FILL B[<nl>] or B[num]

Fill the output buffer.

where num is a 16 bit hex fill number

used with both Formatter/Debugger and Standard ROMs

CREATE SPARE C<nl>

Force the drive to spare the current block.

must be the first command on the command line

used only with Standard ROM

DISPLAY D[<nl>] or D[I] or D[O] or D[S]

Display the I/O buffers.

used with both Formatter/Debugger and Standard ROMs

FORMAT F<nl>

Format the device.

this is a dangerous command

used only with Formatter/Debugger ROM

386

GET STATUS G[<nl>]

Get extended status information.

Remember! When using the ProFile, RFFFFFF (read block FFFFFF) also returns status information

used with both the Formatter/Debugger and Standard ROMs

HELP H[<nl>] or H[E] or H[E2] or H[char]

Print a list of commands, or errors, or a detailed description of each command.

where E is to display errors when the Standard ROM is installed

where E2 is to display errors when the Formatter/Debugger ROMis installed

where char is any legal command shown in this command summary

INITIALIZE SPARE TABLE I<nl>

Clear the spare block table.

this is a dangerous command

must be the first command on the command line

used only with Formatter/Debugger ROM

requires confirmation (Y or N)

LOOPON LPF[S,H,D]

Send Loop on Format commands to the firmware.

where S = sector marks
H = address headers
D = data field

LPF can be followed by any combination of S, H, and D

executed following a Read Track command to loop on format of the current track (eg., R[T] reads the T byte of the block variable to determine the current track)

used only with Formatter/Debugger ROM

MACRO M[A(0-9),C,L,0-9]

Use macro functions (alternate command lines).

where A = add current line to macro table
C = clear all macros
L = list macros

used with either Formatter/Debugger or Standard ROM

N,O

Not implemented yet

PAUSE P[<nl>] or P[A(num),C,,E(num),N(num)]

Pause and wait for user.

Where A = any error/nonerror
C = clear any error/nonerror
E = on error
N = on no error
(num) is a 16-bit mask of S1 S2

Note: All four digits of the 16-bit mask must be turned on

used with either Formatter/Debugger or Standard ROM

QUIT Q[<nl>]

Return to calling routine (This module is a subroutine).

READ R[(num),T(num),H(num),S(num)]

Read a block.

where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

Remember! The firmware of the device tested determines how the number is treated. See the following example:

	T	H	S	- Formatter/ Debugger ROM
ProFile	00	00	00	
	Hi	Mid	Lo	- Standard ROM

>0 means read to the output buffer (Eg., R23>0)

used with either Formatter/Debugger or Standard ROM

SCAN S<nl>

Order the firmware to scan the entire disk (read only).

Requires confirmation (Y or N)

after scanning, use D[I] to get list of any bad blocks found during scan

used only with Formatter/Debugger ROM

TURN OFF STEPPER T[<nl>

Turn off the power to the stepper motor.

388

V

Not implemented yet

WRITE W[(num),T(num),H(num),S(num)]

Write to a block (same format as read).

where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

Remember! The firmware of the device tested determines how the number is treated. See the following example:

	T	H	S	- Formatter/ Debugger ROM
ProFile	00	00	00	
	Hi	Mid	Lo	- Standard ROM

<I means write from the input buffer (eg., W23<I)

used with either Formatter/Debugger or Standard ROM

XECUTE X[<nl>] or X[E(num)]

Execute the current command line again.

where E = execute the line to this point on error
num = 16 bit error mask of S1 S2

X[E] will execute a function over and over again, if an error has occurred, up to the point where the error occurred

```
+     +[(num),T(num),H(num),S(num)]

        Increment the block number.

                where (num) = 24 bit number (8 bit if T,H,S)
                          T = first (Hi) byte of block
                          H = second (Mid) byte of block
                          S = third (Lo) byte of block


-     -[(num),T(num),H(num),S(num)]

        Decrement the block number.

                where (num) = 24 bit number (8 bit if T,H,S)
                          T = first (Hi) byte of block
                          H = second (Mid) byte of block
                          S = third (Lo) byte of block


/     /[C,D(num),M,O,S(num),T]

        Choose options  (/H for help).

                where C = compare input to output buffer
                      D = device set
                      M = move input to output buffer
                      O = option flag setup
                      S = slot set
                      T = translate current block to cylinder/head/sector


?

        Help.
```

390

4. NEW FEATURES

The following new features have been developed since the release of DSKDBG V-E00.16 and are included in DSKDBG V-E00.17.

* The Help errors command has been slightly modified. HE or HE1 provides help for the Standard ROM and HE2 returns help for the Formatter/Debugger ROM.

* The Help command lists the form of the extended help commands.

* Display now has a command to allow the decoding of the status table into the version, spare list, and bad block list.

DS = display status info.

* Use ESCAPE or RETURN to terminate a long list of spares or bad blocks.

* The /C command allows you to compare the input buffer to the output buffer.

* The /M command moves the contents of the input buffer to the output buffer.

* An additional modifier is available for read/write commands that allows you to specify which buffer you are reading to or writing from. For example, a read command followed by >0 means read to the output buffer. A write command followed by <I means write from the input buffer.

391

5. SPECIAL DATA RECOVERY FUNCTIONS OF THE XECUTE (X) COMMAND.

The X command has a special function that allows primitive loop control. This function was added (V-E00.14) to allow very basic data recovery for profile. It performs a simple loop back to the beginning of the command line on and error condition occurring in either status bytes 1 or 2 of the profile status. Activating this function is the same as activating Pause on error.

Syntax X[<nl>] or X[E]

where X is always the last command executed or XE is the last command executed if an error occurred

5.1 USING THE X COMMAND WITH THE STANDARD ROM

Following is an example of using the X command when the Standard ROM is installed in the drive:

```
R0<nl>     Read Block 0
/S1 / R>0 XE /S4 / W<0 XE + X
```

In this example the inital R0 followed by RETURN sets up the program to start reading on block 0. In the next command line, S1 selects slot 1 as the input drive (drive to read from).

REMEMBER! Don't forget the space after the 1 as it is required. You must include a space whenever a blank area is indicated as shown in the above command line example.

The R command reads the block (0 at first). If an error occurs during read (indicated in status bytes 1 and 2) then the XE command will cause control to transfer back to the beginning of the line to S1 and the program tries again. If the read is ok, the program goes on to S4 which switches to slot 4. Then it writes to the current block. (notice that the program hasn't changed the block number yet.) Another check of the status is made by the XE command. The program will continue on and if no error occurs on the write it will go back to the beginning of the line and try again. If no error is detected, the + command causes the program to increment the block counter. Finally, the X command at the end of the command line returns the program to the beginning of the line where it goes back to slot 1 and reads the next block.

The program will not stop at the end of the disk but it will not harm anything if it goes passed the end. The current block number being read/written will be greater then 25FF and the user can stop the program by pressing the escape key.

5.2 USING THE X COMMAND WITH THE FORMATTER/DEBUGGER ROM

If the Standard ROM can't be used and the Formatter/Debugger ROM is required then the the command takes on one minor change. Instead of incrementing the 24-bit number with the + command, you increment the 8-bit S variable as shown in the following example:

```
/S1 / R>0 XE /S4 / W<0 XE + S
```

This command works the same except we now increment track (cylinder), head, and sector which is the form expected by the Formatter/Debugger. As with the previous command the program will not stop at the end of the disk. It will, however, wrap around to T0 H0 S0 and start through again so you have to watch and stop the program at this point.

393

6. SAMPLE SCREEN DISPLAY

Here is a sample readout from the screen. Remember that this display moves very fast and you may not see it clearly. Also Status S5 and S6 are for Twiggy and are set to 0 by the program when using Profile.IO.

			Block	I/O R1	R2	R3	R4	Profile S1	S2	S3	S4	S5	S6
READ	S1	D1	00 00 01	00	00	00	00	00	00	00	00	00	00
WRITE	S4	D1	00 00 01	00	00	00	00	00	00	00	00	00	00
READ	S1	D1	00 00 02	00	00	00	00	00	00	00	00	00	00
WRITE	S4	D1	00 00 02	00	00	00	00	00	00	00	00	00	00
READ	S1	D1	00 00 03	00	00	00	00	00	00	00	00	00	00
WRITE	S4	D1	00 00 03	00	00	00	00	00	00	00	00	00	00
READ	S1	D1	00 00 04	00	00	00	00	08	00	00	00	00	00
READ	S1	D1	00 00 04	00	00	00	00	00	80	00	00	00	00
READ	S1	D1	00 00 04	00	00	00	00	00	00	00	00	00	00
WRITE	S4	D1	00 00 04	00	00	00	00	00	00	00	00	00	00
READ	S1	D0	00 00 05	00	00	00	00	00	00	00	00	00	00
WRITE	S4	D0	00 00 05	00	00	00	00	00	00	00	00	00	00

394

7. TYPICAL PROBLEMS AND HOW TO AVOID THEM.

* The + (Increment) command doesn't work.

This is to date the most common problem. Usually the command line will look something like

```
R0 PE + X
```

What you were trying to do is sequentially read thru the disk starting at block 0. What you said was

```
Read block 0 (block = 0), Pause on error,
Increment block (block = 1), Repeat line,
Read block 0 (block again = 0) ........
```

To fix this problem just do one line to set block 0 and then go on to the next line to do the Read, Pause on error etc. For example:

```
R0<nl>
R PE + X<nl>
```

* Display Status Won't Work Or Won't Stop

The display status command allows you to display the input buffer and make assumptions about what the data means. For it to work, the Get status command must successfully read the status informtion from the drive. If the input buffer has garbage in it, the Display Status command may go round and round trying to display all the spared sectors and bad blocks.

To stop the command when it is caught in a loop or is listing a bunch of bad blocks, simply press the RETURN key or the ESCAPE key. The functions of the other keys are the same as for all other display commands.

395

8. KNOWN BUGS IN VERSIONS E00.17 AND E00.16

The Macro command has 3 known bugs. The first occurs when a macro is added to the list and there is already one there of that number. The effect is to make both of them disappear. The second bug occurs when the macro is invoked on a line with the X command (eg., M1 X). This disables the ESCAPE key so the program can only be halted by re-booting. The third bug also involves the X command. When a macro is added (eg., R MA1 X), it will keep asking to delete the old macro.

Pause on any error doesn't work.

The - command doesn't decrement properly for H and S modifiers

396

9. STATUS BYTE DESCRIPTIONS

This section provides a bit-by-bit description of the four status bytes available with the Standard ROM or the Formatter/Debugger ROM.

9.1 STATUS BYTES WITH STANDARD ROM V-3.98 INSTALLED

STATUS 1

7 = 1 if ProFile did not receive 55 to its last response
6 = 1 if write or write/verify was aborted because >532 bytes if data were sent
if 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/A
0 = 1 if operation unsuccessful

STATUS 2

0=0 1=1 0=2 1=3 0=4 0=5 0=6 1=7 0=8 1=9 0=A 1=B 0=C 0=D 0=E 1=F

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/A
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/A

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/A
3 = N/A
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 SOS ProFile driver and are not used by the Dskdbg ProFile.IO.

9.2 STATUS BYTES WITH FORMATTER/DEBUGGER ROM FD3.98 REV 11 INSTALLED

STATUS 1

7 = 1 if Profile did not receive 55 to its last response
6 = 1 if no index found during formatting
5 = 1 if no sector mark found during formatting
4 = 1 if SEEK ERROR - unable to read 3 consecutive headers on a track (one try only)
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/A
0 = 1 compare error on a write compare

STATUS 2

7 = 1 if ProFile has been reset
6 = 1 if track number invalid while reading or writing a sector
5 = N/A
4 = N/A
3 = N/A
2 = N/A
1 = 1 if seek to wrong track occurred
0 = N/A

398

STATUS 3

D7, D6, and D5, along with D4 and D3 from STATUS 1, tell why a write compare operation failed.

	D7	D6	D5
write timeout	0	0	0
read timeout	1	0	0
read CRC	1	1	0
data compare	1	1	1

STATUS 4

7 - 0 = the number of errors encountered when formatting data fields or scanning the disk.

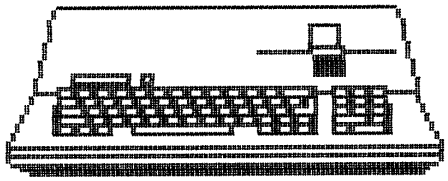
FINIS

399

7

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

How to Use the Rev. 18 Debugger to Copy ProFiles

Author: Apple

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

How To Use The REV. 18 Debugger To Copy Profiles

400

TO COPY THE CONTENTS OF ONE PROFILE TO ANOTHER THE FOLLOWING COMMAND LINE CAN BE USED WITH EITHER THE SYSTEM PROM OR THE DEBUG PROM. HOWEVER THERE IS A SLIGHT DIFFERENCE SO BOTH EXAMPLES WILL BE SHOWN.

THE EXAMPLES ASSUME YOU HAVE THE MASTER DRIVE IN SLOT1 AND THE SLAVE DRIVE IN SLOT3. YOU CAN USE ANY SLOT YOU WANT.

WITH THE 3.98 SYSTEM PROM AND THE REV. 18 DEBUGGER THE COMMAND LINE WOULD LOOK LIKE THIS

/S1 (RETURN) R0000 (RETURN)
/S3 (RETURN) R0000 (RETURN)

/S1–/–R–XE–/S3–/–W<I–XE–+–X(RETURN)

THE (–) INDICATES A SPACE MUST BE INSERTED HERE.

WITH THE REV. 11 DEBUGGER PROM AND THE REV. 18 DEBUGGER SOFTWARE, THE COMMAND LINE WOULD LOOK LIKE THIS.

/S1 (RETURN) RT0H0S0 (RETURN)
/S3 (RETURN) RT0H0S0 (RETURN)

/S1–/–R–XE–/S3–/–W<I–XE–+S–X(RETURN)

AGAIN THE (–) INDICATES A SPACE MUST BE INSERTED.

1-1

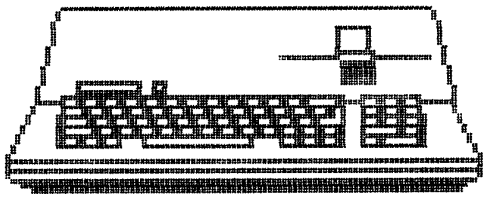
FINIS

401

8

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Production Format Test Procedure with Apple III

Author: Apple

18 May 1982

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

402

PROFILE PRODUCTION FORMAT TEST PROCEDURE WITH AIII

9/82 ry
---4/6/82 ry
B---4/19/82 ry
REV C---4/21/82 ry
REV D---5/7/82 ry
REV E---5/10/82 ry
REV F---5/18/82 ry

Pages: 6

1.0. OBJECTIVE:

The objective of this procedure is to format the ProFile disk drive and reject improperly formatted HDA assemblies.

2.0. EQUIPMENT NECESSARY:

2.1. ProFile format test fixture
2.2. AIII/MIII
2.3. Shorting plug or jumper switch.
2.4. Format EPROM---FD3.98
2.5. ProFile Format/Certify disk---~~E00.11-5~~ E00.12

3.0. TEST TIME/UNIT: 4.5 minutes

4.0. PRELIMINARY TEST:

4.1. Allow drive to warm up (>1min) at the warm-up station.
4.2. Adjust index to 16.667ms (+/-) 0.5%. The limits are 16.584ms to 16.750ms.

5.0. TEST PROCEDURE:

5.1. Insure that the DC power on the test fixture is OFF!
5.2. Place drive vertically onto the test fixture and connect plugs P6, P7, and P8. Now, set drive horizontally and connect DC power plug and head cable (P5).
5.3. Turn on AC power to the ProFile.
5.4. Insert the ProFile Format/Certify Production disk into the AIII. After a delay of 20 seconds, press <RTN> and the ProFile's READY LED should turn on.
5.5. The monitor should prompt "PRESS "RETURN" WHEN PROFILE IS READY". Type '<RTN>' and the monitor should prompt "INSTALL JUMPER, PRESS ANY KEY TO CONTINUE". Install jumper, or place JUMPER SWITCH on INSTALL position and type '<RTN>'. The ProFile should begin the Format cycle. After the ProFile has completed the Format cycle successfully, the monitor should prompt "REMOVE JUMPER, PRESS ANY KEY TO CONTINUE". Remove jumper, or place JUMPER SWITCH on REMOVE position and type '<RTN>'. The ProFile should now execute the following commands automatically:
5.5.1. Scan all sectors, heads, and tracks
5.5.2. Compare buffers on TRK077

1-6

403

5.5.3. Certify all sectors on TRK077
5.5.4. Initialize spare tables on TRK077.

5.6. The AIII will ring its bell when the test has been completed or failed. Be sure to turn OFF AC power to the ProFile test fixture here.

5.6.1. A good HDA unit will pass each of the above commands and should be sent to the certification test station.

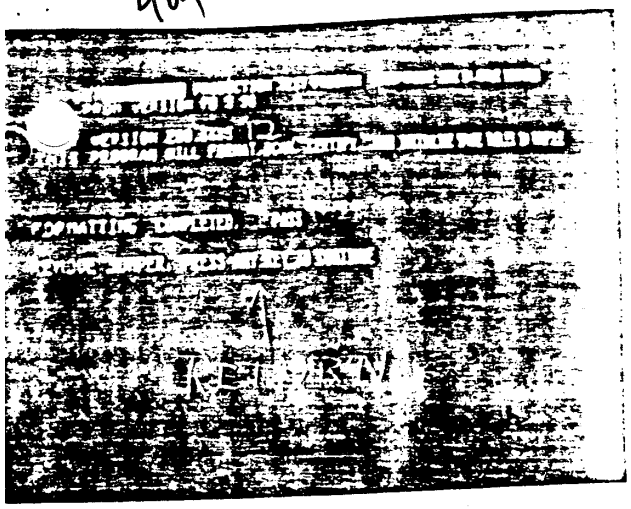
5.6.2. Any failure to any of the commands will halt the automatic sequence, indicate fault, and halt test progress. If a failure occurs, identify drive with problem mode, and set HDA aside as failure.

5.7. For consecutive format test runs, do not power the AIII off. Instead, simply connect the next HDA unit, turn ON the ProFile test station, and wait for 20 seconds until the ready light turns on. Return to step 5.5 and repeat test procedure.

-----end of test-----

2-6

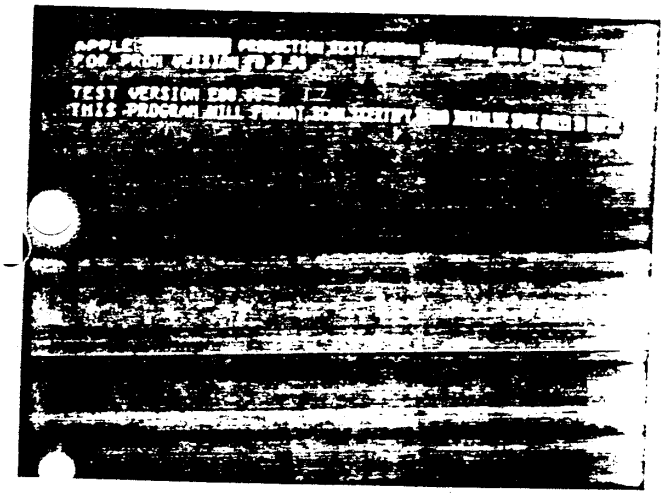
404



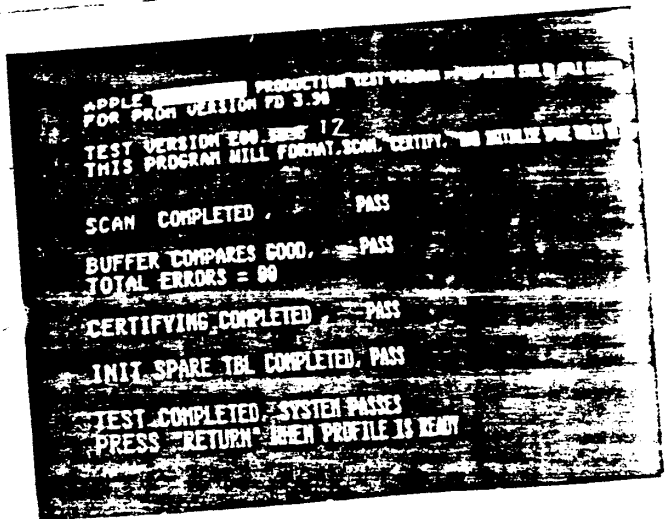
AFTER FORMATTING IS COMPLETE SCREEN WILL LOOK LIKE THIS

NOTE: THE OPERATOR MUST INSURE THAT THE JUMPER IS REMOVED.

PRESS RETURN



WHILE SCAN OPERATION IS BEING DONE THE SCREEN WILL LOOK LIKE THIS

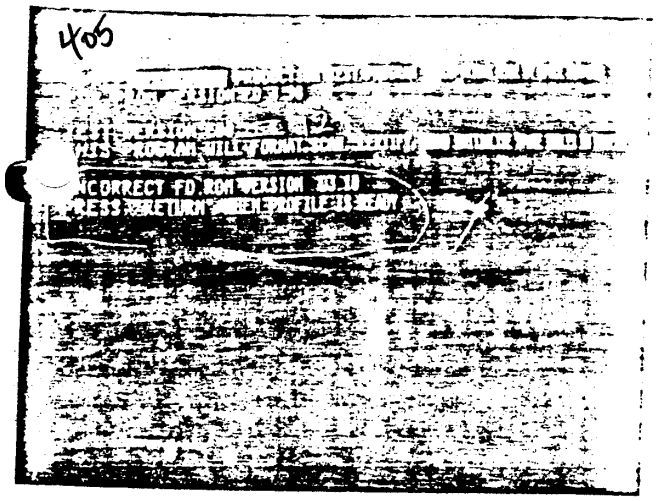


IF EVERYTHING IS OK THE SCREEN WILL END UP LIKE THIS

AND A PRINT OUT WILL BE GENERATED BY THE SILENTYPE.

REMOVE HDA DRIVE + INST. NEW ONE.

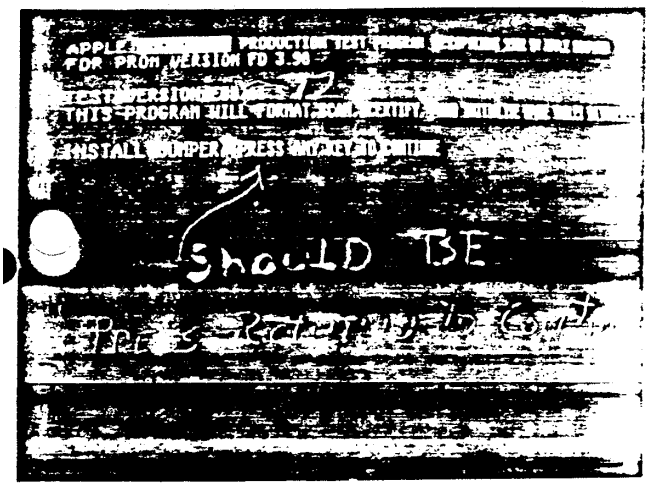
3-6



Correct Sequence

System Booted UP WITH ProFile connected AND READY

Press Return



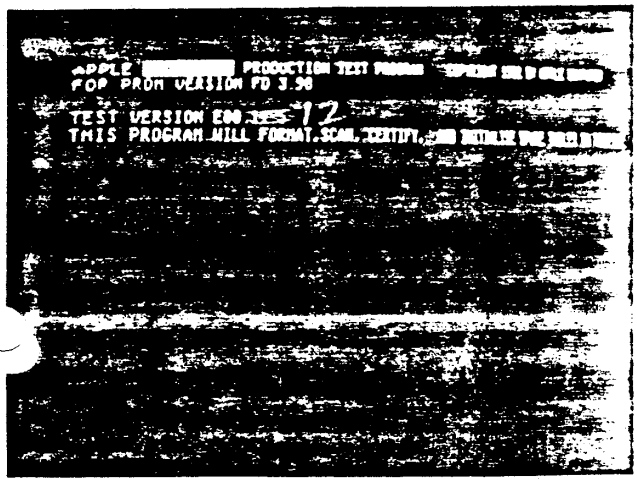
After you Press Return you will See This

NOTE: The OPERATOR MUST INSURE THAT The JUMPER IS INSTALLED

INSTALL JUMPER

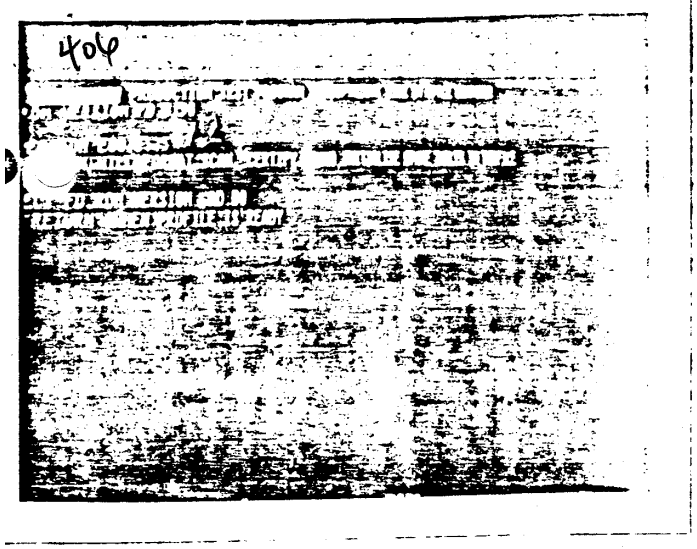
↓

Press Return

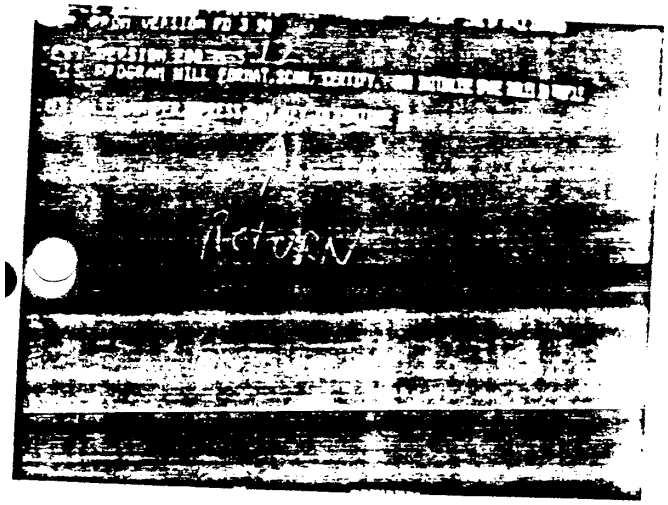


Screen will Look Like this while Formatting

4-6



IF PROFILE IS NOT CONNECTED



←BOOT

↓

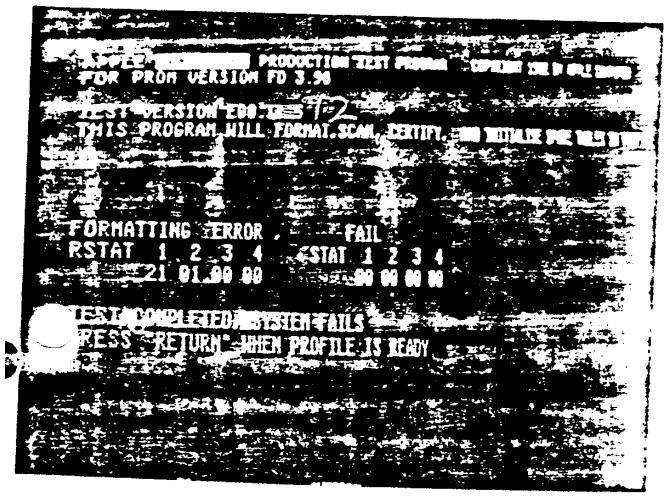
PRESS RETURN

↓

INSTALL JUMPER

↓

PRESS RETURN

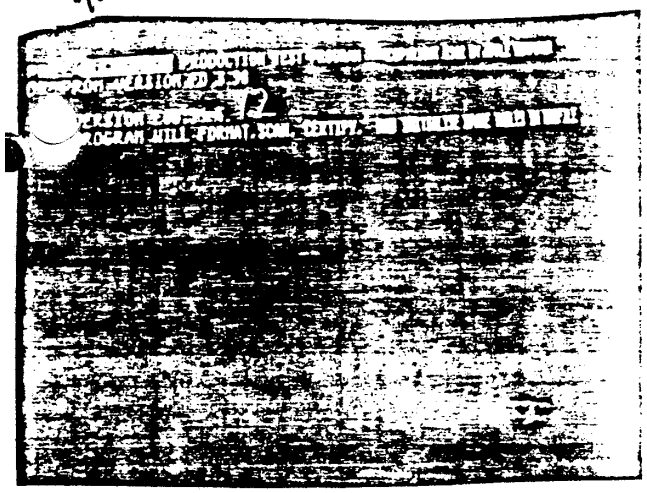


↓

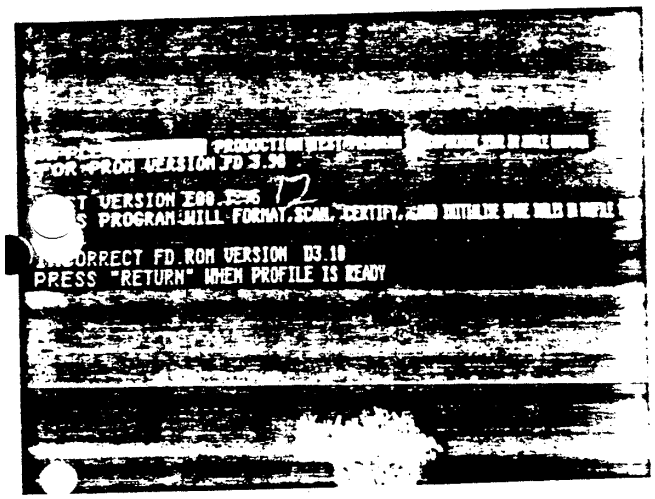
FAILURE STATUS

5-6

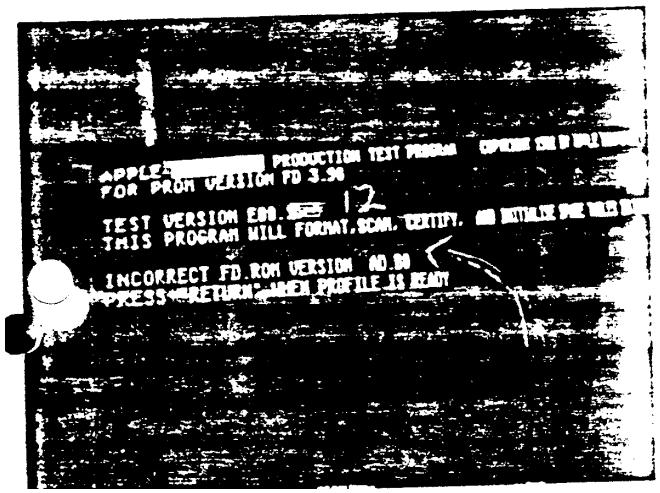
407



1) IF PROFILE IS Connected BUT NOT TURNED ON OR NOT READY The Screen DISPLAY WILL LOOK LIKE This on Boot UP To Get INTO The FORMATTING MODE you MUST REBOOT the DISKETTE



2) IF The PROFILE IS Connected AND READY you will See This on Boot UP



3) IF the PROFILE IS NOT Connected you will See This on Boot UP

6-6

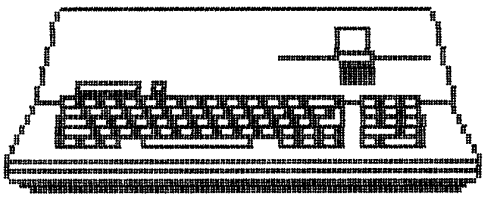
FINIS

408

9

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///

Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Converting 5M ProFile to 10M

Author: ?

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Apple /// ProFile HD Converting 5M HD to 10M HD (Pages: 4)

U5 MUST BE 10114

409

10 MHz Hybrid Oscillator

REMOVE U40 AND SOCKET

REMOVE CRYSTAL

REMOVE R15 OR PUSH LEFT

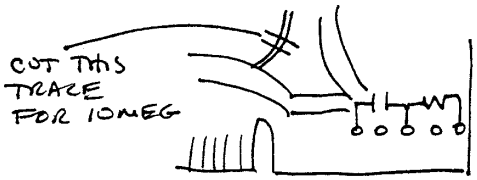
PUSH C35 RIGHT

REPLACE WITH 10.0 MHz - 14 PIN PROFILE

+5V RESET LINE - POWER CONNECTOR - BYPASS TO GND .1NF

ADD .1NF P3-3 TO GND (P.N5)

ADD 5.1K Ω FROM P3-3 TO P3-1 +5V 5%

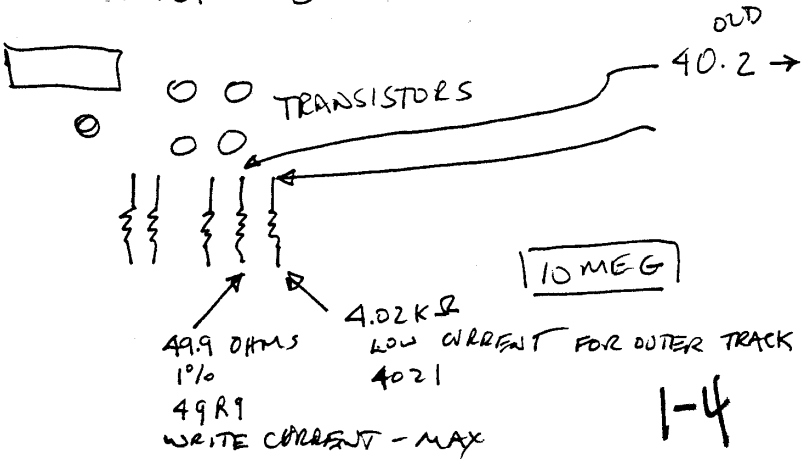


CHANGE TERMINATING RESISTORS FROM 47 Ω (or 330 or 47Ω)

NEED -101 INSTEAD - 470

ANALOG CARD

10131 — SHOULD BE ~~A0101RC0E-A~~



PULL UP RESISTORS ON BACK

2.2K - 4.7K

PULL UP TTL INPUT

1-4

410

PROFILE I/O CARD

RESISTOR NETWORKS SHOULD BE 100 Ω

341-0035 - KR3600

-

4K ZB - TAKES 2732 4KX8 10 MB

- 1 EPROM DIAG
- 1 EPROM -

STANDARD RUNNING SOFTWARE 3.9B

ZB ASSEMBLER — SEE JOHN

2-4

411

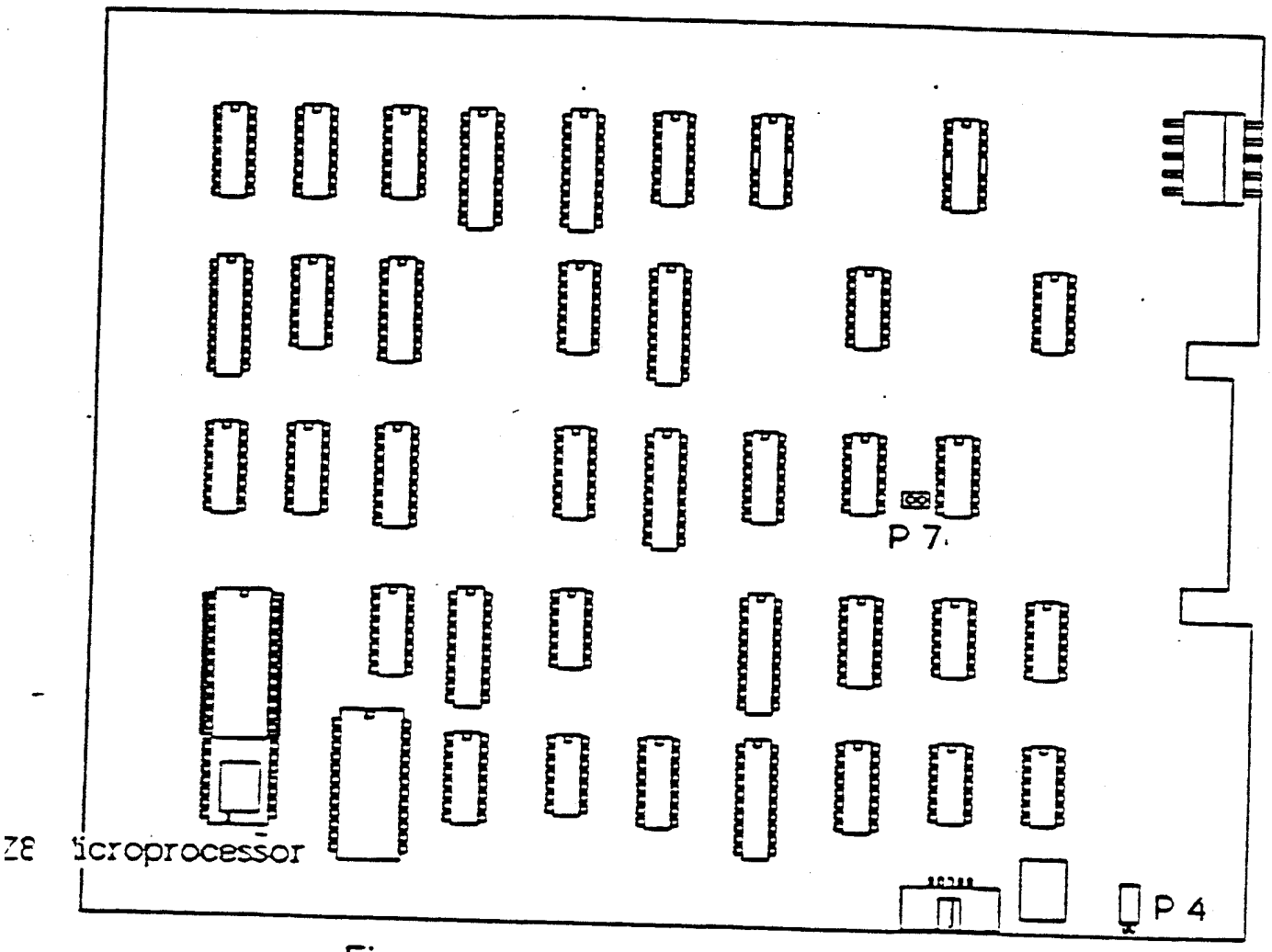


Figure 1 - Profile Controller Board

3-4

4/2

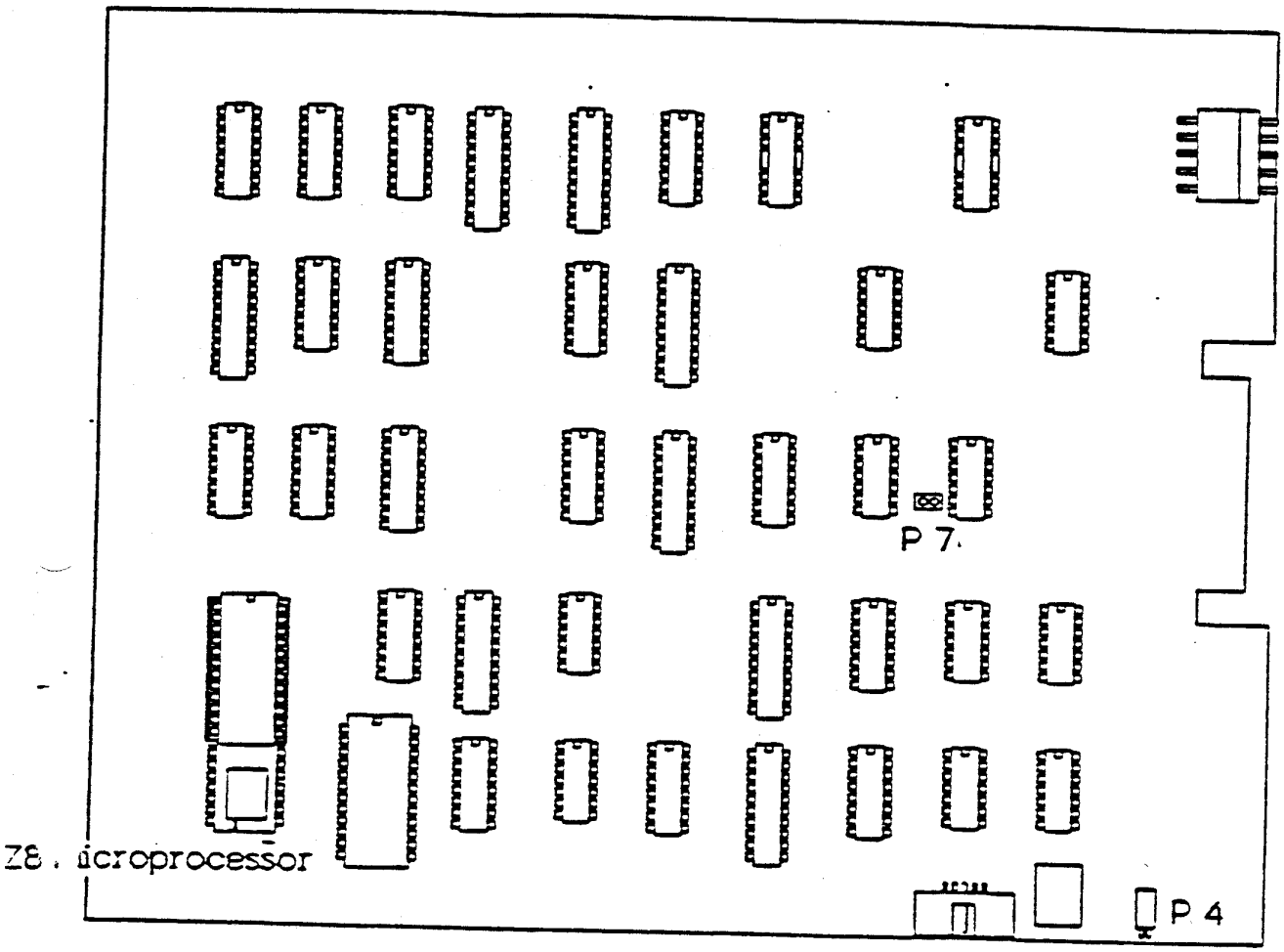


Figure 1 - Profile Controller Board

4-4

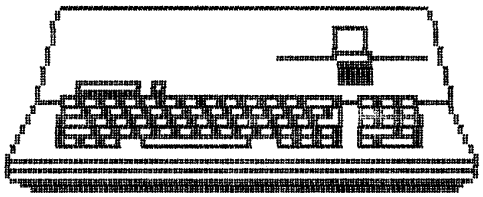
FINIS

413

10

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Certification and Failure Criteria

Author: Apple

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

414 Apple /// ProFile HD Pages: 2

CERTIFY CRITERIA

IMMEDIATE FAILURES

1	- 61 00 00 00	-	3 BYTE SEPERATOR BETWEEN WRITE BUFFER AND STATUS TABLES BAD OR S2-3 SET. SPARE TABLES UPDATE OCCURED, BUFFER DATA IS CHANGED. OPERATION FAILED.
1	- 00 02 00 00	-	SEEK TO THE WRONG TRACK OR PLUGS 6 & 8 ON ANALOG BOARD ARE SWITCHED AROUND.

MAXIMUM ERRORS	CODE S1 S2 S3 S4		DATA
10 - SOFT -	08 00 00 (00...09)	-	READ ERROR. (00...09) PERCENTAGE OF READ FAILURES AFTER READ COMMAND.
2 - HARD -	08 00 00 (10...)	-	READ ERROR. (10...) PERCENTAGE OF READ FAILURES AFTER READ COMMAND.
2 -	09 00 00 00	-	READ ERROR. OPERATION FAILED.
2 -	05 00 00 00	-	CANNOT FIND TARGET HEADER IN 9 ROTATIONS. OPERATION FAILED.
2 -	19 80 00 00	-	CAN'T READ 3 SECTORS AFTER 2 RESEEKS. READ ERROR. OPERATION FAILED. CAN'T READ 3 SECTORS AFTER SEEK.
2 -	11 80 00 00	-	CAN'T READ 3 SECTORS AFTER 2 RESEEKS.OPERATION FAILED.CAN'T READ 3 SECTORS AFTER SEEK.

* IN ALL CASES WHEN MAXIMUM ERRORS ARE REACHED, FAIL UNIT AND SEND TO REWORK.

1-2

Tom Craig

415 Failure Criteria

1. MORE THAN 15 SPARE SECTORS
2. MORE THAN 100 SOFT ERRORS
3. MORE THAN 20 HARD ERRORS
4. MORE THAN 9 02 ERRORS (SEEK TO WRONG TRACK)
5. MORE THAN 1 BLOCK ID MISMATCH
   OR BUFFER COMPARE FAILURE
   OR FF-FF-FF ERRORS.
6. MORE THAN 2 09 ERRORS

Timeout - profile tries to talk to computer and computer is out to Lunch

2-2

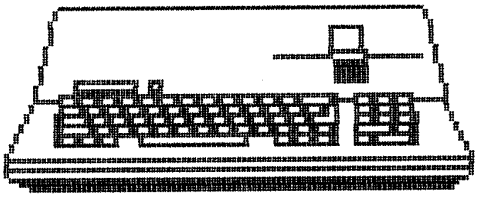
FINIS

416

II

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Debug Data Status Byte Definitions

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

STATUS BYTE DEFINITIONS:

PROFILE DEBUG DATA
STATUS BYTE DEFINITIONS

Pages: 3

STATUS #1: 417

byte number	definition
7	other than $55 responce from host
6	3 byte seperator between write buffer and status tables bad or S2-3 set.
5	spare tables update occured, buffer data is changed
4	cannot read 3 sectors after 2 reseeks
3	read error
2	cannot find target header in 9 rotations
1	
0	operation failed

STATUS #2

byte number	definition
7	cannot read 3 sectors after seek
6	more than 32 spares
5	
4	more than 100 bad blocks
3	cannot read status sectors
2	sparing occured
1	seek to the wrong track

STATUS #3

byte number	definition
7	profile was reset
6	requested block # out of range
5	block I.D. mismatch set by host
4	
3	
2	host reset profile
1	bad command responce from profile
0	parity error on cable

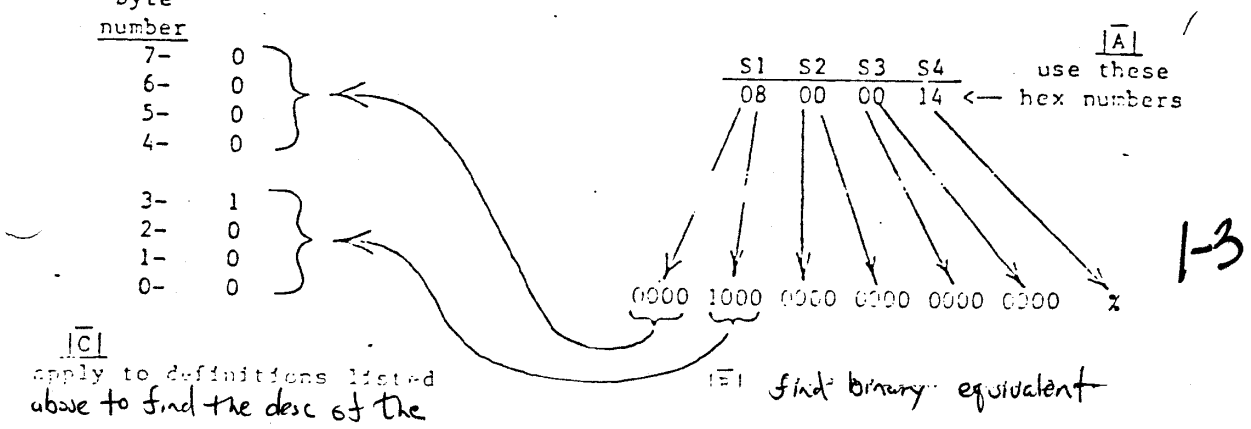
STATUS #4

percentage of read failures after read command

Conversion Chart

Hex	Binary	Dec.
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Status bit identification example:



1-3

416

ERROR SYMPTOM CODES

0002 00 00	2 - Seek to wrong track occurred
0080 00 00	8 - Unable to read/write status sectors
0082 00 00	8 - Unable to read/write status sector 2 - Seek to wrong track occurred
0400 00 00	4 - Time out error
0404 00 00	4 - Time out error 4 - Sparing occurred
0408 00 00	4 - Time out error 8 - Unable to read/write status sectors
0500 00 00	5 - Unable to read/write data (data is bad) Time out error, couldn't find header in 3 revolutions
0580 00 00	5 - Time out error, could find header in 3 revolutions-unable to read/write data (data is bad) 8 - Unable to read/write status sectors
0804 00 00	8 - CRC error 4 - Sparing occurred
0900 00 00	9 - CRC error-unable to read/write data (data is bad)
0580 00 00	1 - Unable to read/write data (data is bad) 5 - Time our error (couldn't find header in 3 revolutions) 8 - Unable to read/write status sectors
0882 00 00	1 - Unable to read/write (data is bad) 8 - CRC error 8 - Unable to read/write status sectors 2 - Seek to wrong track occurred

2-3

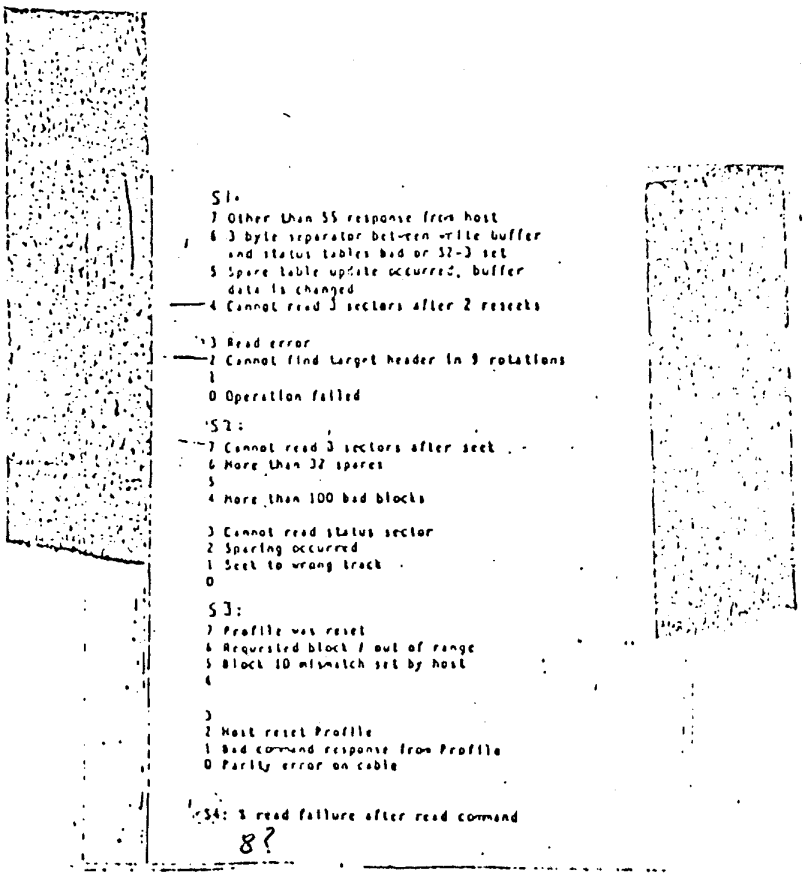
#19

60 00 00 00

6 - Time out error (couldn't find header in 3 revolutions)
invalid data, requested block is in bad block table

1D 80 00 00

1 - Unable to read/write data (data is bad)
D - Time out error (couldn't find head)
CRC error occurred
8 - Unable to read/write status sectors



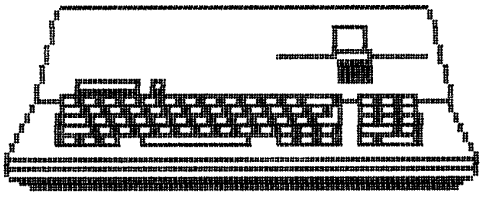
3-3
FIXITS

420

12

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///

Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Power Supply (class B) Specification

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

REV.	ZONE	ECO #	REVISION	APPD	DATE
A		P662	INITIAL RELEASE	[illegible]	11-21

421

Pages: 7

(1 of 1, 1-6)

This assembly meets Apple Computer Specification 062-0182, and includes the equivalent circuitry for Apple assy #656-4106, "PCB, Assy, Monitor", Tested, is mechanically equivalent to #656-5104, "Subassy, Power Supply", and is purchased only from U.S. ASTEC as ASTEC part number AA11771 B

PRODUCTION RELEASE

This revision supersedes all previous versions. Please destroy all old copies.

DRAWING NUMBER 699-5009-A

SH 1 OF 1

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. DECIMALS .X ± .XX ± .XXX ± ANGLES XX.X ± FRACTIONS ± DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS.	DRAWN BY D.Haahr DATE 11/83	apple computer inc.
	CHECKED BY [illegible] DATE [illegible]	
	APPROVED BY [illegible] DATE 11/21	TITLE Purchased Assy, Domestic, Power Supply CLASS B/PROFILE ~~All/Profile~~
	RELEASED BY [illegible] DATE [illegible]	SIZE A — DRAWING NUMBER 699-5009-A
MATERIAL: ----------		
FINISH: ----------		SCALE: ------- — SHEET 1 OF 1

T ASSY

422

REV.	ZONE	ECO #	REVISION	APPD	DATE
A		P662	INITIAL RELEASE	CM	11-21

PRODUCTION RELEASE
This revision supersedes all previous versions.
Please destroy all old copies.

1.0 ELECTRICAL CHARACTERISTICS

1.1 INPUT VOLTAGE: 115 VAC ~~or 230 VAC~~
~~Selected by jumper on pcb~~
~~47 TO 63 Hz~~
48 – 62 Hz

1.2 OPERATING RANGE: 90 to 135 VAC RMS
180 to 270 VAC RMS

1.3 CONVERSION EFFICIENCY: 75% minimum acceptable; with 78% as a target in production.

1.4 DELIVERED POWER: 30 watts steady state.
55 watts starting for a minimum of 14 seconds.

1.5 OUTPUT VOLTAGES AND CURRENTS:

$Vout_1$ +12 VDC ± 6% 1.5 Adc steady state and 3.5 to 4.0 amps for 10 - 14 seconds.

$Vout_2$ +5 VDC ± 2% 2.0 Adc continuous.

$Vout_3$ -12 VDC ±6% 0.1 Adc continuous.

1.6 RIPPLE AND NOISE CONTENT; OUTPUT: 50MVP-Pon +5VDC;100MV P-P +12 VDC. 1Hz to 10 MHz.

DRAWING NUMBER 062-0182-A
SH 1 OF 6

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES DECIMALS .X ± .XX ± .XXX ± ANGLES XX.X ± FRACTIONS ± DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS	DRAWN BY D.Haahr DATE 11/83 CHECKED BY DATE 11/21 APPROVED BY DATE 11-21 RELEASED BY DATE 11/21	apple computer inc.	
		TITLE POWER SUPPLY, ~~ALL/PROFILE~~ DOMESTIC CLASS "B"	
		SIZE A	DRAWING NUMBER 062-0182-A
NEXT ASSY.	MATERIAL: ---------- FINISH: ----------	SCALE: ------	SHEET 1 OF 6

423

1.7 OPERATING TEMPERATURE: 0 to +70°C (Ambient)

1.8 STORAGE TEMPERATURE: -20 to+85°C.

1.9 PROTECTION CIRCUITS: The input must be protected by a fast blow fuse and a thermister inrush current limiter.

The +5 volts d.c. TTL voltage must be protected from over voltage output by means of an active crowbar.

All three d.c. outputs must be short circuit capable for an indefinite period.

1.9.1 HOLD UP TIME: 20 msec nominal at 30 watts load.

1.9.2 TEMPERATURE COEFFICIENT: 0.02%.

1.9.3 AC ISOLATION: to safety ground and A.C. input to output 4.5KVDC

1.9.4 OUTPUT TO SAFETY GROUND: 0.5KVDC

1.9.5 INSULATION AC TO GROUND: 50 MEG ohm nominal.

1.9.6 LEAKAGE CURRENT: 240 VAC input
IRMS = 3.5 ma RMS

1.9.7 LINE CONDUCTED EMI: FCC 20780 limits. See attached specifications.

1.9.8 SAFETY APPROVALS: UL and CSA required
VDE required after 9/81

2.0 MECHANICAL REQUIREMENTS:

The supply shall conform to attached envelope.

2.1 THERMAL: The power supply shall be capable of operating under all conditions of line and load at 0-70°C continuously.

2.2 STORAGE TEMPERATURF: -20 to +85°C.

2.3 HUMIDITY: Operating: 95% RH @ 35°C.
Storage: 95% RH @ 50°C.

2.4 VIBRATION: 10 to 500 Hz double sweep at 1 active per minute with pk-pd excursion of 1.5mm or 10g acceleration.

2.5 RANDOM DROP: 45 min. at rate of 5 RPM.

2.6 BURN IN: A minimum 24 hour burn in at low line, full DC load at 70°C is required. Vendor will burn in all units.

apple computer inc.	SIZE A	DRAWING NUMBER 062-0182-A
	SCALE	SHEET 2 OF 6

424

2.7 **SERIALIZATION:** All supplies shall have a serial number affixed and recorded so that test and failure records can be tracked throughout the life of the product.

2.8 **SAFETY REQUIREMENTS:** UL 478
UL 1201
CSA 22.2 No. 154
VDE after 9/81

2.9 INDUCTORS

No solenoidal filter inductors should be used in this product.

2.95 INPUT AND OUTPUT CONNECTOR: Molex Connector Pin Designation.

A.C. Input	D.C. Connector
1. AC Neutral	1. Reset Monitor Output
2. Key	2. Key
3. AC Line	3. -12V
	4. +12V
	5. +12V
	6. Common
	7. "
	8. "
	9. "
	10. Common
	11. +5V
	12. +5V
	13. +5V

Mating Molex Connectors:

DC P/N 09-50-3131

AC P/N 09-50-3030

The output connector will be a single, in-line connector combining the above two parts number.

apple computer inc.

SIZE	DRAWING NUMBER
A	062-0182-A
SCALE:	SHEET 3 OF 6

425

SPECIFICATION: ELECTROMAGNETIC COMPATIBLITY

This unit must pass FCC Class B limits in both conducted and radiated emissions.

1. Emissions: Applicable assemblies, subassemblies and peripheral devices shall be 6 dB below limits of (1) Federal Communications Commission (FCC) Parts 15 Sections 15.830 (radiation limit) and 15.832 (conduction limit) for units operated from 60Hz line voltage and (2) VDE 0871/6.78 section 3.2.1 (conducted) and sections 3.2.2 and 3.2.3 (radiated) for units operated from 50 Hz line voltage. Those units rated 50/60 Hz shall meet both requirements.

   The upper frequency limit for both FCC and VDE conducted emissions limits is 30 MHz. However, due to radiated emissions from the AC power cord the Apple conducted limit is extended to 60 MHz.

   For convenience, limit amplitudes, less 6 dB, are reproduced herein. (Tabulated limits are Apple Computer EMI limits; FCC and VDE limits and 6 dB higher). However, the in-effect version of FCC part 15 or VDE 0871 are the binding documents.

   RADIATED

	Frequency Range (MHz)	Field (uV/m)	Strengths (dB uV)	Distance (meters)
1. FCC Part 15	30-88	50	34	3
	88-216	75	38	3
	216-1,000	100	40	3
2. VDE 0871	0.01-30	20	26	30
	30-470	20	26	10
	470-1,000	80	38	10

   CONDUCTED

	Frequency Range (MHz)	Voltage (uV)	(dB uV)	LISN Impedance
1. FCC Part 15	0.45-60	125	42	50
2. VDE 0871/6.78	0.01-0.15	*		150
	0.15-0.50	200	46	
	0.50-60	100	40	

   *straight line from 10 kHz (3.5mV, 71 dB V) to 150 kHz (300uV, 50 dBuV)

Test and measurement equipment and procedures shall be as specified in applicable specifications. Final acceptance tests are performed with assembly or peripheral installed in system intended to be marketed with; such system to consist of the basic Apple Computer (II, III, etc.) and full memory installed and as many peripheral devices (disk drives, printer, monitors) and optional components (language card, serial card, parallel card, etc.) as possible to simulate worst-case operating conditons as closely as possible. Qualification tests with "remote exercisers", generators or other manufacturer Personal CPUs are unacceptable.

apple computer inc.

SIZE	DRAWING NUMBER
A	062-0182-A
SCALE:	SHEET 4 OF 6

4216

II. SUSCEPTIBILITY (to be determined).

Under consideration: The device shall not have uncorrectable data errors when subjected to the following field strengths or voltages - Irradiated: 0.0'-1,000 MHz, 5V/m (100% moudulated with 1 kHz square wave).

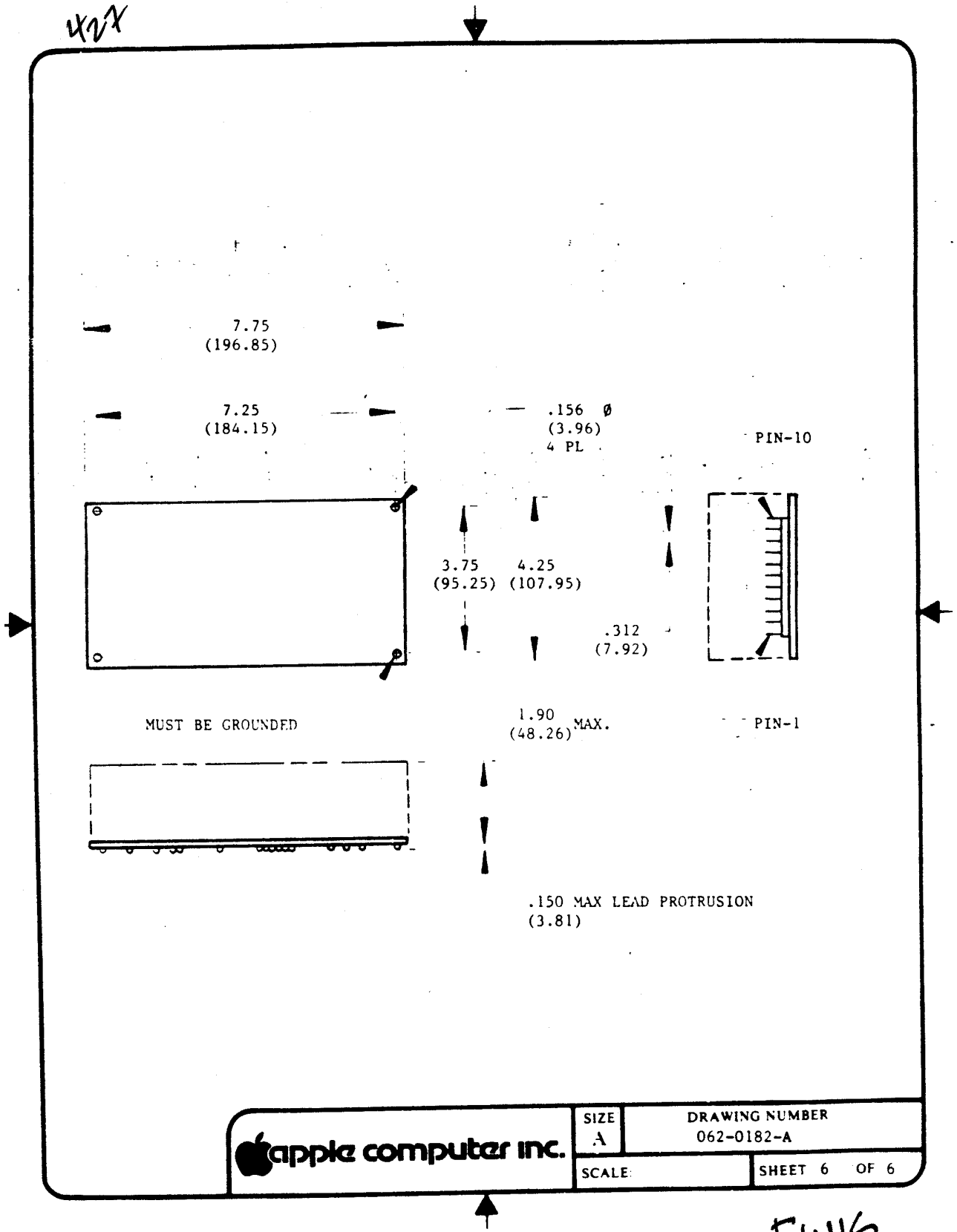
Transient Line Noise:** 1. Class A products: 400V pulse with 100 nonosecond width and 10 nsec risetime.

2. Class B products: 200V pulse (same characteristics)

Conducted RF: 0.01-100 MHz: 3V rms.

**no soft errors allowed.

apple computer inc.	SIZE A	DRAWING NUMBER 062-0182-A
	SCALE	SHEET 5 OF 6



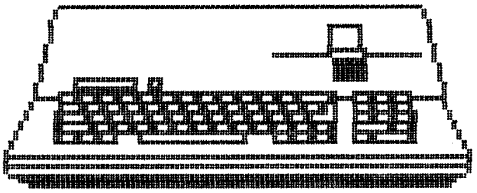
FINIS

428

13

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Power Supply (class A2, International) Specification

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 9

429

REV	ZONE	ECO #	REVISION	APPD	DATE
A		P656	INITIAL RELEASE		

International

PRODUCTION RELEASE

NOTICE OF PROPRIETARY PROPERTY

DRAWING NUMBER 699-5008-A SH 1 OF 9

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES DECIMALS .X ± .XX ± .XXX ± ANGLES XX.X ± FRACTIONS ± DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS	DRAWN BY D.Haahr DATE 11/83 CHECKED BY DATE APPROVED BY DATE 11/83 RELEASED BY DATE	apple computer inc. TITLE SPECIFICATION; A2 POWER SUPPLY, PROFILE INTERNATIONAL SIZE A DRAWING NUMBER 699-5008-A SCALE SHEET 1 OF 9
MATERIAL NEXT ASSY FINISH		

?

430

1.0 **TITLE:** **Specifications**, Profile OEM International Power Supply Acceptance.

2.0 **PURPOSE:** **The purpose of this document is to define the criteria, both electrically and mechanically, that Profile power supplies, part number 699-5008, must meet.**

3.0 REQUIREMENTS:

3.1 Materials and construction shall be exactly as initial qualifying pieces. No deviation is allowed without written approval from the M.S.D.Engineering Group and written authorization from Apple Purchasing.

3.2 The power supply is a totally enclosed module provided with a standard IEC connector, On-Off switch and output connector/cable. The power supply is intended to be conduction cooled and maintain specifications up to 75 Degrees Celsius.

3.3 All markings shall be permanent in nature and remain legible when subject to the specified operating, storage, and environmental requirements. All marking shall be insoluable in such standard solvents as water and halogenaced hydrocarbons.

WARNING: Do not use halogenated hydrocarbon solvents for cleaning circuit boards containing aluminum electrolytic capacitors.

3.4 Cosmetics, Aluminum Enclosure:

3.4.1 Definition of Cosmetic Surfaces and Defects:

3.4.1.1. Surfaces:

3.4.1.1.1. Class A Surface: The end of the power supply with silk screening and on-off switch.

3.4.1.1.2. Class B Surface: The bottom surface; the right side (when looking at the silk screened end).

apple computer inc. | SIZE A | DRAWING NUMBER 699-5008-A | SCALE ——— | SHEET 2 OF 9

4/31

3.4.1.2. **Defects:**

3.4.1.2.1. Fine Scratch: A scratch with no apparent visible width when viewed from 18 inches.

3.4.1.2.2. Coarse Scratch: A scratch with obvious width when viewed from 18 inches.

3.4.1.2.3. Scuff: An area with many fine scratches that do not totally obscure the original texture finish.

3.4.1.2.4. Abrasion: An area where the original texture is obliterated. An area where the surface finish has been removed and the base material has been exposed.

3.4.2 Silk Screen Printing, Class A Surface Only:

3.4.2.1. All printing and screening shall conform to artwork, and shall be legible.

3.4.2.2. The printing shall be free of voids and smears when viewed from 18 inches for 5 seconds.

3.4.2.3. The printing shall not be tilted more than 2.5 degrees with respect to the bottom surface.

3.4.3. Class A Surfaces:

3.4.3.1. There shall be no more than 3 coarse scratches which are longer than 0.1" on any surface.

3.4.3.2. There shall be no more that 1 square inch of scuffing on any surface.

3.4.3.3. There shall be no more than 0.25 square inch of abrasion on any surface.

3.4.4. Class B Surfaces:

Any surface finish is acceptable provided there are no dents, butts or deep gouges.

apple computer inc. | SIZE A | DRAWING NUMBER 699-5008-A | SCALE ——— | SHEET 3 OF 9

432

3.4.3. Finish:

All Surfaces to be finished to prevent oxidation.

3.5 Electrical Characteristics:

3.5.1. No failure shall occur when the power supply is subjected to the following maximum conditions:

Vin AC Continuous	280vac rms
Vin AC Surge	360vac rms for 100msec
Short Circuit Loads	Indefinite Period
Open Circuit Loads	Indefinite Period
Operating Temperature	0° to 75°
Storage Temperature	-40° to 85°

3.5.2. Operating Characteristics:

PARAMETER	MIN.	NOM.	MAX.	UNITS
Vin Range	170	230	270	Vrms
Vin Frequency	49	50	51	Hz
Vout Range	+ 4.90	+ 5.00	+ 5.10	Volts
	+11.30	+12.00	+12.60	Volts
	-10.80	-12.00	-13.20	Volts

Load Range

+5v Out	2.00	Amps
+12v Out	1.50*	Amps
-12v Out	0.25	Amps

*3.5 - 4.0 Amps for 10-14 Seconds.

AC Input Power, any Output Short Circuited (<.2 ohm) 5 Watts
(Note: All outputs turn on or off together and power supply must be self recovering when short circuit removed)

apple computer inc.

SIZE	DRAWING NUMBER
A	699-5008-A
SCALE ———	SHEET 4 OF 9

433

	MIN.	NOM.	MAX.	UNITS
+5v Crowbar Firing			7	Volts
+12v Crowbar Firing (Tested with a 1000MV Cap across +12v line)		12.60	13.60	Volts
DC Conversion Efficiency Full Load	75			%
+5/-5v Ripple pk. to pk.			50	mV
+12/-12v Ripple pk. to pk.			100	mV
Line/Load +5v Transient Envelope	4.75		5.25	V
25% Load Step Response Time (Recovery of output to within .1% of final voltage)			500	uSec
Line Drop Out . Capability (full load, nominal)	1 Hold up time 8			Cycle mSec
Isolation: AC Input to SELV Output	3.75			KVAC
Isolation: AC Input to Safety Gnd.	3.75			KVAC
Insulation Resistance: AC to Gnd.	7			Mohms
Insulation Resistance: AC to output	7			Mohms
Insulation Resistance: Output to Gnd.	7			Mohms
Leakage Current Vin 230 VAC to Gnd.			0.75	mAmps

3.6 Electromagnetic Compatibility:

3.6.1. **Emissions:** The power supply, when measured in a fully configured Apple III system shall be 6 dB below limits of VDE 0871/6.78 Section 3.2.1 (conducted) and sections 3.2.2 and 3.2.3 (radiated).

For convenience, limit amplitudes (less 6dB) are reproduced herein. However, the in-effect version (less 6dB) of VDE 0871 is the binding document.

apple computer inc.

SIZE	DRAWING NUMBER
A	699-5008-A
SCALE ———	SHEET 5 OF 9

434

RADIATED	Frequency Range (MHz)	Field (uV/m)	Strengths (db uV)	Distance (meters)
VDE 0871	0.01-30	20	26	30
	30-470	20	26	10
	470-1,000	80	38	10

CONDUCTED	Frequency Range (MHz)	Voltage (uV)	(db uV)	LISN Impedance
VDE 0871/6.78	0.01-0.50	*		150
	0.15-0.50	200	46	
	0.50-30	100	40	

* Straight line from 10kHz (3.5 mV, 71dBuV) to 150kHz (300uV, 50dB uV).

A fully configured Apple III system and compatibility will be determined by the Apple Computer Corporate Engineering EMC Group and the above limits verified. The system will include an Apple III with maximum memory, Disk III disk drive, language card, Profile, Silentype Printer, and Monitor ///.

3.7 Environmental:

3.7.1. Acoustic Noise Output: The power supply shall emit no perceivable and audible noise when located in a library-like acoustic environment; defined as less than 50 db sound power. This acoustic output from the power supply shall contain no predominant tones greater than 5 db above ambient noise referenced against adjacent 1/3 octive band sound levels as measured using the methods of ANSI S1.13 - 1971, Appendix A. This condition shall hold for the power supply when connected to the computer or peripheral device, during all normal modes of computer or device operation and environment.

3.7.2. Temperature Range

Operating: 0° to 75° C
Storage: -40° to 85° C

3.7.3. Humidity: (non-condensing)

25° C to 40° C (ambient) with a relative humidity of 5 to 95%.

apple computer inc. | SIZE A | DRAWING NUMBER 699-5008-A | SCALE ——— | SHEET 6 OF ?

4/35

3.7.4. Vibration and Shock:

The power supply must be able to survive a vibration within a frequency range of 3 of 200 and 200 to 3 Hz at a constant acceleration of 0.5 G's, for on (1) hour in its' shipping orientation axis. (Reference the Apple shock and vibration document #062-0086.)

3.8 Reliability and Quality Assurance:

3.8.1. The complete power supply shall have an MTBF of no less than 30,000 hours when operated over the entire range of operating parameters specified in 3.5 and environments specified in 3.7. This reliability level shall be demonstrated at a minimum 80% confidence level by a stress model reliability prediction and a reliability qualification test based on MIL-HDBK-217C.

3.8.2. Burn-in (accelerated aging) of production level supplies shall be performed to assure that the instantaneous failure rate of supplies delivered to Apple does not exceed 110% of the inverse of the MTBF specified in 3.8.1. Following acceptance by Apple of qualification data no changes shall be made to the burn-in process unless authorized in writing by Apple Computer.

3.8.3. Apple Computer shall be provided with reproducible copies of a complete set of documentation including PCB artwork, envelope drawings, schematics, bills of material, component specifications, approved vendor lists for all components, and manufacturing instructions (including assembly,test, detail/fabrication, inspection, workmanship, adjustment/ calibration procedures, and flow-routing sheets). No changes shall be made to the design or component mix of this supply without prior written agreement of Apple Computer 60 days in advance of any changes or implementation. It is understood that this documentation is the property of Apple Computer.

3.8.4. The vendor will provide Apple Computer with copies of all Engineering Change Notices, Manufacturing Change Orders, and the like to assure that current documentation is available at Apple Computer and establish a knowledgeable documentation liaison to ensure this process.

3.8.5. Apple Computer has the right to perform source inspection and manufacturing audits at the manufacturing sites for the supply. Acceptance of product by Apple representatives does not guarantee final acceptance of the material by Apple Computer. Final acceptance shall be at Apple Computer. The vendor shall perform final inspection to assure each power supply meets the requirements of this specification.

apple computer inc.

SIZE	DRAWING NUMBER
A	699-5008-A
SCALE ———	SHEET 7 OF 9

436

3.8.6. The vendor shall maintain verifiable evidence of all inspections and tests performed on finished supplies, results obtained, and dispositions of nonconforming materials. These records shall be traceable to individual supplies or production lots and shall be made available to Apple representatives upon request.

3.8.7. The contractor shall provide service documentation, including theory of operation, at a level adequate for both Field and Factory service.

3.9 Shipping Protection:

Each unit is to be protected from damage by a corrugated box. The packing shall be adequate to provide protection against damage, breakage or loss during overseas shipment and shall be of a type not destroyed by opening.

Each master shipping carton shall be legibly marked with the following: part numbers, date of shipment, quantity, and country of origin.

3.10 Safety Agency Approval:

The design, component selection, and construction shall conform to and meet the following (Class 1, SELV) Product Safety Requirements:

IEC 380, Safety of Electrically Energized Office Machines.

Note: The main power transformer must be a safety isolating type for SELV circuits.

3.11 Construction:

The enclosure for the power supply is to be fabricated from steel, plated to prevent corrosion.

apple computer inc. | SIZE A | DRAWING NUMBER 699-5008-A | SCALE — —— | SHEET 2 OF 3

437

3.12 Input and Output Connector: Molex Pin Designations

A.C. Input	D.C. Connector
1. A.C. Neutral	1. Reset Monitor Output
2. Key	2. Key
3. A.C. Line	3. -12v
	4. +12v
	5. +12v
	6. Common
	7. Common
	8. Common
	9. Common
	10. Common
	11. +5v
	12. +5v
	13. +5v

Mating Molex Connecotrs

D.C. P/N 09-50-3131
A.C. P/N 09-50-3030

The output will be a single in line connector combining the above two part numbers.

apple computer inc.

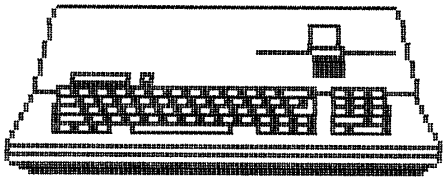
SIZE	DRAWING NUMBER
A	699-5008-A
SCALE ———	SHEET 9 OF 9

FINIS

14

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Head Disc Assembly Specification
(from Seagate Techology)

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

MLC		REVISIONS DESCRIPTION	INC BY	APPROVAL & DATE

PAGES: 12

APPLICABLE DOCUMENTS

Motor Control PCB Assembly	20217-001

Engineering Specifications:

The HDA components shall meet all requirements and specifications set forth in the following documents unless separately specified herein.

Motor Control PCB (Fab drawing)	20218-001
HDA Assembly drawing	58157-001
HDS Electrical interconnect drawing	
Magnetic Disc	30126-001
Read/Write Heads	30134-001
Head Connector	10420-016
Spindle Motor Connector	10417-006
Stepper Motor Connector	10417-005
Index Connector	10417-005

DWG. NO.

MODEL NO. FIRST USE	ST-412.9	NEXT ASSY FIRST USE FINAL ASSY

DRAWN		
CHECK		
APPD(E)		
APPD(M)	[illegible]	1.19.81

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES ON
DECIMALS .XX .XXX
ANGLES ±

SEAGATE TECHNOLOGY

ST412.9 HEAD DISC ASSEMBLY

SCALE	SIZE	DWG NO.	REV. EC
	A	30173-001	C242

DO NOT SCALE DRAWING

SHEET 1 OF 14

440 Engineering Specification

30173-001 0292

ST 412.9 HEAD DISC ASM SPECIFICATION
TITLE HEAD DISC ASSEMBLY

SHEET 2 OF

APPLICABLE DOCUMENTS (continued)

TK Ø Connector	10417-005
Spin Motor PCB Connector	10419-002
10416-XXX (Ref) Mates with	10417-XXX
10415-XXX (Ref) " "	10420-XXX
10418-XXX (Ref) " "	10419-XXX
Shock & Vibration Spec.	30138-001
Acoustical Specification	30131-001
Shock & Vibration Spec.	30138-001

SEAGATE TECHNOLOGY

441 Engineering Specification

30173-001 0292

TITLE ST412.9 HEAD DISC ASM SPECIFICATION 30173-001

SHEET 3 OF

CONTENTS

I. SCOPE

The Head Disc Assembly (HDA) is a sub-assembly of the ST412.9 Final Disc Drive Assembly. It will undergo all the drive testing before being shipped to the customer. The Head Disc Assembly will be a complete ST412.9 disc drive less the following hardware:

1) Main control PCB

2) Front panel

II. CAPACITY

Unformatted

Mbytes/Drive	12.76
Bytes/Track	10416

Formatted 32 Sectors/Track (Soft)

Mbytes/Drive	10.0
Bytes/Track	8192

SHUGART TECHNOLOGY

442 Engineering Specification

30173-001 029.

SHEET 4 OF

ST 412.9 HEAD DISC ASM SPECIFICATION

TITLE 30173-001

	Max recording frequency (MHZ)	2.5
	Transfer Rate	
	Mbits/Second	5.0
	Density	
	Flux Changes/Inch	9074
III.	MEDIA	
	Tracks	
	Per Inch	345
	Per Surface	306
	Per Drive	1224
	Defects (Hard Errors)	
	Per Drive	12
	Per Cylinder Zero	0

IV. ENVIRONMENTAL

	Operating	Shipping	Storage
Ambient Temperature*	39° - 135°F 10° - 50	25° - 144°F -4° - 62°C	-8° - 176° -22°C - 80°
Max Temperature Gradient/Hour	18°F	18°F	18°
Relative Humidity	20 - 80%	TBD	TBD
Maximum Wet Bulb	78°	No Condensation	
Stray Magnetic Field	(1" from casting)	20 Gauss Max	
Altitude		10,000 ft. Max	

* Ambient Temperature is defined as the HDA casting temperature.

443 Engineering Specification | 30173-001 | 0292

TITLE ST 412.9HEAD DISC ..SM SPECIFICATION 30173-001 | SHEET 5 OF

V. RELIABILITY

Error Rate (Excluding defects) - When used with Seagate Technology PCB P/N TBD.

Soft read errors (16 retries min.)	1 per 10^{10} bits transferred
Hard read errors	1 per 10^{12} bits transferred
Seek errors	1 per 10^{6} seeks
MTBF (Meantime before failure)	
Typical usage	11,000 hours
PM (Preventive Maintenance)	None
MTTR (Meantime to Repair)	30 minutes
Component Life	5 years
Media Life	10,000 starts/stops

VI. POWER

DC Voltages (To motor speed control)

+12 VDC ± 10%, .9 AMP running current; 2.7 AMPs typical motor start current. (see ST412 manual for start profile).

BTU/Hr. (1 watt = 3.413 BTU/Hr.)

Watts	25	± 40%
BTU/Hr.	85.3	± 40%

VII. INTERFACE

TBD

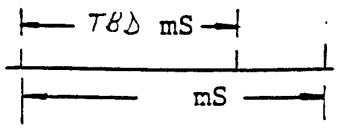
VIII. DRIVE

A. Mechanical

Seek Settle Time*

Uncontrolled TBD

Controlled TBD using this step sequence



Track to Track (Seek = 3 ms)

* Using Seagate recommended PCB

SEAGATE TECHNOLOGY

Engineering Specification

ST412.9 HEAD DISC ASM SPECIFICATION
30173-001

30173-001 0292

SHEET 6 OF

Drive Motor Start Time — 15 sec.

Spindle Speed** — 3600 rpm ± 1%

HDA Cleanliness

The air inside the HDA shall be class 100 (0.5) micron particles)

Mounting

See Figures 1 & 2

Vibration and Shock

Shall meet the requirements of Seagate Specification 30138-001

B. Electricial

For purposes of normalized test data, and unless otherwise stated, all test measurements shall be taken at the ambient conditions of 68 + 5°F (20 + 3°C) and 40 to 60% relative humidity after the HDA has been stabilized at the test environment for at least 1 hour.

Not withstanding this, the HDA shall be capable of meeting all the requirements of this specification over the full operating environment of Paragraph IV, Page 4.

Warm Up Time

The HDA shall meet all operating specifications within 15 seconds after spindle motor has reached 3600 RPM at all operational environmental conditions.

1. Spindle motor speed regulation — 3600 ± 1% RPM

2. Spindle motor current

Start	2.75amps max.
Running	1.0 amps max.

3. Stepper motor/phase — TBD

**Adjustments are customer responsibility
***For shipment, the heads shall be moved center track while the spindle is rotating.

4. Performance (Head Disc Asm)

Average 2F amplitude	1.0 mv p-p min.
Inside window margin (without write precomp)	30 ns total
Minimum resolution on any track (without precomp.)	60%

Wide Gap Measurement

1. DC erase tracks 303, 304, 305
2. Write 2F on track 304
3. Write 1F on tracks 303 & 305
4. Measure 1F amplitude relative to 2F amplitude on track 151 using H.P. spectrum analyzer Model No. 3585A or equivalent.
5. Failure is defined as a measurement $\geq$ - 26dB.

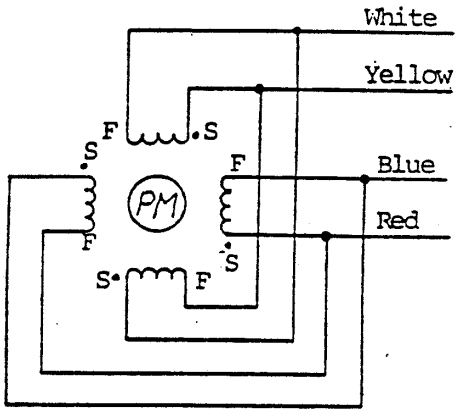
SEAGATE TECHNOLOGY

440 Engineering Specification

30173-001 029

TITLE ST412.9 HEAD DISC ASM SPECIFICATION 30173-001

SHEET 8 OF



S: Start F: Finish

ELECTRICAL REQUIREMENTS

STEPPER MOTOR

Switching Sequence for CCW Rotation
Facing Mounting End

Step	Pin 4 Blue	Pin 5 Red	Pin 2 White	Pin 1 Yellow	Pin 3 Not Us
1	-	+	+	-	
2	+	-	+	-	
3	+	-	-	+	
4	-	+	-	+	
5	-	+	+	-	

SCHEMATIC

SPECIFICATIONS
Step per revolution: 400 (0.9° per step)
Step to step accuracy (Notes 1,2,): ± 6%
Positional Accuracy (1,3): ± 6%
Rotor Inertia: 20 Gcm^2 (.28 MOISS)
DC Phase Resistance: 38 ± 3.8 at 25° C
Phase Inductance: 27 mH ± 20% at 1 KHZ
Phase Voltage: 9.2
Phase Current (Steady State): 240ma
Holding Torque: 720 gcm (10 oz. - IN) Nom.
Pull-out Torque: 360 gcm (5 oz. - IN) NOM.

NOTES:

1. Measurements made at rated current on each phase.
2. Between any two adjacent step positions.
3. Maximum error in 360°.
4. Motor to be driven bipolar.
   The above specifications is of a motor in bipolar mode.
5. Leads: 4, No. 26 AWG PVC insulation UL & CSA approved.

TBD.

IX. PHYSICAL REQUIREMENTS

The HDA can be mounted on any side except upside down or front and back. (See Figures 1 & 2)

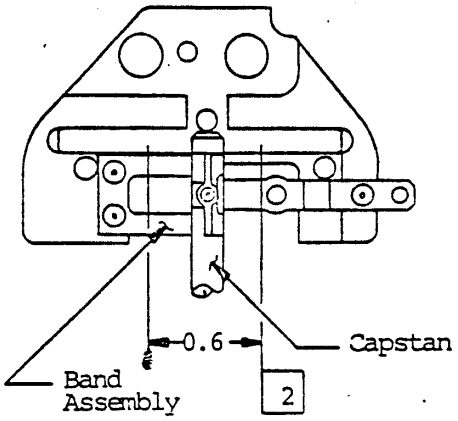
The HDA is shock mounted for vibration isolation. In the final mounting configuration, care shall be taken to insure that the operation of the three shock mounts is not restricted.

SHEET 10 OF

ST412.9 Head Disc ASM Specification
TITLE 30173-001

30173-001 0592

BAND ASSEMBLY



BAND SPECIFICATIONS

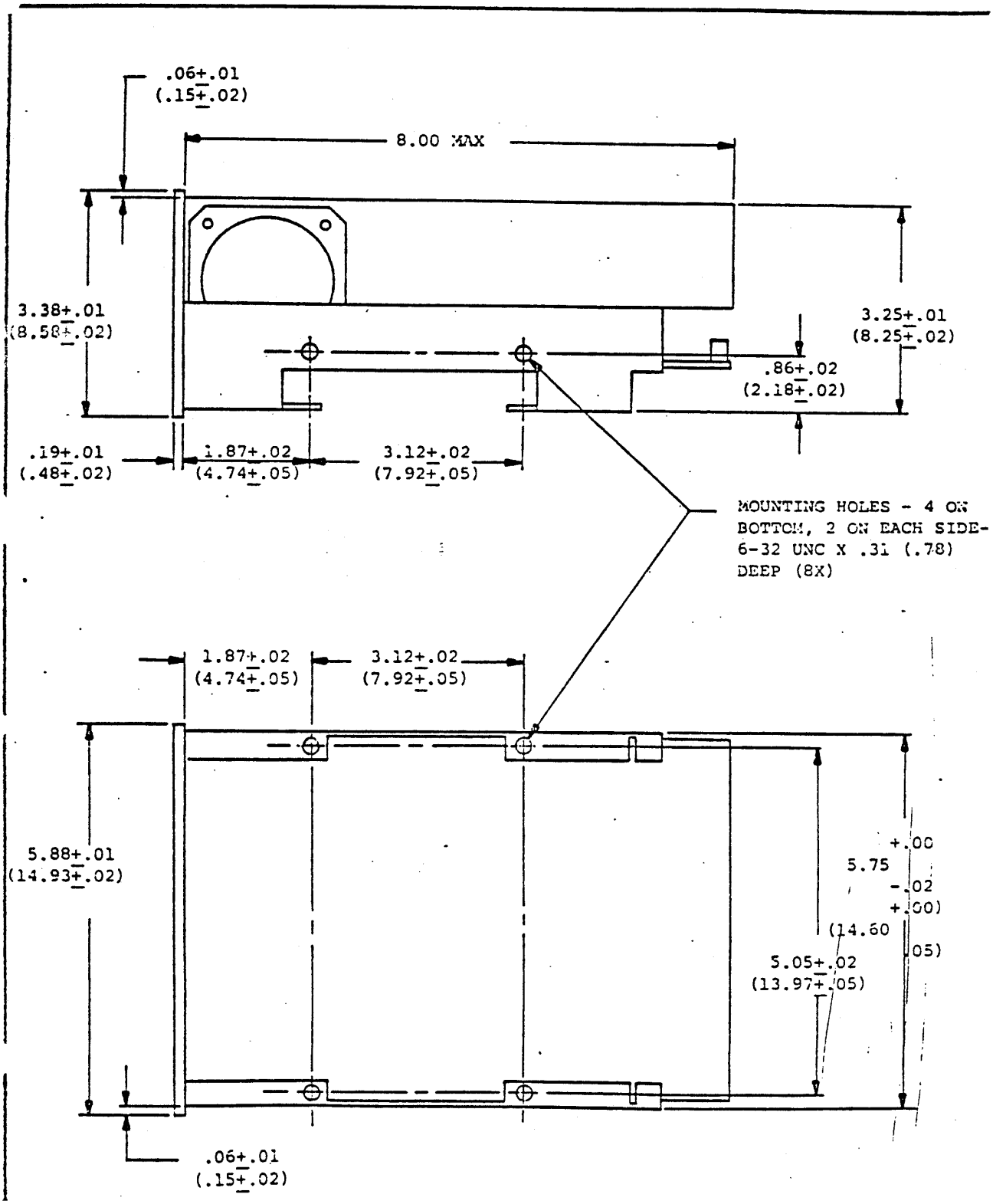
1. Band shall be capable of withstanding 2.5 lbs. Max. tension.
2. There shall be no creases on the active portion of the band.
3. Band shall be free of contaminants that would decrease it's life, such as finger prints and edge nicks.

449

30173-001

TITLE: ST-400 SERIES INTERFACE SPECIFICATION

SHEET 11 OF

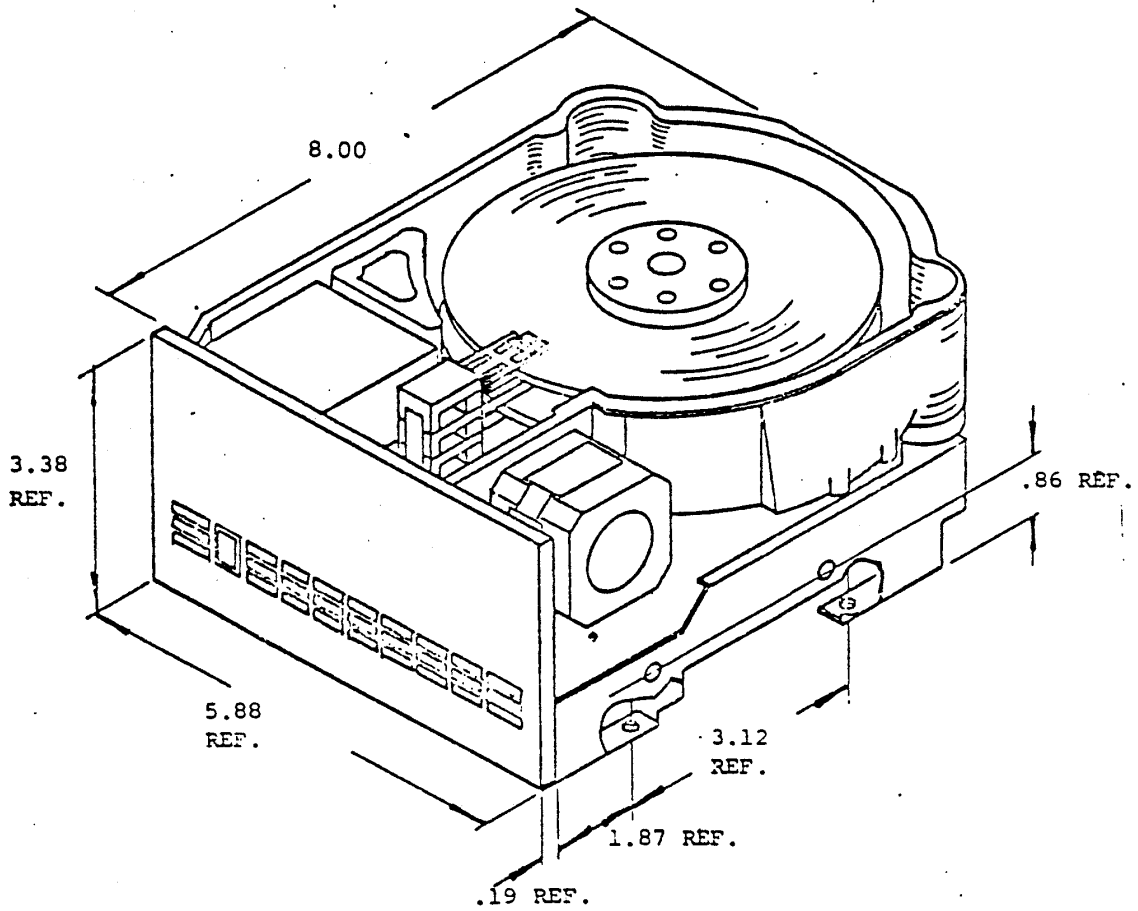


450

30173-001

TITLE: ST-400 SERIES INTERFACE SPECIFICATION

SHEET 12 OF



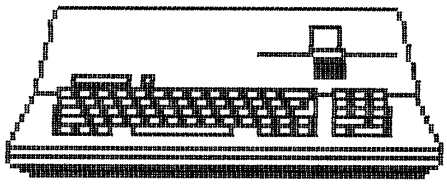
FINIS

451

15

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Interface Card Set-Up and Test Procedure, Apple III/ProFile

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

452

REV.	ZONE	ECO #	REVISION	APPD	DATE
A		M-702	INITIAL RELEASE		

Pages: 9

- 6 sheets
- 3 drawings

DRAWING NUMBER 063-0128-A SH 1 OF 6

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. DECIMALS .X ± .XX ± .XXX ± ANGLES XX.X ± FRACTIONS ± DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS	DRAWN BY E. Gibbin DATE 7/82 CHECKED BY W. Garcia DATE 7/82 APPROVED BY Kay Yokam DATE 8/82 RELEASED BY Tim Gardner DATE 8/82	apple computer inc.
MATERIAL: N/A		TITLE: INTERFACE CARD SET-UP AND TEST PROCEDURE, PROFILE/APPLE III
NEXT ASSY. FINISH: N/A		SIZE A · DRAWING NUMBER 063-0128-A · SCALE: N/A · SHEET 1 OF 6

apple computer inc.

DOCUMENT NO. 063-0128-A

PAGE 2 OF 6

1.Ø TITLE: PROFILE/APPLE III INTERFACE CARD SET-UP AND TEST PROCEDURE

2.Ø PURPOSE: This is a guide to setting up and automatically testing the Profile Apple III Interface Card. This test is used in production to test and debug the boards.

3.Ø EQUIPMENT:

3.1 Apple II Computer
3.2 Disk Drive with Controller
3.3 Monitor
3.4 Silentype Printer and Interface Card
3.5 VIA Test Interface Card
3.6 Profile Interface Card Test Fixture with Cables
3.7 Oscilloscope (nothing fancy needed)
3.8 Profile Apple III Interface Automatic Test Disk
3.9 Profile Apple III Interface Schematic

4.Ø REFERENCE DOCUMENTS:

8ØØ-8ØØ2	Assy, Interface Tester, Profile/Apple III
8ØØ-6ØØ2	Assy, PCB, Interface Test Card, Profile/Apple III
8ØØ-6ØØ3	Assy, Enclosure/Interface Test Card, Profile/Apple III
8ØØ-Ø2Ø1	Enclosure, Interface Tester, Profile/Apple III
8ØØ-Ø2Ø2	Strain Relief 2.5 x .5 x .12, Interface Tester Profile/Apple III
8ØØ-Ø2Ø3	Strain Relief 3.44 x .5 x .12, Interface Tester Profile/Apple III
Ø51-Ø194	Schematic, VIA Card, Interface Tester, Profile/Apple III
Ø51-Ø195	Schematic, Interface Test Card, Profile/Apple III
821-Ø183	PCB, Interface Test Card, Profile/Apple III
889-ØØ19	Diskette, Auto Interface Test, Profile/Apple III

5.Ø TEST PROCEDURE SET-UP:

5.1 Connect the two cable connectors to J3 and J4 on the VIA (Apple III Tester Interface) Card. The Cables are labeled accordingly.

5.2 Plug the VIA Card into the Apple slot #2.

5.3 Plug the Silentype into slot #1.

5.4 Attach the disk Drive in the conventional manner into slot #6.

6.Ø TEST PROCEDURE:

6.1 Insert the Test Disk and turn the Apple II Power on.

6.2 The Disk will boot with the message: "LOADING PROFILE/APPLE III INTERFACE CARD TEST PROGRAM" and then ask you to plug in the Interface Card and attach the DB-25 connector. It is important

that the Board Under Test is inserted into the zero force insertion test socket with the components facing you with the DB-25 connector on your right. Also, note and use the lever on the connector for opening and closing the pins. After the Card is plugged in corectly press either a "P" or an "S" to start the test and print any errors to the P(rinter or the S(creen.

6.3 The program will take about 30 seconds to test the Card and will respond with the message: "NO PROBLEM FOUND" or will print the failures on the selected output device.

6.4 If the Board is good type an "A" and the original prompt will be shown. If the Card is bad, the selected output device will print the message: "PROFILE INTERFACE CARD TEST IC LIST/LINES IN ERROR:" followed by some short IC addresses ("A2 A5 B4 C8" for instance) followed by a list of the incorrect lines. In many cases the IC addresses will not be displayed. This will happen if the failure pattern is not in the Apple fault tables.

6.5 After the fault is printed the following message is displayed on the screen: "DO YOU WISH TO LOOP THROUGH THE TEST? (Y/N) ". If you choose N(o, the program will display the beginning test prompt and you can continue to test as before. If you choose Y(es, the program will continue to send test data to the Card and you can use the scope for troubleshooting. Press the ESC key to end the looping. (It could take up to 30 seconds for the present cycle to complete itself and the Apple will appear as if it is just sitting there.)

6.6 During the GO-NO GO testing tear the print-out off the printer and tape it to the board so the debug person can save this step in the debug process.
NOTE: Typically there is more than one person testing and debugging the boards. The Go-No Go testing can be accomplished in the printer mode and this person can tape the printed error messages to the boards and another person can do the debugging in the screen mode and use the looping function with the scope.

ADDENDUM

APPLE III INTERFACE FOR PROFILE INTERFACE TESTER

VIA CARD MODIFICATIONS DESCRIPTION

The VIA card in its original condition will not work properly if it is fully stuffed with components and connected to the Interface Card Test Interface Board. There are a few simple modifications that have to be made so that the board will work properly. In the first place, only the top three VIA chips are needed in the board to communicate with the board under test. Do not insert chips U5 and U8 into the board. Likewise, do not insert connectors J1 and J2 into the board as they are not used either. Also, U1 is not needed and there needs to be a jumper soldered between pads 4 and 7 of this chip on the board. If you look up at the top left hand corner of the schematic you will see that this will enable pin 1 of U2 permanently. There is one last modification to the board and the purpose of this is to bring +5V out to pins 26, 28, 3Ø, and 32 of connector J4. To do this, the four traces that connect VIA chip U11 to connector J4 must be cut on the front side of the board. To do this, locate the group of traces connecting the upper right-most VIA chip to the connnector above it. The traces that need to be cut with the Exacto knife are the fifth, sixth, seventh, and eighth traces over from the left. Just to be sure, there should be four traces still connected to the left of these. The final step in this modification is to connect a large jumper wire between pins 26 through 32 of the back side of connector J4 and the +5V bus. This is used to bring a solid +5 volts out to the card under test.

TESTER PC BOARD MODIFICATIONS

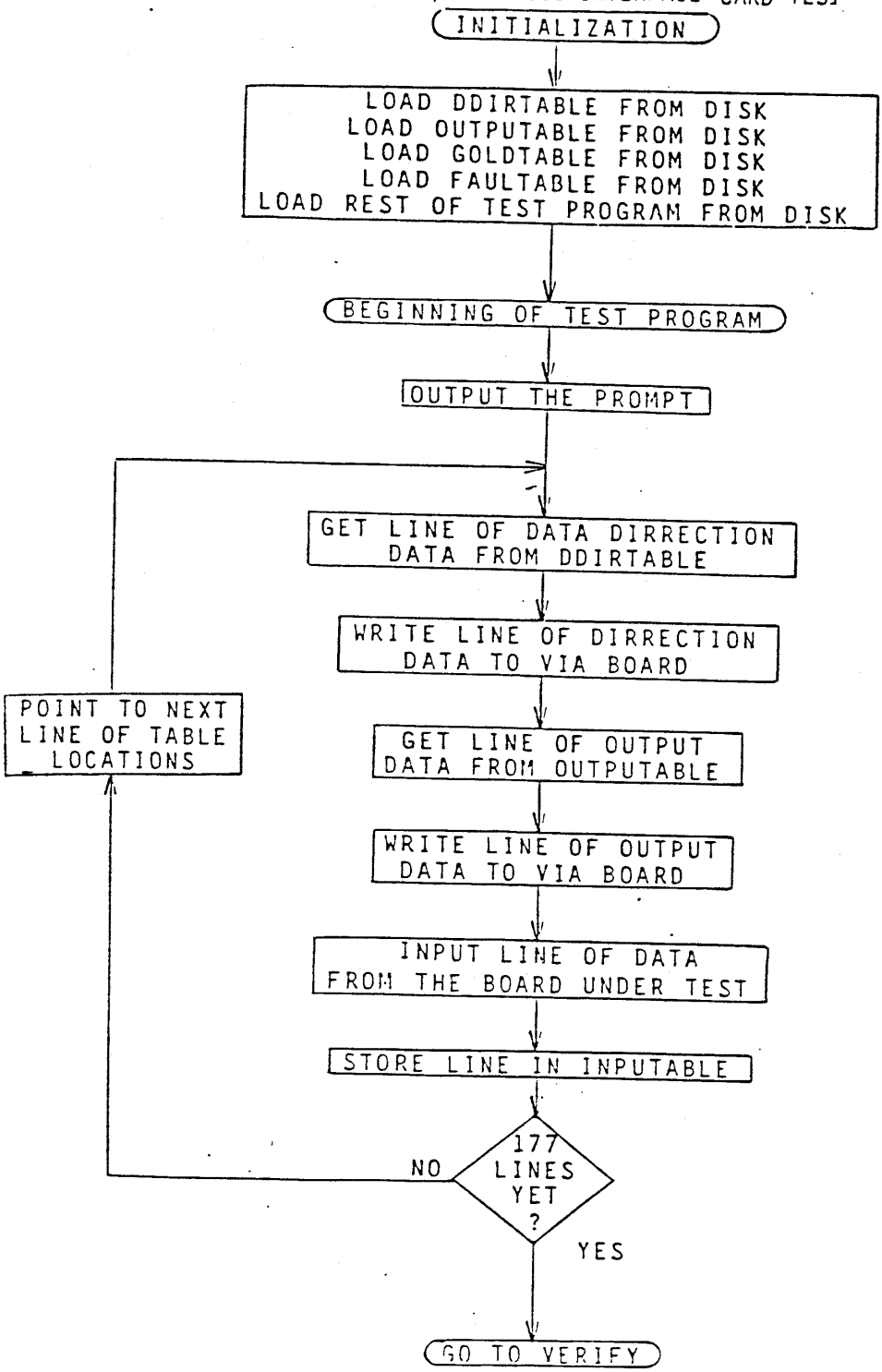
Originally there was a large resistor in series with the board under test. There was also a LED and resistor across it which would light when there was a short. This has been eliminated. Add a jumper wire between the Pads of R1.

4/50

DOCUMENT NO. 063-0128

PROFILE/APPLE III INTERFACE CARD TEST

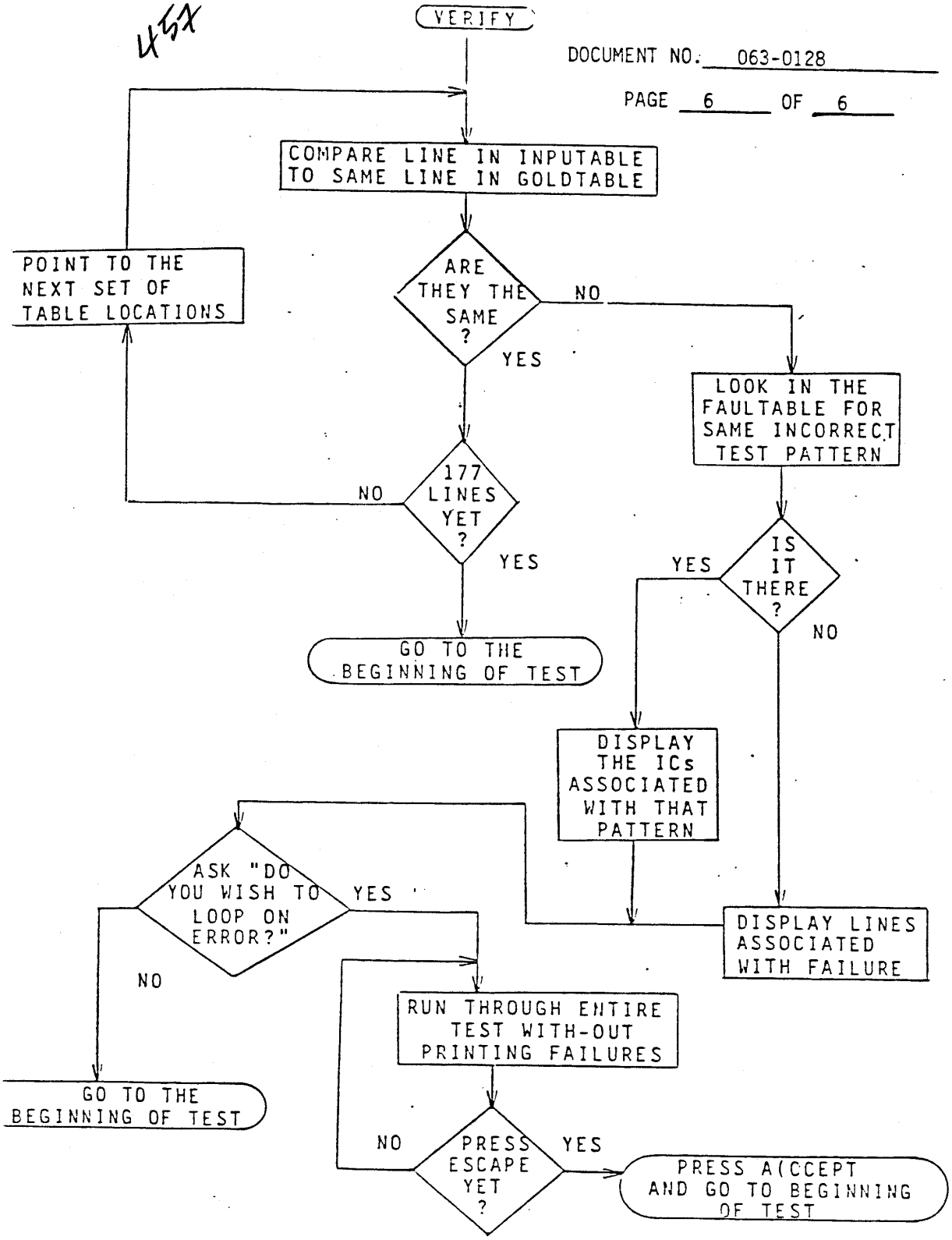
PAGE 5 OF 6



4552

DOCUMENT NO. 063-0128

PAGE 6 OF 6

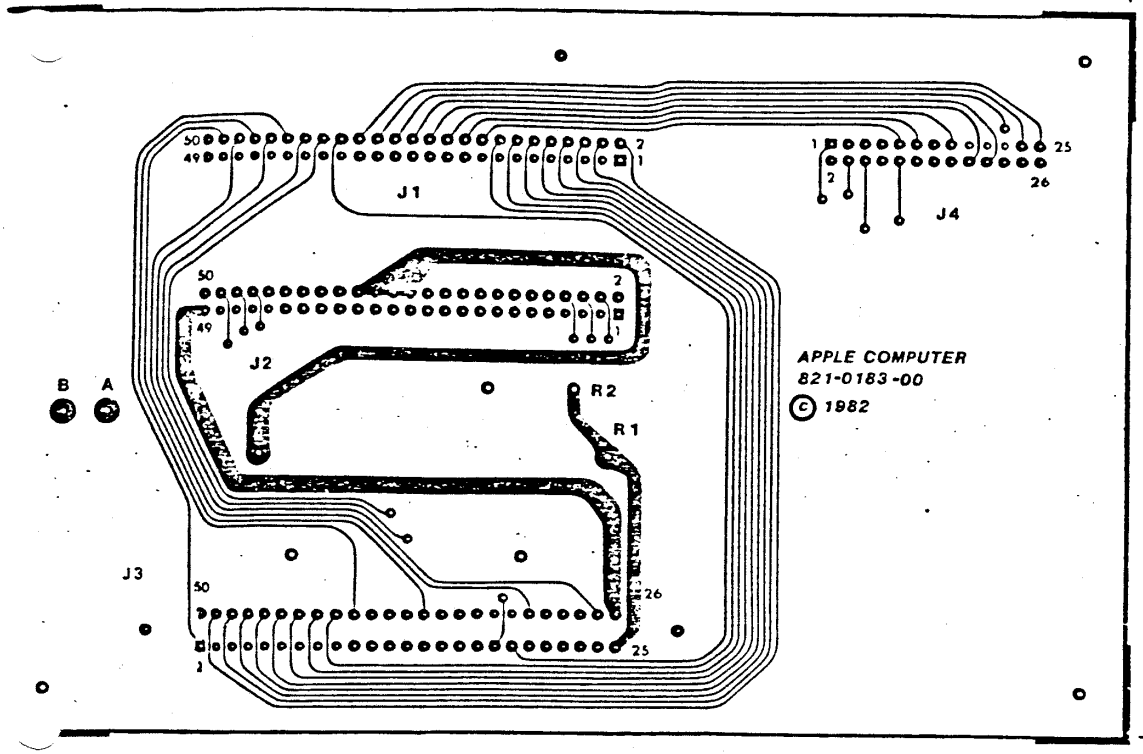


458

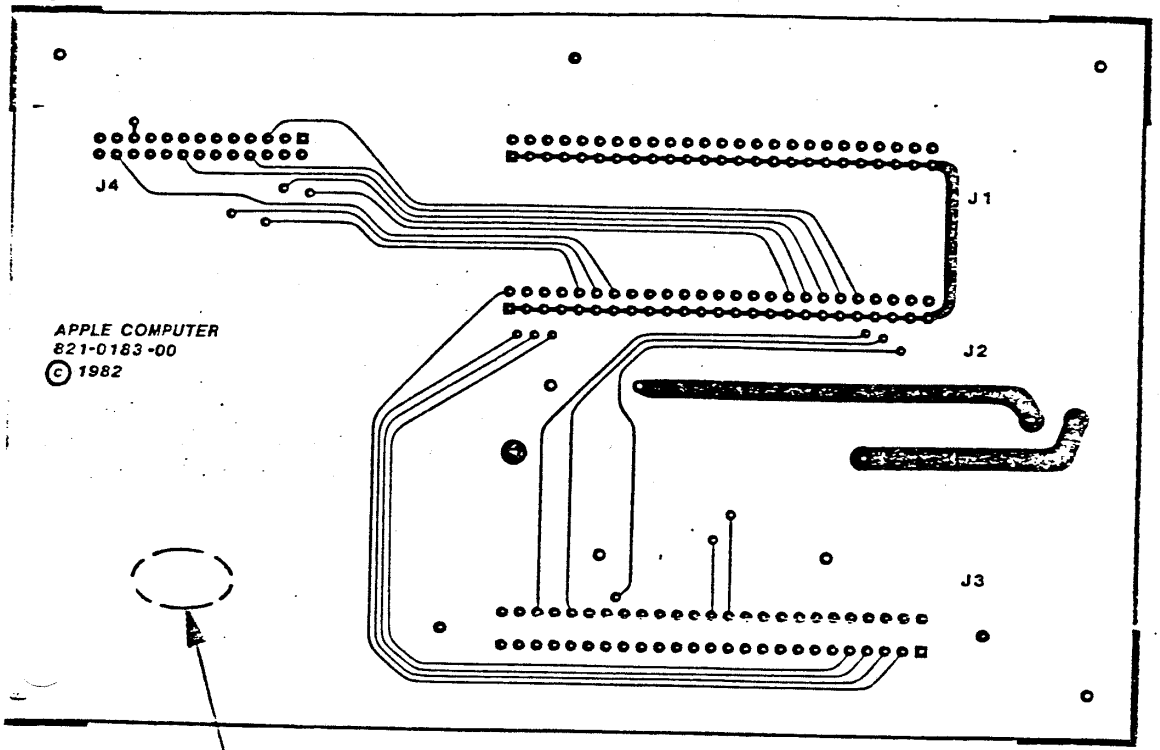
REF

COMPONENT SIDE

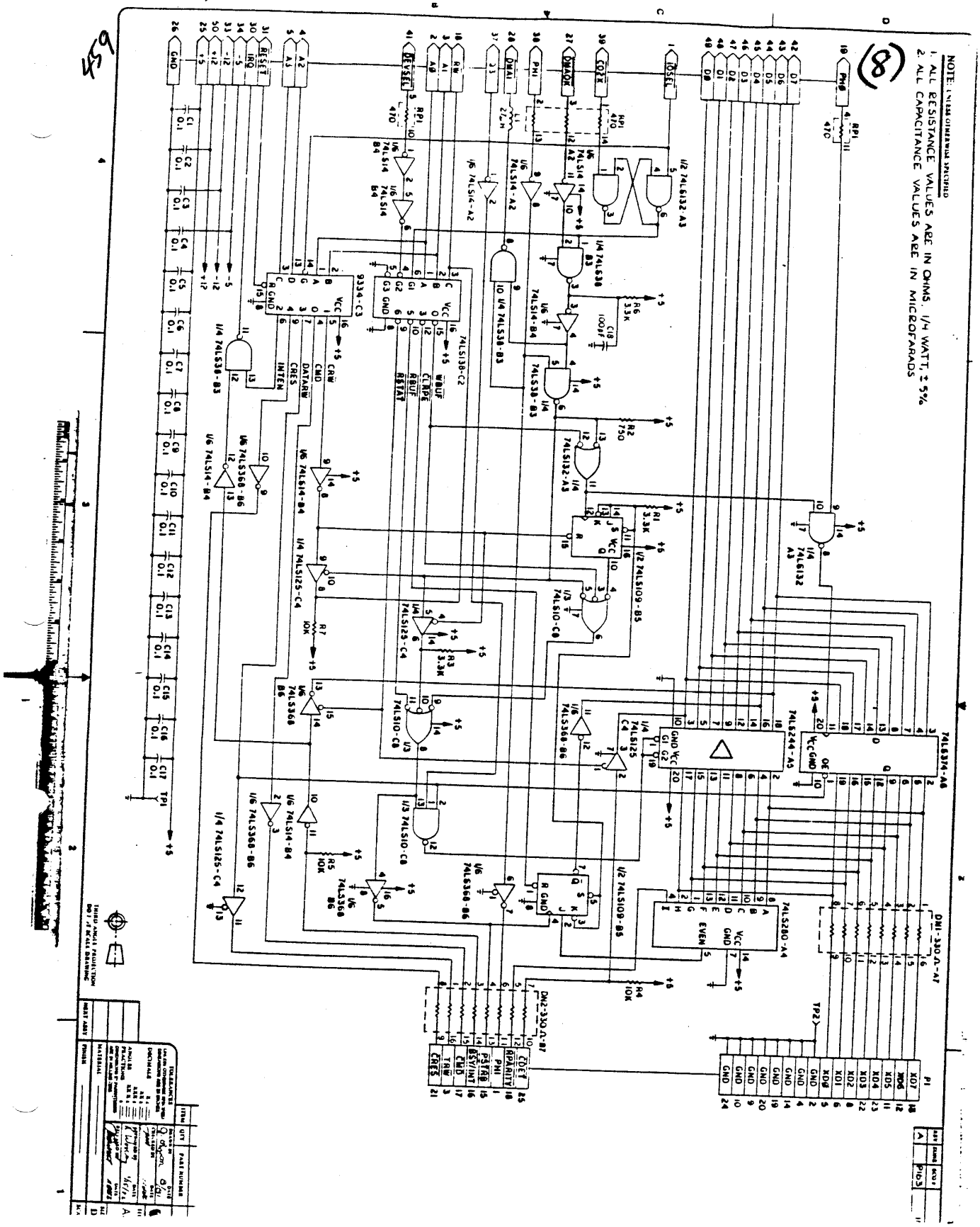
(7)



107.95
REF

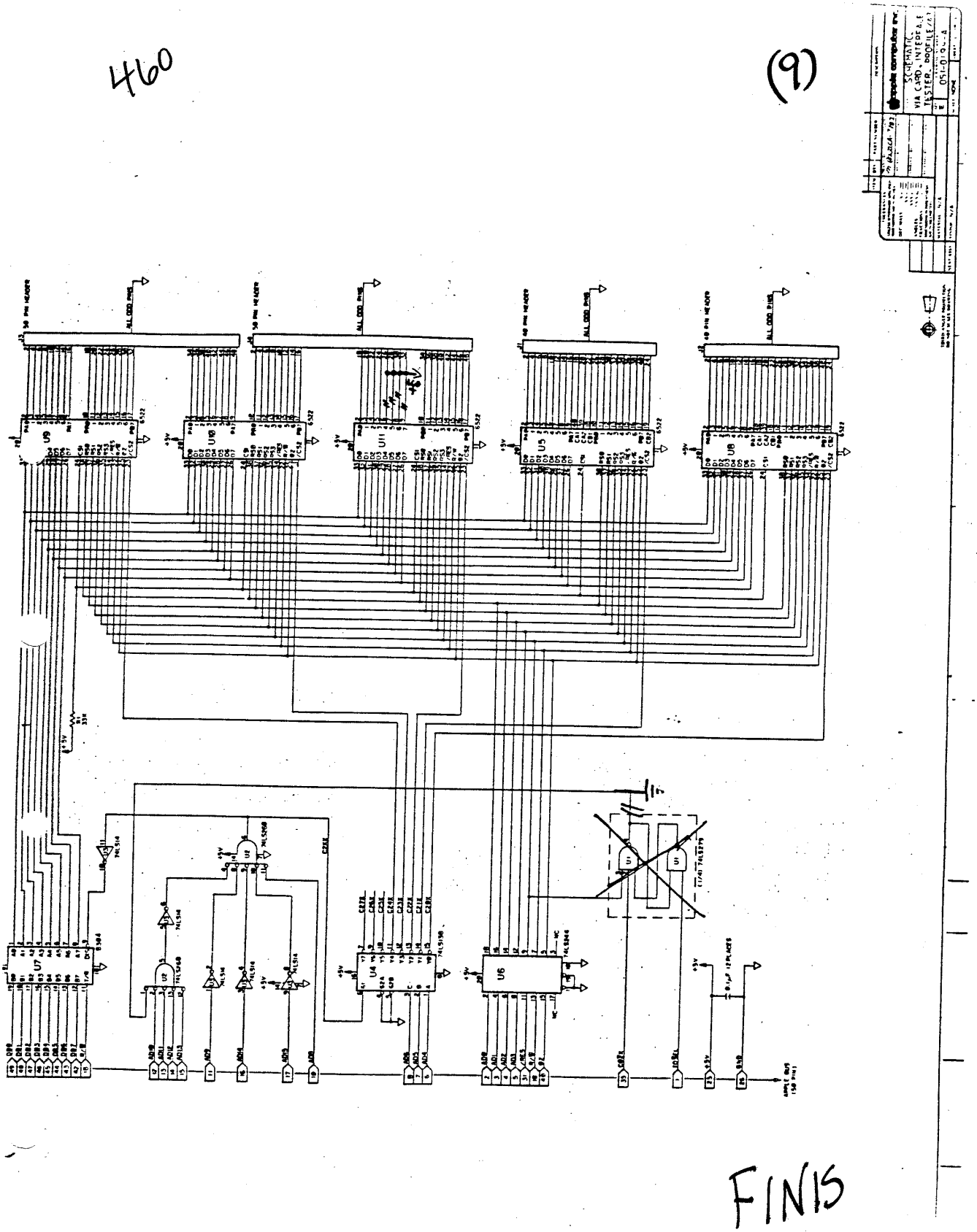


NON-COMPONENT SIDE



460

(9)



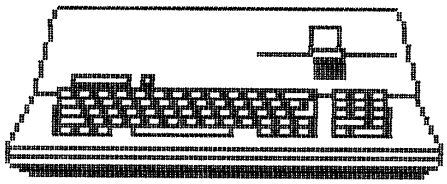
FINIS

461

16

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Analog Board Functional Test
(05 Jan 1984)

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

HDV

January 5, 1983 ~1984?

PROFILE ANALOG BOARD FUNCTIONAL TEST

Pages: 6

ANALOG BOARD READ TESTS WITH DEBUG TESTER

EQUIPMENT:

1. SCOPE, TEKTRONIX 465 or equivalent 100 Mhz scope
2. SCOPE PROBES, x10, 3 probes, two w/ground
3. PROFILE, known good, without Analog board
4. PROFILE A/// I/O board and cable
5. A/// computer, M/// monitor
6. RIBBON CABLE, 12 inches long, Controller to Analog board
7. EXTENTION ADAPTOR CABLE, P6, P7, P8 from HDA
8. PIGGY-BACK Z8, with "Debugger Rev. 11" Eprom
9. DISKETTE, Debug Test (889-6003), "Disk Debugger Rev. 17"
10. TRAY, Profile production, foam lined (918-0123A)

STATION SETUP:

1. Remove top cover of Profile.

2. Remove Power Supply/Controller board, and HDA/Analog board assemblies from the Profile plastic base and put them in the Profile tray.

3. Remove ribbon cable from controller to analog board, set aside.

4. Turn HDA upside down so flat cover plate with WARNING label is down, and round end of HDA is toward controller board.

5. Connect long ribbon cable to controller board connector P2 from HDA.

6. Insert A/// I/O board in slot 4 (floppy drive side). Connect A/// I/O board to controller board P1 connector with I/O cable.

7. Remove plastic Z8 (341-0171) from Controller U25 socket and save.

8. Install ceramic piggy-back Z8 with "Debugger Rev 11" mounted on top into 40 pin socket at Controller board U25. Check pin 1 direction is correct (toward P1 connector).

9. Attach one scope probe with no ground clip to trigger input of scope, and other end to TP1 (near P2) on controller board. Set time base to 1 mS/div, External triggering.

10. Attach probe with ground clip to scope channel 1. Set to 1 volt/div, AC coupled.

11. Attach probe with ground clip to scope channel 2. Set to .5 volts/div, DC coupled. Set scope trace (grounded) to a reference line. Variations from this line will be used to adjust VCO pot, and show VCO problems later.

12. Plug in AC power cord to Profile power supply under controller board.

13. Insert "Debug Test" disk into A/// built-in drive and boot up the computer. Computer may be left on while testing and during Analog board under test changes.

1-6

PROCEDURE: 463

1. Place Analog board to test on HDA. Plug in head connector J5 as the board is slipped in.

2. Plug P6, P7, P8 from the HDA into Analog board connectors J6, J7, and J8. Check that they are plugged into the correct connectors.

3. Install hold down screw at ground lug (near TP5). Ground strap from HDA MUST be attached here also. No other mounting is required.

4. Plug in ribbon cable from controller board to J2 on Analog board.

5. Turn ON Profile power switch at back of supply.

6. Attach channel 1 scope probe at Analog board TP1 or TP2 Read Amp, next to U7 (CA3127). Attach ground clip from scope probe to nearest ground test point.

7. Scope probe from channel 2 clip onto TP8 VCO, near U2 (7474).

8. Wait for the HDA stepper arm to move from "park" to track 0. The analog data signal on scope channel 1 should look like the photos of "good signal". If not, check that the spindle motor is turning, and that scope probe 1 is properly connected.

9. Break VCO pot on the Analog board (R56) loose, so it will turn freely. Set the VCO pot by setting scope channel 2 to ground, and adjusting to a ground reference line. Set back to DC coupling, and turn the pot (R56) from lock to lock. The VCO signal on the scope should move up and down at least .50 Volts.

10. Set VCO pot to ground reference line (VCO = 0 Volts +/-.25 Volts) and glue.

11. Type "R0" RETURN to start at track 0. The stepper arms should move around about 1/2 turn.

12. Enter the command to start scanning the sectors, "R PE +H X" RETURN. The monitor will show that the track, head, and sector numbers are increasing. The stepper arm will move around counter-clockwise one track at a time.

13. Watch the read data on scope channel 1 and the VCO signal on scope channel 2, and FAIL the analog board if it does not meet the following:

    Read signal = 2 Volts minimum (peak to peak).

    AGC signal = 1 Volt base to peak on at least one side (top or bottom).
    AGC shall show "bell" effect immediatly after sector gap periods.

    VCO signal = No excessive noise or oscillation, 200mV MAXIMUM peak/peak.

14. If a data read error occurs, the monitor display will stop and say "ERROR - Press space key to continue". Record the track/head/sector, and the error number from the last data line of the monitor on the board reject tag.

2-6

464

15. Press the A/// space bar to continue if an error occured. Stop the test if more than 7 errors occur or when the track number goes around to zero by pressing the ESCAPE key.

16. If any errors occured on the monitor, re-try that sector to verify by typing "RTxxHxSx" RETURN where "x" are the numbers of track/head/sector that error occured on. Then type "R PE X" RETURN. If the error appears continuously, the Analog board is defective. REJECT the analog board if any of the errors appear consistantly.

17. IMPORTANT:

    Enter the command "RT99" RETURN before ever shutting power OFF to park the heads off the formatted data areas. Failure to do this will eventually destroy the disk surface causing permanent errors.

18. Turn OFF the power switch in back of the Profile power supply.

19. Remove the scope probes attached to the Analog board TP1 or TP2, and TP8. Carefully unplug the connectors at J6, J7, and J8.

20. Remove the ground wire screw and start to pull up the board from the J2 connector edge. Push off the HDA head connector (J2) when the Analog board is part way out.

3-6

II. ANALOG BOARD WRITE TEST WITH ANALOG TEST FIXTURE

EQUIPMENT NECESSARY: 4465

1. PROFILE ANALOG TESTER with cables (890-0184)
2. POWER SUPPLY, 3 output, Lambda LPT-7207-FM (or equivalent)
3. DVM, Fluke 8010A (or equivalent)
4. SCREWDRIVER, medium Phillips
5. SCREWDRIVER, small flat
6. JUMPER, with clip on each end

TESTER SET-UP:

1. Install and secure the Analog board Under Test to the test fixture using standoff to hold the board in place.
   (The screw MUST be mounted in the GROUND LUG hole)

2. Connect the board to the tester as follows:

   1. Tester J2 to board J2 (Controller to Analog connection)
   2. Tester J5 to board J5 (Head Cable connection)
   3. Tester J6 to board J6 (Index Cable connection)
   4. Tester J7 to board J7 (Stepper Motor connection)
   5. Tester J8 to Board J8 (Track 00 connection)
   6. Tester J9 to board TP1 and 2 (2 wires)
   7. Tester J9 to board TP8 (1 wire)
   8. Jumper from TP8 on Reference Analog Board to R80 next to J2 on Board Under Test (use jumper for this).

NOTE:

The Analog Board Tester has a modified Profile Analog Board mounted on it for signal reference. This board can be identified as missing R-106, Pin 5 of IC U2 pushed out of its socket, and a jumper to TP8 from U2 Pin 5 on the back of the board. The Reference Board will be driven by the clock signal on the Board Under Test.

DO NOT REMOVE any ICs or the Reference Analog Board for any reason!!!

NORMAL TEST SWITCH POSITIONS:

Set the ^F switch to the "MIDDLE" position

Set all the other switches toward you (OFF)

Set the Bit Shift dial pot at 0.0

Set the X, Y pots both at 5.0 .

Several LEDs will be lit on the board when power is applied to it. This is normal, do not pay attention to them unless told to do so in a test.

4-6

POWER SUPPLY SET UP:

4bb Move the DVM probes to the Reference Analog board and re-adjust the power supply so that the voltages on the board are correct.

+12V +/-0.1 volt (across Analog board Under Test C38)
-12V +/-0.1 volt (across Analog Board Under Test C39)
+ 5V +/-0.1 volt (across Analog Board Under Test C40)

If the voltages are low or shorted, do the following for the board under test:

A. Locate C38, C39, and C40 on the board. Use an ohmmeter to determine if they are shorted.

B. Also check for any cut or shorted traces on the board. If the +5V line is the only one shorted, locate and check RG1 (three terminal regulator 1) on the Analog board Under Test.

CAUTION:

DO NOT apply power to a known shorted Analog board. You may DAMAGE the board and/or the power supply beyond repair.

TEST PROCEDURE:

1. WRITE CURRENT TESTS: (Lower and Higher)

A. Write Current Test (Lower)

(1) Set PCP, WCA, POK, and WC switches ON. Set WRT/RD to WRT.

(2) Check DVM on board. Reading should be 310mV to 370mV.

(Divide reading by 10 for peak to peak write current)

If test fails, check IC U4, U1, R14, and R10.

B. Write Current Test (Higher)

(1) Leave WCA, POK, WRT, and WC switches set ON.

(2) Set PCP switch OFF.

(3) Check DVM on board. Reading should be 410mV to 500mV.

(4) Using Head Select switches HS0 and HS1, verify that current checks good in all 4 heads. This verifies the head diode matrix.

(5) Set POK switch OFF. DVM reading will decrease at least 100mV, indicating write enable circuit operation.

If test fails, check IC U1, U4, U31, R10, R14, R16, and R17.
If any one head selection failed, check diodes CR5-CR12, U31.

C. Return all controls to normal position.

5-6

2. WRITE PRECOMP TEST:

A. Select HD0. (Verify - WCA switch OFF, ^F switch MIDDLE)

B. Set PCP, POK,WRT switches ON.

C. Set X Pot at 5.0

D. Set Y and Bit Shift Pots at 0.0

E. Turn Y Pot CLOCKWISE until PCMPA LED goes ON.

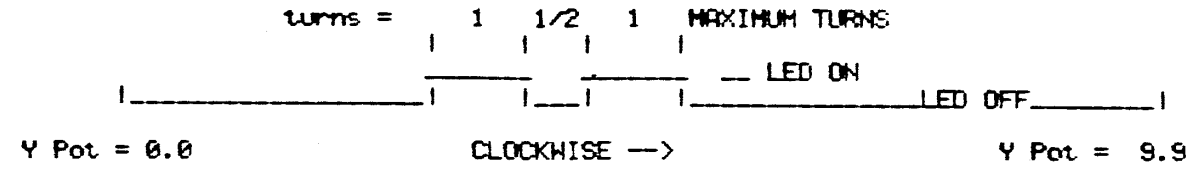
F. Note Y Pot reading.

G. Continue turning Y Pot CLOCKWISE until PCMPA LED goes OFF.

H. Note Y Pot reading. The difference between the two readings should be less than 1 turn.

I. Continue turning the Y Pot CLOCKWISE until the PCMPA LED goes ON again or the Pot is turned all the way around. If the LED turns on again, there should have been no more than 1/2 turn between OFF and ON.

J. If the PCMPA LED came on again, turn the Y Pot until the PCMPA LED goes OFF again. The LED should have been on for no more than 1 turn of the Y Pot.



If any test fails, check IC's U27, U28, U29, U30, and U2.

K. Return all controls to normal position.

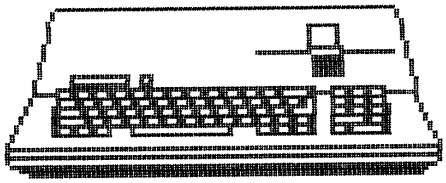
6-6 FINIS

468

17

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Analog Board Debug Test
(08 Dec 1983)

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 3

HDA

December 8, 1983

PROFILE ANALOG BOARD DEBUG TEST

EQUIPMENT:

1. SCOPE, TEKTRONIX 465 or equivalent 100 Mhz scope
2. SCOPE PROBES, x10, 3 probes, two w/ground
3. PROFILE, known good, without Analog board
4. PROFILE A/// I/O board and cable
5. A/// computer, M/// monitor
6. RIBBON CABLE, 12 inches long, Controller to Analog board
7. ADAPTOR CABLE, P6, P7, P8 extention from HDA
8. PIGGY-BACK Z8, with "Debugger Rev 11" Eprom
9. DISKETTE, "Disk Debugger Rev 17"
10. TRAY, Profile production, foam lined

STATION SETUP:

1. Remove top cover of Profile.

2. Remove Power Supply/Controller board, and HDA/Analog board assemblies from the Profile plastic base and put them in the Profile tray.

3. Remove ribbon cable from controller to analog board, set aside.

4. Turn HDA upside down so flat cover plate with WARNING label is down, and round end of HDA is toward controller board.

5. Connect long ribbon cable to controller board connector P2 from HDA.

6. Insert A/// I/O board in slot 4 (floppy drive side). Connect A/// I/O board to controller board P1 connector with I/O cable.

7. Remove plastic Z8 (341-0171) from Controller U25 socket and save.

8. Install ceramic piggy-back Z8 with "Debugger Rev 11" mounted on top into 40 pin socket at Controller board U25. Check pin 1 direction is correct (toward P1 connector).

9. Attach one scope probe with no ground clip to trigger input of scope, and other end to TP1 (near P2) on controller board. Set time base to 1 mS/div, External triggering.

10. Attach probe with ground clip to scope channel 1. Set to 1 volt/div, AC coupled.

11. Attach probe with ground clip to scope channel 2. Set to .5 volts/div, DC coupled. Set scope trace (grounded) to a reference line. Variations from this line will be used to adjust VCO pot, and show VCO problems later.

12. Plug in AC power cord to Profile power supply under controller board.

13. Insert Disk Debugger disk into A/// built-in drive and boot up the computer. Computer may be left on while testing and Analog board changes.

1-3

PROCEDURE:

4-10

1. Place Analog board to test on HDA. Plug in head connector J5 as the board is slipped in.

2. Plug P6, P7, P8 from the HDA into Analog board connectors J6, J7, and J8. Check that they are plugged into the correct connectors.

3. Install hold down screw at ground lug (near TP5). Ground strap from HDA MUST be attached here also. No other mounting is required.

4. Plug in ribbon cable from controller board to J2 on Analog board.

5. Turn ON Profile power switch at back of supply.

6. Attach channel 1 scope probe at Analog board TP1 or TP2 Read Amp, next to U7 (CA3127). Attach ground clip from scope probe to nearest ground test point.

7. Scope probe from channel 2 clip onto TP8 VCO, near U2 (7474).

8. Wait for the HDA stepper arm moving from "park" to track 0. The analog data signal on scope channel 1 should look like the photos of "good signal". If not, check that the spindle motor is turning, and that scope probe 1 is properly connected.

9. Break VCO pot on the Analog board (R56) loose, so it will turn freely. Set the VCO pot by setting scope channel 2 to ground, and adjusting to a ground reference line. Set back to DC coupling, and turn the pot from lock to lock. The VCO signal on the scope should move up and down at least .5 Volts.

10. Set the pot to ground reference line (VCO = 0 Volts +/-.25 Volts) and glue.

11. Type "RT60" to start at track 60. The stepper arms should move around about 1/2 turn.

12. Enter the command to start scanning the sectors, "R PE +S X" RETURN. The monitor will show that the track, head, and sector numbers are increasing. The stepper arm will move around counter-clockwise one track at a time.

13. Watch the read data on scope channel 1 and the VCO signal on scope channel 2, REJECTING the analog board for the following:

    Read signal = 2 Volts minimum (peak to peak).
    AGC signal = 1 Volt base to peak on at least one side (top or bottom). AGC shall show "bell" effect immediatly after sector gap periods.
    VCO signal = no excessive noise or oscillation.

14. If a data read error occurs, the monitor display will stop and say "ERROR - Press space key to continue". Record the track/head/sector, and the error number from the last data line of the monitor on the board reject tag.

15. Press the A/// space bar to continue. Stop the test if more than 7 errors occur or when the track number goes around to zero by pressing the ESCAPE key.

2-3

4X1

16. If any errors occured on the monitor, re-try that sector to verify by typing "RTxxHxSx" where "x" are the numbers of track/head/sector that error occured on. Then type "R PE X" RETURN. If the error appears continuously, the Analog board is defective. REJECT the analog board if any of the data errors appear consistantly.

17. IMPORTANT:

    Enter the command "RT99" RETURN, parking the heads off the data areas BEFORE ever turning OFF the HDA. Failure to do this will eventually destroy the disk surface causing permanent errors.

18. Turn OFF the power switch in back of the Profile power supply.

19. Remove the scope probes attached to the Analog board TP1 or TP2, and TP8. Unplug the connectors at J2, J6, J7, and J8.

20. Remove the ground wire screw and start to pull up the board from the J2 connector edge. Push off the HDA head connector when the Analog board is part way out.

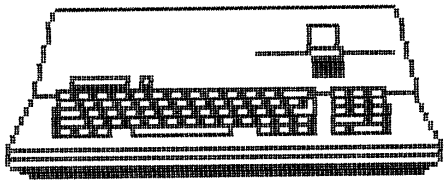
3-3 FINIS

477

18

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///

Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Analog Board Debug Test Repair Notes (08 Dec 1983)

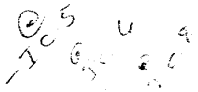
DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 1

PROFILE ANALOG BOARD DEBUG TEST December 8, 1983

REPAIR NOTES

For AGC:



1. Check C5, C10, C19 for damage.
2. Check IC U6 is not Signetics (NE592).
3. If AGC signal is less than 2 volts peak-peak:
   A. Change U7, then U6, then U8.
   B. Change Q1 (VCR2N).
   C. Remove DLY1 (Delay Line), if AGC signal improves, change DLY1.

For VCO:

1. Check IC U17, U19 are not Motorola 10131.
2. Check capacitors C28, C29, C30, C32, C33, C34, C35, C53 for cold solder connections.
3. Check RP2, RP3, RP4, RP5, RP6, RP7, RP8 are not Allen Bradley SIP resistor packs (white ceramic with black paint, 108A561).
4. If VCO OSCILLATES 200 mV or more, change U19, then U25.
5. If VCO is NOISY 200 mV or more, change U25, then U19.
6. If no VCO signal, change U16, then U14.
7. If VCO drifts up or down off scope, change U24.
8. If VCO will not adjust 500 mV, change U26, then R56.

For stop on error (04, 08, 14):

04: Read sector header was bad (track, head, sector address) usually read amp, AGC, or data separator related. (14 SAME AS 04)

CR 18 CK19 CA20

08: Read sector data was bad usually read amp, or AGC related.

1. Change U13, then U12.
2. Change U23, then U21.

For won't step (or can't find track 0):

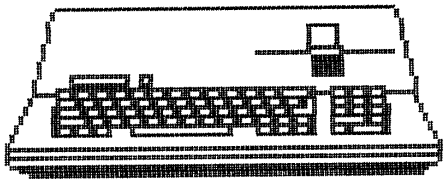
1. change U30.

19

434

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Analog Board Upgrade

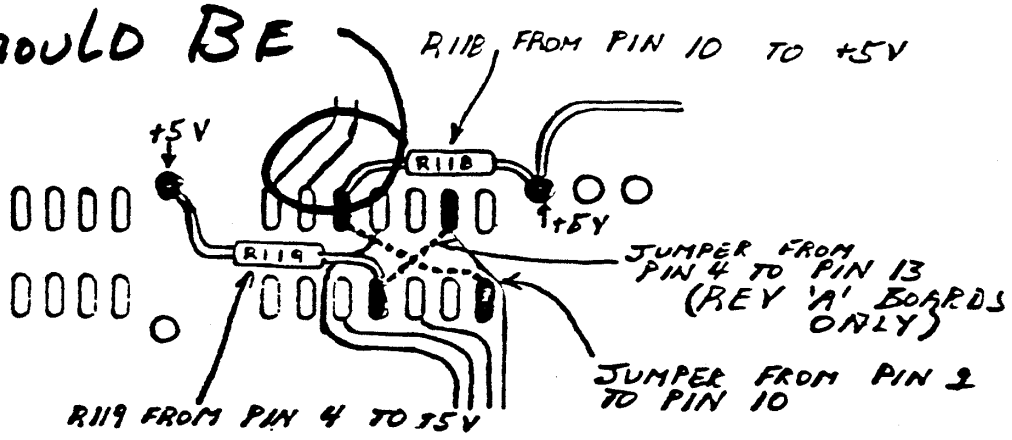
DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

44th PROFILE ANALOG BOARD UPGRADE

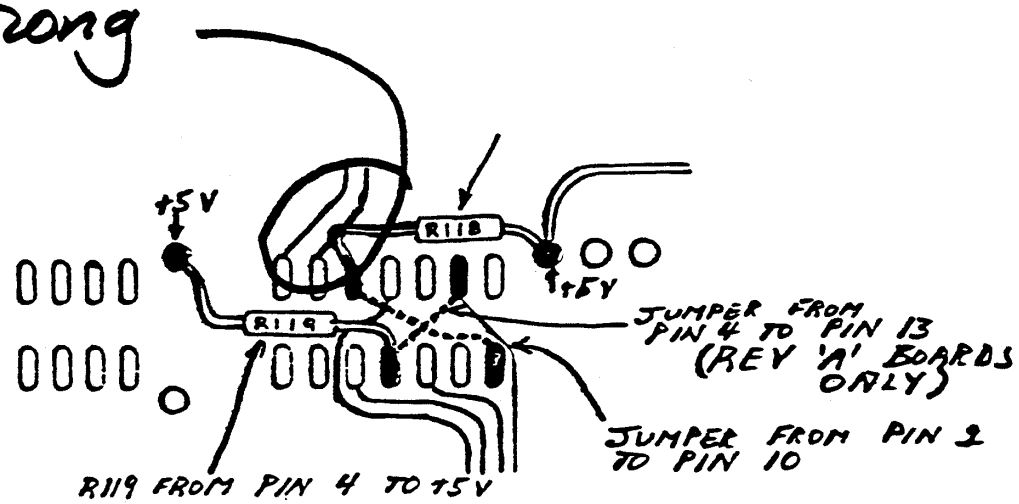
Pages: 2

UNDERSIDE OF BOARD - LOCATION U2

Should BE

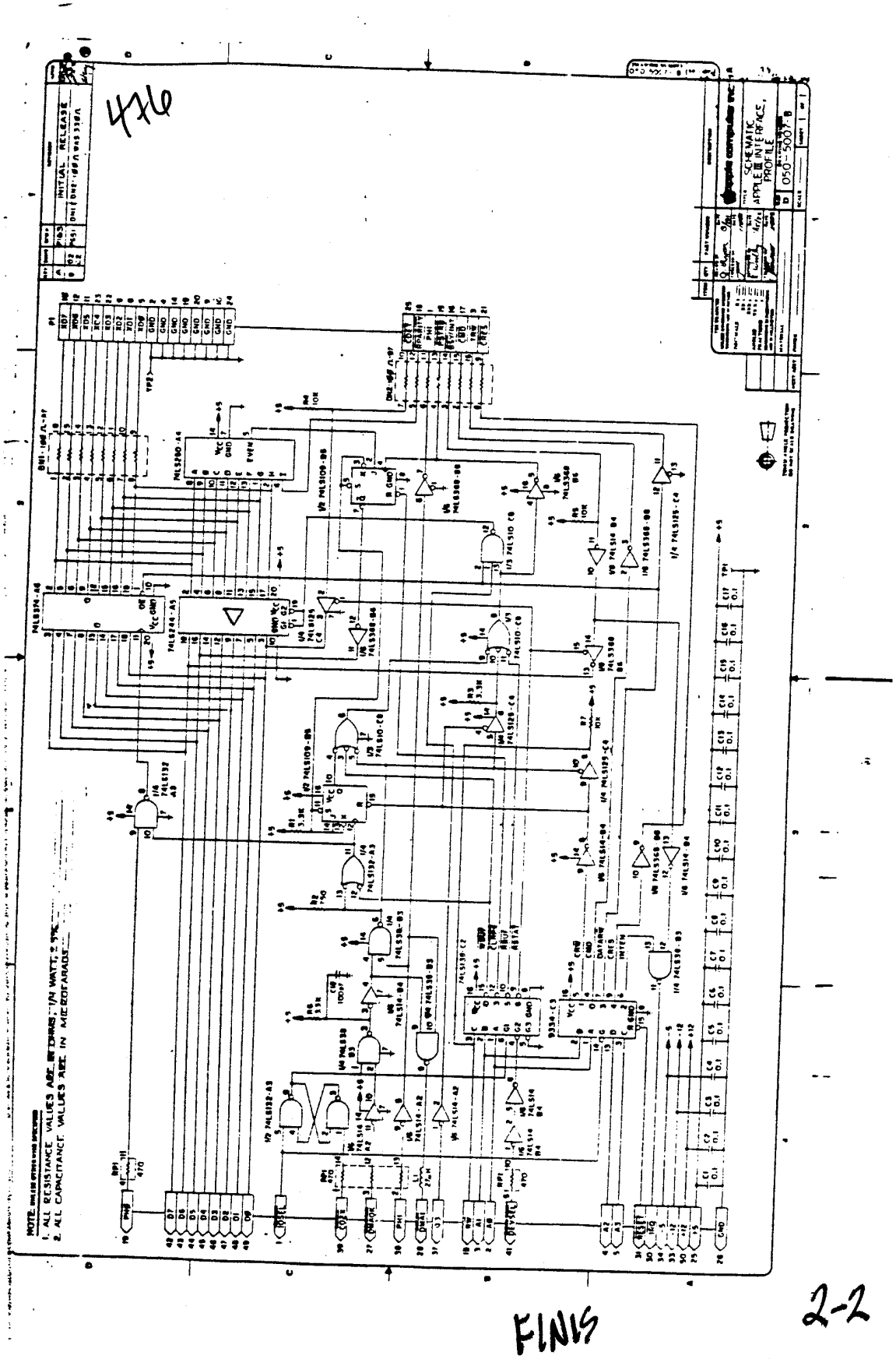


Wrong



Pin 10 Shorted To Pin 9 OF U2

1-2

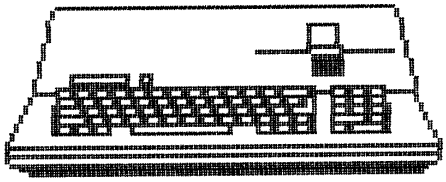


20

4XX

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

ProFile Interface Board Tester
Operation and Maintenance Manual

1982

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

4 of 6

Pages: 8

APPLE COMPUTER SERVICE ENGINEERING

PROFILE INTERFACE BOARD TESTER

OPERATION AND MAINTENANCE MANUAL

OBJECTIVE: To provide a test method to determine if the Profile Interface board is functional. This manual outlines the setup, operation, troubleshooting methods, and maintenance of the Profile Interface Board Tester.

1-8

479

TABLE OF CONTENTS

 ? missing

2-8

480

SECTION 1 - EQUIPMENT NECESSARY:

1. Apple II computer (48K) and disk drive or Apple /// computer with internal disk drive and Apple II emulation diskette.
2. Monitor.
3. Profile Interface Board tester. (Apple # 890-0183)
4. Interface tester test program diskette.
5. Media Tester Interface (VIA) board. (Apple # 890-6009)
6. VIA board test diskette.
7. Interface board schematic. (Apple # 050-5005B)
8. VIA board schematic. (Apple # 051-0194A)
9. Oscilloscope. (Tektronix 463 or equivalent)
10. Silentype printer. (optional)

3-8

4-61

SECTION 2 - SET-UP PROCEDURE:

2.1. Connect the two cables from the interface tester to J3 and J4 on the Media Tester Interface (VIA) card. The cables are labeled accordingly.

2.2. Make sure the computer power is OFF and plug the VIA card into Slot 2 of the computer. In the Apple II, the third slot from the left. In the Apple ///, the second slot from the left.

2.3. Open the ZIF (zero insertion force) connector on the tester and insert the interface board to be tested in it. Make sure the card is properly oriented by the short end of the interface board being at the same end of the connector as the board release handle. Close the handle, making sure that the board does not tilt or twist in the connector. Next, connect the DB-25 connector at the end of the tester cable to the corresponding socket on the interface board.

4-8

4682

Section 3 - OPERATION:

3.1. Booting the diskette will display the message:

LOADING PROFILE/APPLE III
INTERFACE CARD TEST PROGRAM

After about 30 seconds, the program will then ask you to plug the interface card to be tested into the test fixture by displaying:

AUTO TEST

INSERT INTERFACE CARD INTO TEST FIXTURE
AND PLUG IN THE DB-25 CONNECTOR

If not already done, connect the DB-25 cable from the test fixture to the card and make sure the board is properly inserted in the connector. The card will only fit one way properly as the connector handle will be blocked if it is backwards. The display will also prompt the test enable with:

TYPE "P" OR "S" TO TEST

P = LIST ERROR ON PRINTER (If available)
S = LIST ERROR ON SCREEN

3.2. The test will take about 15 seconds to run. If no errors are found, the computer will print:

NO PROBLEM FOUND

3.3. If any errors are detected, the computer will indicate the possible problem area and display the message:

PROFILE INTERFACE CARD TEST
IC LIST/LINES IN ERROR

This will be followed by a list of the incorrect signal lines on the board and/or the location of any associated ICs. The signal lines will be identified by the signal mnemonics used on them such as DMA1 or PSTRB. If an IC is determined to be the failure the device position will be called out, such as A2 or C4.

If the error is still in the board, it has to be either a bad line or faulty component other than an IC. You will need to repeat the test and look for the bad line with a scope. Go to step 3.4 in this test sequence.

5-8

1K67

3.4. If a fault is detected and displayed the computer will ask:

DO YOU WISH TO LOOP THROUGH TEST

This allows you to cycle the test program while looking for the error line with a scope. To do this, choose "Y". If you do not wish to do so, choose "N", and the program will display the beginning test prompt.

3.5. To stop the looping cycle, press ESCAPE. If the program has just started to cycle, pressing ESCAPE will not halt it, but just keep it from repeating itself. You will have to wait 15 seconds or so before getting the beginning prompt again.

SECTION 4 - ERROR ANALYSIS

4.1. The computer indicates the probable error by matching the detected error with a series of error messages in the test program. These error messages were developed by using gate isolation procedures, identifying all the individual gates in the circuit and the problems each one would cause if it failed. This is an extensive circuit mapping technique and will catch a majority of failures. However, if a board has massive failures in the way of shorted or open lines, the errors may not be able to be decoded by the computer.

4.2. This means that the computer can only match up the error with what it has on file. In the event of this happening, the computer will give all the failure listing associated with the detected errors, both component and/ or line.

If the computer did not associate any one error found with any of the error table entries in memory, it will indicate all the component and line failures detected. At first glance the printout may seem confusing but it is easy to understand if the individual failures are related to the schematic. All of the ICs on the board have several different circuit functions. If a device has massive failures all of these associated functions will be identified, giving a complex failure readout.

4.3. The easiest way to troubleshoot a problem like this is to replace the first IC identified in the printout and run the test again. If the problem is still in the board, swap out the second identified IC. Keep doing this until all of the ICs in the failure printout have been swapped. If the error table changes or indicates a repaired board, go back to the components first replaced. Try them again one at a time until all of them have been verified as either good or bad.

6-8

4-84

SECTION 5 - TESTER TROUBLESHOOTING PROCEDURES

5.1. In the event of a failure in the tester, the following procedures should be used by the technician to isolate, identify, and correct the faulty component.

5.2. Using a known good Interface Board, run the test sequence. Power down the computer and boot it up again using the other set of tester software. If the test results are different, swap the known good board for another known good board and determine if the copy of the tester software that gives the errors is bad. If this happens, throw away the bad diskette. Make another copy of the good software diskette before running the tests. Verify both copies by running the tests again.

5.3. To diagnose a possible failure of any device on the VIA card, run the VIA board tester. An Apple /// is needed to do this but first read the listing below and follow the instructions.

7-8

486

(5.3 - continued)

To enable the program, type RUN followed by a Carriage Return. The display will now show:

VIA BOARD TESTER

STEP-BY-STEP INSTRUCTIONS FOR TESTING VIA BOARDS:

1 OBTAIN TWO SHORT TEST JUMPERS (40 AND 50 PIN)
2 OBTAIN FULLY OR PARTIALLY STUFFED VIA BOARD TO TEST
3 JUMPER J1-J2 AND (OR) J3-J4 IN PAIRS - WHICHEVER IS APPROPRIATE
4 INSERT BOARD INTO SLOT 2 OF THE APPLE /// (SEE NOTE)
5 BOOT THE PROGRAM AND FOLLOW THE TEST INSTRUCTIONS
6 REFER TO THE VIA BOARD SCHEMATIC TO CHECK THE TEST RESULTS WHEN USING A PARTIALLY STUFFED BOARD

NOTE: IN CASE YOU HAVEN'T NOTICED, IT IS NECESSARY TO TURN THE APPLE ON TO SEE THE TEST INSTRUCTIONS. IT IS ALSO IMPORTANT TO HAVE THE POWER OFF WHEN THE BOARD IS INSERTED.

IF ALL THE PREVIOUS TEST INSTRUCTIONS HAVE BEEN FOLLOWED AND YOU'RE READY TO TEST, PRESS ALMOST ANY KEY AND THE TEST WILL START

These instructions are shown here to provide a check list to follow before powering up the Apple ///. The program will exercise all of the VIA board lines, looking for any shorts, opens, or data errors. It is a good idea to run this program on the board before testing any interface boards, and to run it in case a high error rate suddenly occurs. Like all software, several copies should be available in case one is destroyed.

5.4. To verify the proper operation of the tester, follow these steps. Test at least two known good boards with the tester. Using a good Profile, set up the Apple /// system and check out the Interface boards with the following criteria:

A. Read the Profile directory file.

B. Write to a file in the Profile, and verify that the write data was recorded.

If both of the boards passed the tester and work in the Profile, the tester may still have a problem. Remove any one of the ICs from the Interface board and test it again. The tester should indicate massive line failures in addition to the removed IC.

8-8

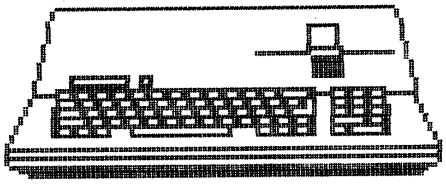
FINIS

21

4680

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Disk Debug Test User's Manual,
ProFile 5MB

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 21

apple computer inc.

DOCUMENT NO. 063-6006-A

PAGE 2 OF 22

1.0 TITLE: DISK DEBUG TEST USER'S MANUAL, PROFILE 5MB (P. 1 of 22 not present)

2.0 INTRODUCTION

The Formatter/Debugger program is provided as a service aid for troubleshooting block I/O devices such as the ProFile disk drive. It is a program that allows you to directly communicate with a block I/O device and exercise it through a series of unique commands or build simple test sequences that will assist in debugging the device.

It is a particularly helpful tool for servicing devices that have been returned because of a malfunction, etc. For example, the program can be used to detect an error and then by using the loop on error function, you can use an oscilloscope to analyse the circuit that caused the error.

You can use the program to service a drive containing either the Formatter/Debugger ROM or the Standard ROM. (The Standard ROM is the operational ROM that is shipped in the disk drive.) The command summary, section 3, specifies which type of ROM must be in the drive to use a given command.

2.1 HOW TO USE THIS PROGRAM

To use this debugger properly, you should be aware of some of the design concepts. In designing the user interface (command structure) it was decided to use single character commands. This permits you to type in the command very rapidly and use options (which are not always required) to modify certain test variables. The most used test variables are the three byte logical block that is treated as either a 24-bit number or as three 8-bit numbers. As a 24-bit number, the variable represents a standard logical block with a decimal range of 0-16 million blocks (Hex 000000-FFFFFF). As three 8-bit numbers, each variable has a range of 0-255 (Hex 00-FF).

The program makes no assumptions about the use of the test variable; it just gives you two ways to talk to it. You can consider it as a large (24 bit) number or as three smaller (8 bit) numbers. This variable is sent to the disk exactly as a 24-bit number which the firmware decides how to interpret. You, the user, must know how the firmware will react to understand what the firmware is doing with the block number.

2.2 TYPICAL COMMAND EXAMPLES

To help make the concept of this program more clear, here are some examples that show what the more common commands do.

R (Read)

This command requests data to be read from the unit being tested. It passes the command and the BLOCK variable to the unit and transfers the data from the unit to the input buffer (see the display command, section 3.2.3, for information on displaying the input buffer).

R by itself does not change the BLOCK variable; it uses its current value. However, you can change the block variable by adding a modifier to the command in the form of a number. This number is normally treated such that leading zeros are assumed, thus Ø, ØØ, ØØØØ, etc. all produce the 24-bit value ØØØØØØ. For example:

RØ or RØØ or RØØØØØØ, etc. (read block ØØØØØØ)
R13 or RØ13 or R ØØ13, etc. (read block ØØØØ13)

The above example is applicable only when using the Standard ROM. Notice that the BLOCK variable sent is either ØØØØØØ or ØØØØ13. The number 13 is the same as Ø13 or ØØØØ13 and produces the value ØØØØ13 in the BLOCK variable.

NOTE

The number 13 in the example is a HEX number, not a decimal number.

13 Hex = 19 Decimal

Remember that the firmware in the disk drive being tested determines how the BLOCK variable will be treated. Normally it is treated as a logical block.

In the next example, let's change the command to show how it would be used with the Formatter/Debugger ROM:

RT1H2S3 or RTØ1HØ2SØ3, etc. (read block Ø1Ø2Ø3)

Notice that the command now has 3 modifiers in the form of T and a number, H and a number, and S and a number. The modifier T (Track) references the first 8-bits of the BLOCK variable, the H (Head) references the second 8-bits, and the S (Sector) references the third 8-bits. Note that they are separate modifiers and can be used independently. For example:

If BLOCK = ØØØØØØ, then the command RT5 will set BLOCK = Ø5ØØØØ
If BLOCK = Ø5ØØØØ, then the command RH3 will set BLOCK = Ø5Ø3ØØ
If BLOCK = Ø5Ø3ØØ, then the command RT16S9 will set BLOCK = 16Ø3Ø9

Note that the last command in the example happened to affect the first and third bytes. Had the last command been RS9 then the BLOCK would have been Ø5Ø3Ø9. When all three modifiers are used, they must be in the T, H, S sequence.

REMEMBER! The numbers used are always Hexidecimal numbers.

W (write) ----------

This command requests data to to be written to the unit being tested. It passes the command and the BLOCK variable to the unit and transfers the data to the unit from the output buffer (see the buffer fill command, section 3.2.1, and the display command, section 3.2.3).

W by itself does not change the BLOCK variable; it uses its current value. However, as was the case with a read command, the BLOCK variable can be changed by adding a modifier. For example, the commands WØ, WT3H2S1, WH9, etc. have the same effect on the BLOCK variable that they do in the Read command.

\+ (plus)

This command increments the BLOCK variable. + by itself increments the variable by 1. For example, if the BLOCK variable was ØØØØ1F, then after + it will be ØØØØ2Ø (Note the hex numbers). +3 will increment the BLOCK by 3 each time. +3T will increment the first 8-bit variable by 3 each time. For example, if the BLOCK was Ø5Ø311, then +3T will change it to Ø8Ø311. H and S work the same way to modify the second and third groups of 8-bit variables.

To provide wraparound and carry, the + command will wrap a 24-bit number at FFFFFF. It will wrap T at 97 (97 increments to Ø) and will set H an S to Ø to provide full wrap. H will wrap at 3 (3 goes to Ø) creating a carry to T and S will wrap at F (F goes to Ø) creating a carry to H. Future enhancements will be to find out what the maximum block count for a device is and then wrap the 24-bit number at that count.

D (Display) ----------

This command displays the contents of the input or output buffers.

D by itself will display the input buffer.
DI is the same as D.
DO will display the output buffer.
DS will decode the contents of the input buffer as extended status information.

The form of the display is the same as that for an APPLE monitor memory dump. It will display the first 256 bytes of the buffer and pause so that you can study the values. At the bottom of the screen it asks for ESCAPE to terminate, RETURN to quit, ANY other key to continue. ESCAPE stops further display and cancels the rest of the command line effectively stopping processing and returning control to the user. RETURN quits the display command and causes the program to go on to the next command. Pressing any of the other keys, except the space bar, will allow the program to display the next 256 bytes of the buffer. Pressing the space bar will allow you to display the buffer line by line.

CAUTION

If you just keep hitting a key to see the next 256 bytes you will see first the buffer you requested and then memory above it. There are areas of ram above the buffers that will, when displayed cause your screen to get VERY sick and you will have to either re-boot the program or be very familiar with APPLE III hardware and monitor.

Just in case you are not familiar with the monitor memory display, here is an example. (The monitor doesn't display the descriptive information above the dashed line.)

```
  Hex
Address  Ø  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
-------------------------------------------------------
 8ØØØ:  ØØ 1A 23 92 FF FE A5 2F 33 34 ØØ ØØ ØØ ØØ ØØ ØØ
 8Ø1Ø:  12 11 34 87 FE 44 ØØ ØØ 3E 4C ØØ 2Ø A9 ØF C5 ØØ
 etc.
```

3.Ø COMMAND STRUCTURE

Each command consists of one or more characters and modifiers in command groups. Each group is separated by one or more spaces. The command line is one or more command groups. There is enough space in the command buffer for 255 characters.

Multiple command lines are possible by use of the Macro Add command.

3.1 COMMAND PROMPT

The program prompt consists of the following two lines:

```
S1 D1
COMMAND (B,C,D,F,G,H,I,L,M,N,O,P,Q,R,S,T,V,W,X,+,-,/,   ?) =>
```

The first line contains two flags, S1 D1, that tell which slot and drive the program is set to test.

The second line of the prompt consists of the first letter of each valid command. A list of the commands is always available by typing H or ? and pressing RETURN. In this manual you will see this written as H<nl> or ?<nl>. (The <nl> is the symbol for newline, often called RETURN.)

NOTE

After typing a command and pressing RETURN to execute it, you can press ESCAPE to terminate the command . However, if you wish to review the command that you just executed for the purpose of changing it, etc., you can type CONTROL A instead of ESCAPE.

3.2 DETAILED COMMAND DESCRIPTIONS

Following are descriptions of some of the commands that require more detailed explanations than others. In depth examples of some of the more commonly used commands such as Read, Write, Increment, and Display were provided in section 2.2. For an overall summary of all commands, refer to section 4.Ø.

3.2.1 BUFFER FILL

Syntax B[num] ---- where num is an optional 16 bit hex number (eg.,BØ123)

The buffer fill command is used to fill the output buffer with a specific data pattern. It can be used with either the Formatter/Debugger ROM or the Standard ROM installed in the drive. To use the command, type B followed by a number which will be treated as a 16 bit (2 byte) pattern. For example, the command

RØ B1234 WØ B45 W1<nl>

would read logical block-Ø, fill the output buffer with 12341234..., write block-Ø, fill the output buffer with ØØ45ØØ45..., and finally write this data to block-1.

3.2.2 CREATE SPARE

Syntax C<nl> ---- the C-command must be the only command on the line.

The create spare command is used to force a peripheral device to transfer a logical block to a new (spare) physical location on the device. The command will then request the block number to be spared and will then request confirmation from the user. If confirmed, the logical block will be spared by the device controller. This command should only be used when the Standard ROM is installed in the drive.

Using this command will allow you to fix flaky blocks found during testing.

The user will be asked for confirmation of this command. Respond with Y (Yes) or N (no).

NOTE

Do not attempt to use the create spare command when the Formatter/Debugger ROM, version D3.11 or earlier, is installed in your drive.

3.2.3 DISPLAY

Syntax D[I,O,S] ---- where I is input buffer, O is output buffer, and S is the extended status information in the input buffer.

The display command can be used with either the Formatter/Debugger ROM or the Standard ROM installed in the drive. It is used to display the contents of the input or output buffers. These buffers occupy the following Hex locations in the computer's memory:

input buffer ----8000H - 8213H
output buffer ---8300H - 8523H

The display will show 256 bytes at a time, scrolling to the next 'page' after each key press. To end the display and continue executing the command line, press RETURN. To abort the command line press ESCAPE. With the display on the screen, you can press the spacebar to cause a scroll to the first line of the next 'page'. Then, each time you press the space bar, the display will scroll one line forward.

The Display Status command works only slightly differently in that it decodes the input buffer into Extended status info about the spares and bad blocks and it will not stop until it is finished or until the space bar, RETURN key, or the ESCAPE key are pressed.

REMEMBER! The Display Status command will decode anything found in the input buffer so be sure to use the Get Status command first to make sure that the information is valid.

Before using the Display Status command with the Formatter/Debugger ROM installed, you must initialize the spare tables (see section 3.2.6) and then use the Get Status command, otherwise the display will contain garbage. With the Standard ROM installed, you only have to issue the Get Status command before displaying the extended status of the input buffer.

Here are some additional tips about using the Display Status command.

- After typing D[S] and pressing RETURN, press the space bar to stop the listing. Press the space bar again each time you wish to step through another line of the listing. Pressing any other key will cause a fast scan of the listing.

- Pressing ESCAPE will cancel the rest of the display function.

- Pressing RETURN will terminate the spare sector listing and start the bad block listing.

3.2.4 FORMAT

Syntax F<nl> ---- this command must be the only command in the command line.

The Format command is used with the Formatter/Debugger ROM installed in the drive. It is used to erase all old data from memory and lay down a new pattern of address and data.

After formatting a drive with the Formatter/Debugger ROM installed, you can type D[I] to get a list of the defective blocks. The list will end with FF FF FF FF. Refer to the documentation provided for the F/D ROM for further details.

WARNING

> This command is very dangerous and should only be used if damage to the address headers has occurred and only after every reasonable attempt has been made to recover other data from the device. The user will be asked to confirm this command. Respond with Y (Yes) or N (No).

Remember! This command will erase all previously recorded data.

Following is an example of the error message that will be displayed if you attempt to format a drive with a Standard ROM installed.

				I/O				ProFile					
		Block		R1	R2	R3	R4	S1	S2	S3	S4	S5	S6
FORMAT	S1 D1	ØØ	Ø3 5F	ØØ	ØØ	ØØ	ØØ	55	55	55	55	ØØ	ØØ

The 55's in this example are the error indicators.

3.2.5 GET STATUS

Syntax G<nl>

The Get Status command works only with the Standard ROM. It is used to obtain extended status information from the drive and place it in the input buffer. (You will need to execute the Display command to view the contents.) The Get Status command makes sure that the status you are going to view is valid.

When using a ProFile you can also get the status by reading block FFFFFF.

Form= A C 101 01

3.2.6 INITIALIZE SPARE TABLE

Syntax I<nl> ---- this command must be the first command in the command line.

The Initialize command is used by the ProFile Formatter/Debugger ROM to setup the spare tables in the ProFile drive. After the Debugger ROM has formatted the disk, the entire disk is available to read from or write on so that certification of the entire disk is possible. After the spare table sectors have been certified, the tables need to be initialized to allow the controller ROM to work properly. The Initialize command is not used when the Standard ROM is installed in the drive.

WARNING

This command is potentially dangerous as it effectively erases any old spared table data and if used incorrectly will cause the loss of valuable data on the ProFile.

The user will be asked to confirm this command. In response, type Y (Yes) or N (No).

4.Ø COMMAND SUMMARY

Following is a summary of all the commands that can be used with this program:

NOTE

Data shown in [] or () is optional.

BUFFER FILL B[<nl>] or B[num]

Fill the output buffer.

where num is a 16 bit hex fill number

used with both Formatter/Debugger and Standard ROMs

CREATE SPARE C<nl>

Force the drive to spare the specified block.

must be the only command on the command line

used only with Standard ROM

DISPLAY D[<nl>] or D[I] or D[O] or D[S]

Display the I/O buffers.

used with both Formatter/Debugger and Standard ROMs

FORMAT F<nl>

Format the device.

this is a dangerous command

used only with Formatter/Debugger ROM

GET STATUS G[<nl>]

Get extended status information.

Remember! When using the ProFile, RFFFFFF (read block FFFFFF) also returns status information

used only with the Standard ROM

HELP H[<nl>] or H[E] or H[E2] or H[char]

Print a list of commands, or errors, or a detailed description of each command.

where E is to display errors when the Standard ROM is installed

where E2 is to display errors when the Formatter/Debugger ROMis installed

where char is any legal command shown in this command summary

INITIALIZE SPARE TABLE I<nl>

Clear the spare block table.

this is a dangerous command

must be the first command on the command line

used only with Formatter/Debugger ROM

requires confirmation (Y or N)

LOOPON LPF[S,H,D]

Send Loop on Format commands to the firmware.

where S = sector marks
H = address headers
D = data field

LPF can be followed by any combination of S, H, and D

executed following a Read Track command to loop on format of the current track (eg., R[T] reads the T byte of the block variable to determine the current track)

used only with Formatter/Debugger ROM

MACRO M[A(Ø-9),C,L,Ø-9]

Use macro functions (alternate command lines).

where A = add current line to macro table
C = clear all macros
L = list macros

used with either Formatter/Debugger or Standard ROM

not effectively implemented at this time

N,O

Not implemented yet

PAUSE P[<nl>] or P[A(num),C,,E(num),N(num)]

Pause and wait for user.

Where A = any error/nonerror
C = clear any error/nonerror
E = on error
N = on no error
(num) is a 16-bit mask of S1 S2

Note: All four digits of the 16-bit mask must be turned on

used with either Formatter/Debugger or Standard ROM

QUIT Q[<nl>]

Return to calling routine (This module is a subroutine).

READ R[(num),T(num),H(num),S(num)]

Read a block.

where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

Remember! The firmware of the device tested determines how the number is treated. See the following example:

	T	H	S	- Formatter/ Debugger ROM
ProFile	ØØ	ØØ	ØØ	
	Hi	Mid	Lo	- Standard ROM

>0 means read to the output buffer (Eg., R23>0)

used with either Formatter/Debugger or Standard ROM

SCAN S<nl>

Order the firmware to scan the entire disk (read only).

Requires confirmation (Y or N)

after scanning, use D[I] to get list of any bad blocks found during scan

used only with Formatter/Debugger ROM

TURN OFF STEPPER T[<nl>

Turn off the power to the stepper motor.

used with Formatter/Debugger ROM but not normally used with Standard ROM

V

Not implemented yet

WRITE W[(num),T(num),H(num),S(num)]

Write to a block (same format as read).

where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

Remember! The firmware of the device tested determines how the number is treated. See the following example:

	T	H	S	- Formatter/ Debugger ROM
ProFile	ØØ	ØØ	ØØ	
	Hi	Mid	Lo	- Standard ROM

<I means write from the input buffer (eg., W23<I)

used with either Formatter/Debugger or Standard ROM

XECUTE X[<nl>] or X[E(num)]

Execute the current command line again.

where E = execute the line to this point on error
num = 16 bit error mask of S1 S2

X[E] will execute a function over and over again, if an error has occurred, up to the point where the error occurred

Form= A C 101-01

\+ +[(num),T(num),H(num),S(num)]

Increment the block number.

where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

\- -[(num),T(num),H(num),S(num)]

Decrement the block number.

where (num) = 24 bit number (8 bit if T,H,S)
T = first (Hi) byte of block
H = second (Mid) byte of block
S = third (Lo) byte of block

the - command does not decrement properly for H and S modifiers

/ /[S(num),T]

Choose options (/H for help).

where S = slot set
T = translate current block to cylinder/head/sector

?

Help.

5.Ø NEW FEATURES

The following new features have been developed since the release of DSKDBG V-EØØ.16 and are included in DSKDBG V-EØØ.17.

* The Help errors command has been slightly modified. HE or HE1 provides help for the Standard ROM and HE2 returns help for the Formatter/Debugger ROM.

* The Help command lists the form of the extended help commands.

* Display now has a command to allow the decoding of the status table into the version, spare list, and bad block list.

  DS = display status info.

* Use ESCAPE or RETURN to terminate a long list of spares or bad blocks.

* An additional modifier is available for read/write commands that allows you to specify which buffer you are reading to or writing from. For example, a read command followed by >O means read to the output buffer. A write command followed by <I means write from the input buffer.

6.Ø TYPICAL PROBLEMS AND HOW TO AVOID THEM.

* The + (Increment) command doesn't work.

This is to date the most common problem. Usually the command line will look something like

```
RØ PE + X
```

What you were trying to do is sequentially read thru the disk starting at block Ø. What you said was

```
Read block Ø (block = Ø), Pause on error,
Increment block (block = 1), Repeat line,
Read block Ø (block again = Ø) ........
```

To fix this problem just do one line to set block Ø and then go on to the next line to do the Read, Pause on error etc. For example:

```
RØ<nl>
R PE + X<nl>
```

* Display Status Won't Work Or Won't Stop

The display status command allows you to display the input buffer and make assumptions about what the data means. For it to work, the Get status command must successfully read the status informtion from the drive. If the input buffer has garbage in it, the Display Status command may go round and round trying to display all the spared sectors and bad blocks.

To stop the command when it is caught in a loop or is listing a bunch of bad blocks, simply press the RETURN key or the ESCAPE key. The functions of the other keys are the same as for all other display commands.

504 apple computer inc.

DOCUMENT NO. 063-6006-A

PAGE 19 OF 22

7.Ø KNOWN BUGS IN VERSIONS EØØ.17 AND EØØ.16

The Macro command has 3 known bugs. The first occurs when a macro is added to the list and there is already one there of that number. The effect is to make both of them disappear. The second bug occurs when the macro is invoked on a line with the X command (eg., M1 X). This disables the ESCAPE key so the program can only be halted by re-booting. The third bug also involves the X command. When a macro is added (eg., R MA1 X), it will keep asking to delete the old macro.

Pause on any error flag (P on a white background) doesn't work.

The - command doesn't decrement properly for H and S modifiers.

Form A.C. 101.01

505

apple computer inc.

DOCUMENT NO. 063-6006-A

PAGE 20 OF 22

8.0 STATUS BYTE DESCRIPTIONS

This section provides a bit-by-bit description of the four status bytes available with the Standard ROM or the Formatter/Debugger ROM.

8.1 STATUS BYTES WITH STANDARD ROM V-3.98 INSTALLED

STATUS 1

7 = 1 if ProFile did not receive 55 to its last response
6 = 1 if write or write/verify was aborted because >532 bytes if data were sent
if 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/A
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/A
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/A

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/A
3 = N/A
2 = 1 if ProFile was reset *
1 = 1 if ProFile gave a bad response *
0 = 1 if parity error *

* These bits are set by the SOS ProFile driver and are not used by the Dskdbg ProFile.IO.

Form: A.C. 101.01

STATUS 4

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

7.2 STATUS BYTES WITH FORMATTER/DEBUGGER ROM FD3.98 REV 11 INSTALLED

STATUS 1

7 = 1 if Profile did not receive 55 to its last response
6 = 1 if no index found during formatting
5 = 1 if no sector mark found during formatting
4 = 1 if SEEK ERROR - unable to read 3 consecutive headers on a track (one try only)
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/A
Ø = 1 compare error on a write compare

STATUS 2

7 = 1 if ProFile has been reset
6 = 1 if track number invalid while reading or writing a sector
5 = N/A
4 = N/A
3 = N/A
2 = N/A
1 = 1 if seek to wrong track occurred
Ø = N/A

STATUS 3

D7, D6, and D5, along with D4 and D3 from STATUS 1, tell why a write compare operation failed.

	D7	D6	D5
write timeout	Ø	Ø	Ø
read timeout	1	Ø	Ø
read CRC	1	1	Ø
data compare	1	1	1

STATUS 4

7 - Ø = the number of errors encountered when formatting data fields or scanning the disk.

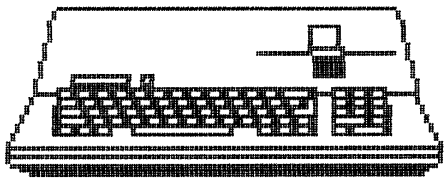
FINIS

508

22

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Formatting the ProFile HDA

25 Feb 1983

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 8

509

Profile Level 2 Technical Procedures

Contents:

(25 Feb 1983)

1-8

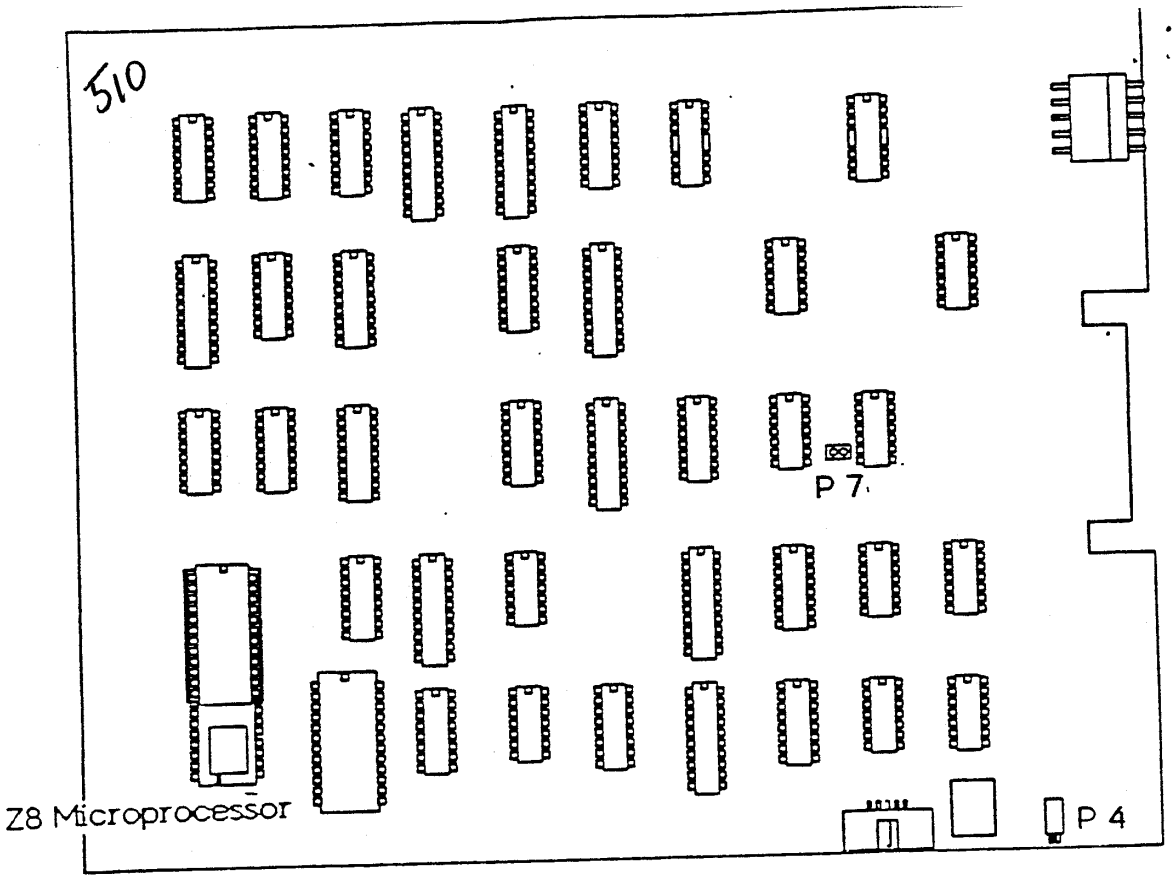


Figure 1 - Profile Controller Board

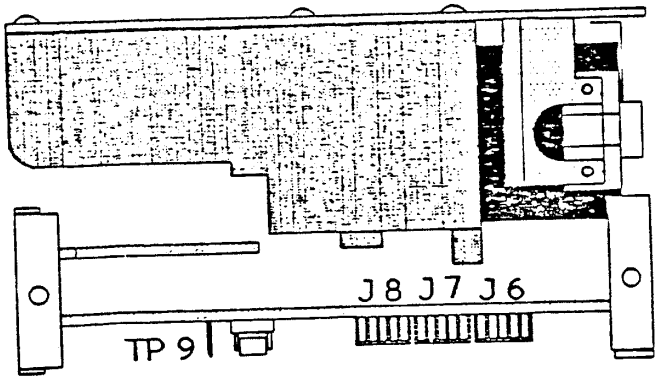


Figure 2 - HDA End View

2-8

511

FORMATTING THE PROFILE HDA

Equipment Necessary:

Format Firmware - Rev 10
Jumper Wire
Profile Format Diskette - Rev 12
Profile Final System Test - Rev 23
Frequency Counter
Small Screwdriver
IC Extractor
Apple /// w/Profile interface and cable
Known good Profile System minus the HDA.

Procedure:

1. Place the HDA to be formatted into a known good Profile system. The controller should contain the piggy-back development type Z8 (shown in Figure 1).

2. Remove the System ROM from the piggy back on the Z8 then install the Format Firmware. Locate the connectors J6, J7 and J8 (shown in Figure 2) that connect the HDA to the analog board. They are a constant source of problems. Move the wires around and make sure that they are securely fastened to the connectors. Connect an external LED to P4 (shown in Figure 1).

3. Power up the Profile and allow 1 minute for the drive speed to stabilize.

4. To check the speed of the HDA, set up the frequency counter to display milliseconds, and connect the input of the frequency counter to TP9 on the analog card. The frequency counter should read 16.67 milliseconds +/- .5%. If it does not, adjust R3, on the side of the HDA (shown in Figure 3) as close to 16.67 milliseconds as possible.

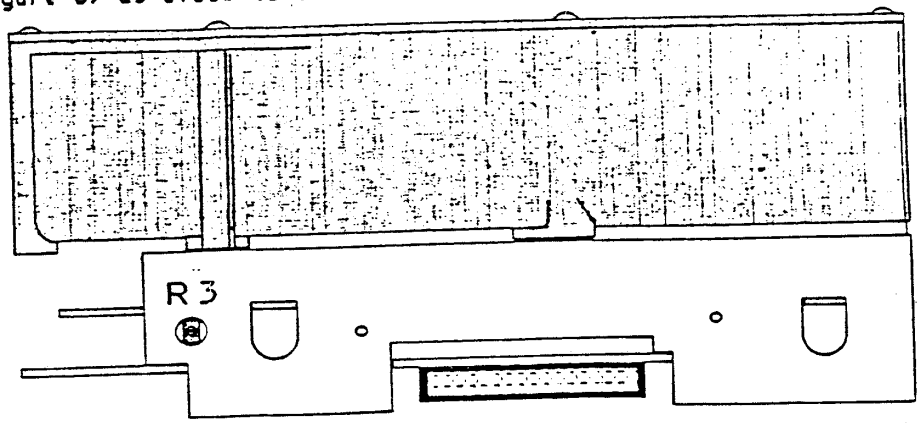


Figure 3 - HDA Side View

3-8

512

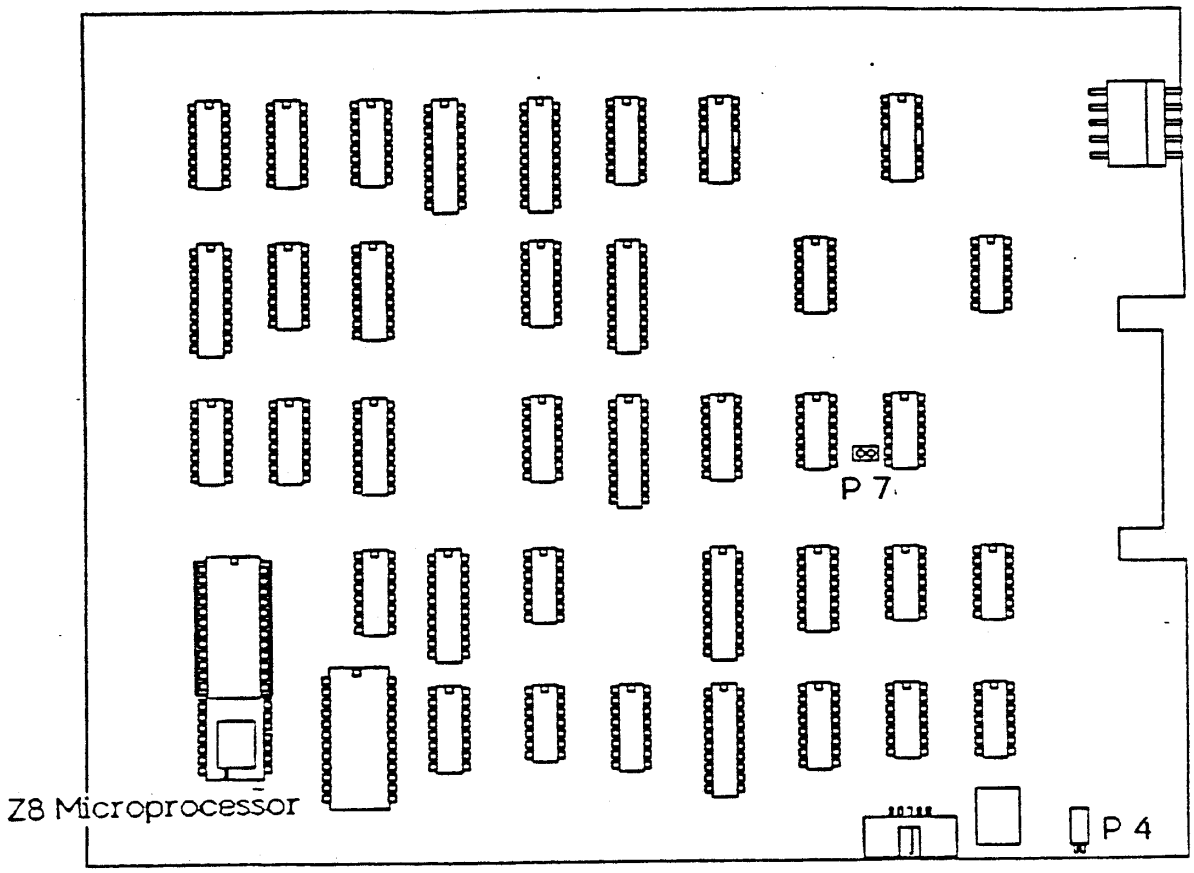


Figure 1 - Profile Controller Board

4-8

513

5. Install the Profile interface card into slot 1, and a Silentype Printer into Port A of the Apple ///. Connect the cable between the interface and the Profile.

6. Boot the Format/Certify Test diskette. When the program is loaded, the following message should appear:

   PRESS "RETURN" WHEN PROFILE IS READY

7. After the <RETURN> key is pressed, the following message will appear:

   INSTALL JUMPER, PRESS ANY KEY TO CONTINUE

8. Install a jumper between the two pins at P7 on the controller card and then press the <RETURN> key. Although the prompt says that you can press any key to continue, only the <RETURN> key will work.

9. When the <RETURN> key is pressed, the LED should start to flash and the stepper motor will step outward. The HDA surface is now being formatted.

10. After approximately 3 minutes, the HDA will be formatted and the following message will appear:

    REMOVE JUMPER, PRESS ANY KEY TO CONTINUE

    Remove the jumper wire from P7 on the controller card. Although the message again states that you may press any key to continue, only the <RETURN> key will work.

11. When the <RETURN> key is pressed the Profile will scan, certify and initialize the spares table on the HDA surface. The results of each process are reported on the printer, ending with the following pass or a fail message for the completion of the test:

    TEST COMPLETED, SYSTEM PASSES (FAILS)

    If the test fails, the HDA is defective and should be returned.

12. Power off the Profile and replace the Format Firmware with the 3.98 System Firmware (P/N 341-0008B) or with the masked Z8 microprocessor (P/N 341-0080B)

13. Power on the Profile, and observe that it goes though the power up sequence. When the LED is steadily on, boot the Profile Final System Test.

5-8

514

14. After the Final Test has booted, observe the printout on the Silentype. The following message is printed:

    CONTROLLER VERSION NO. D3.98

    Directly below, the printer will print:

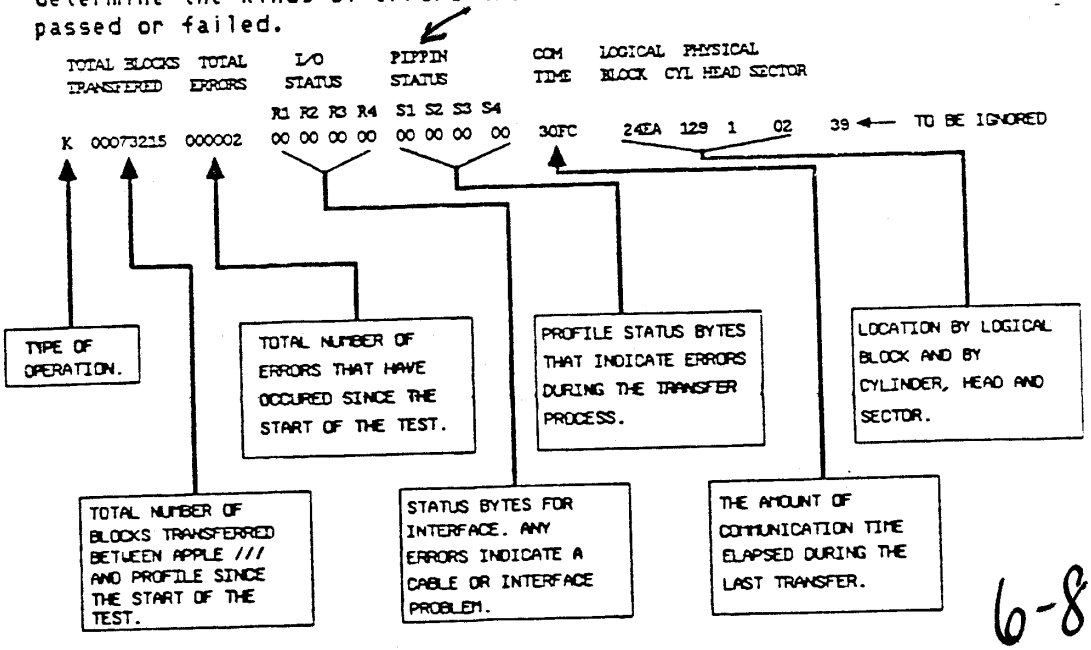
    SPARE SECTORS BAD BLOCKS
    XX XX

    The number under "SPARE SECTORS" (this should be SPARED SECTORS) is the number of sectors that have already been spared by the Profile. A spared sector is a logical block that has been assigned a new physical location because the old physcial location has reached an unaccectable error rate threshold during transfer operations. There are 32 spare sectors on the Profile that can be used for this purpose.

    The number under "BAD BLOCKS" is the number of blocks that cannot be accessed at all. Any number here indicates a potentally bad Profile. If there are any bad blocks listed here, run the test and match it to the FST Results Chart at the end of this procedure.

15. After a 5 second pause, the final test will begin transferring blocks of data between the Apple /// and the Profile. Figure 1 illustrates the monitor display. The bottom line of the header is the status line, which returns the inforamtion about the current block being transferred. When an error occurs, the status will be printed on the bottom line of the monitor and on the Silentype.

16. After the test has completed 500,000 block transfers (about 24 hours), the test is completed. Use the FST Results Chart on the next page to determine the kinds of errors that have occurred and if the unit has passed or failed.



6-8

Figure 4

515

Profile Status Bytes Description

STATUS BYTE 1

7 = 1 if Profile did not receive 55 to its last response
6 = 1 if write or write/verify was aborted because more than 532 bytes of data were sent or Profile could not read the spares table.
5 = 1 if host data is no longer in RAM because Profile updataed its spares table.
4 = 1 if SEEK ERROR - unable in 3 trys 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 was unsuccessful

STATUS BYTE 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 (greater than 100 bad blocks in table.
3 = 1 if Profile unable to read its status sector.
2 = 1 if sparing occured.
1 = 1 if seek to wrong track occurred
0 = N.C.

STATUS BYTE 3

7 = 1 if Profile has been reset.
6 = 1 if block number is invalid.
5 = 1 if block ID 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 occured*

STATUS 4

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

*This bits are set by the Profile driver.

7-8

5.14

FST RESULTS CHART

The Profile under test FAILS if:

1. There are 26 or more soft errors.
   a. To identify a soft error look at the Profile Status bytes.

```
S1  S2  S3  S4*   *S4 is a decimal number
00  00  00   09 <-- S4 is less than 10
```

2. There are 6 or more hard errors on the Silentype printout of the following type:

```
S1  S2  S3  S4*
00  00  00   10 <-- S4 is equal to or greater than 10.
```

3. There are 2 or more lines on the Silentype printout that are seek errors:

```
S1  S2  S3  S4*
00  02  00  00
```

4. There are ANY of the following errors or the blocks transferred counter stops counting:

   UNIT FAILS
   BUFFER COMPARE ERROR
   INVALID RANDOM SEED

   Any error with an S1 that has an odd value.
   Any error where S1 equals 04
   Any error where S1, S2 or S3 has value of FF.

5. After the Profile Final System Test as been rebooted:

   The number of SPARE SECTORS is 16 or more.
   The number of BAD BLOCKS is 1 or more.

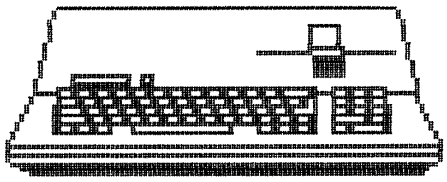
8-8 FINIS

517

23

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Power Supply, ProFile, Class A
Specification

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

ZONE	REV		APPD
A		INITIAL RELEASE	
B	P181	Revised to include Monitor CRT and Mechanical Equivalency	

518

Pages: 7
- Sheet 1 of 1
- Sheets 1 of 6

This assembly meets Apple Computer Specification #062-0074, and includes the equivalent circuitry for Apple assy #656-4106, "PCB, Assy, Monitor", Tested, is mechanically equivalent to #656-5102, "Subassy, Power Supply", and is purchased only from U.S. ASTEC as ASTEC part number AA11771.

699-0059-B

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. DECIMALS .X ± .XX ± .XXX ± ANGLES XX.X ± FRACTIONS ± DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS.	DRAWN BY Jamie Frederick DATE 8-81	apple computer inc.	
	CHECKED BY RRM DATE 3-30-82	TITLE Purchased Assembly Power Supply, PROFILE CLASS "A"	
	APPROVED BY DATE 83184		
	RELEASED BY DATE	SIZE A	DRAWING NUMBER 699-0059-B
NEXT ASSY.	MATERIAL: FINISH:	SCALE:	SHEET 1 OF 1

REV.	ZONE	ECO #		APPD
A		806	INITIAL RELEASE	
B		P120	Deleted pg. 4, AC Line Monitor Elec. Req.; also pgs. 7-9, dwgs & schematic	
C		P171	Dimension change (page 6) 3.75 was 3.65	

59

1.0 ELECTRICAL CHARACTERISTICS

1.1 INPUT VOLTAGE: 115 VAC or 230 VAC
Selected by jumper on pcb.
47 to 63 Hz

1.2 OPERATING RANGE: 90 to 135 VAC RMS
180 to 270 VAC RMS

1.3 CONVERSION EFFICIENCY: 75% minimum acceptable; with 78% as a target in production.

1.4 DELIVERED POWER: 30 watts steady state.
55 watts starting for a minimum of 14 seconds.

1.5 OUTPUT VOLTAGES AND CURRENTS:

$Vout_1$ +12 VDC ± 6% 1.5 Adc steady state and 3.5 to 4.0 amps for 10 - 14 seconds.

$Vout_2$ +5 VDC ± 2% 2.0 Adc continuous.

$Vout_3$ -12 VDC ± 6% 0.1 Adc continuous.

1.6 RIPPLE AND NOISE CONTENT; OUTPUT: 50MVP-Pon +5VDC;100MV P-P, +12 VDC. 1Hz to 10 MHz.

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. DECIMALS .X ± —— .XX ± —— .XXX ± —— ANGLES XX.X ± —— FRACTIONS ± —— DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS.	DRAWN BY [illegible] DATE 7-7-81	apple computer inc.	
	CHECKED BY MG DATE 7-81	TITLE POWER SUPPLY, PROFILE	
	APPROVED BY W. Dirks DATE 7/7/81		
	RELEASED BY [illegible] DATE 7/8/81	SIZE A	DRAWING NUMBER 062-0074-C
MATERIAL: ——			
NEXT ASSY.	FINISH: ——	SCALE: ——	SHEET 1 OF 6

CLASS "A"

520

1.7 OPERATING TEMPERATURE: 0 to +70 °C (Ambient)

1.8 STORAGE TEMPERATURE: -20 to +85°C.

1.9 PROTECTION CIRCUITS: The input must be protected by a fast blow fuse and a thermister inrush current limiter.

The +5 volts d.c. TTL voltage must be protected from over voltage output by means of an active crowbar.

All three d.c. outputs must be short circuit capable for an indefinite period.

1.9.1 HOLD UP TIME: 20 msec nominal at 30 watts load.

1.9.2 TEMPERATURE COEFFICIENT: 0.02%.

1.9.3 AC ISOLATION: to safety ground and A.C. input to output 4.5KVDC

1.9.4 OUTPUT TO SAFETY GROUND: 0.5KVDC

1.9.5 INSULATION AC TO GROUND: 50 MEG ohm nominal.

1.9.6 LEAKAGE CURRENT: 240 VAC input
IRMS ≤ 3.5 ma RMS

1.9.7 LINE CONDUCTED EMI: FCC 20780 limits. See attached specifications

1.9.8 SAFETY APPROVALS: UL and CSA required.
VDE required after 9/81.

2.0 MECHANICAL REQUIREMENTS:
The supply shall conform to attached envelope.

2.1 THERMAL: The power supply shall be capable of operating under all conditions of line and load at 0-70°C continuously.

2.2 STORAGE TEMPERATURE: -20 to +85°C.

2.3 HUMIDITY: Operating: 95% RH @ 35°C.
Storage: 95% RH @ 50°C.

2.4 VIBRATION: 10 to 500 Hz double sweep at 1 active per minute with pk-pk excursion of 1.5mm or 10g acceleration.

2.5 RANDOM DROP: 45 min. at a rate of 5 RPM.

2.6 BURN IN: A minimum 24 hour burn in at low line, full DC load at 70°C is required. Vendor will burn in all units.

apple computer inc.

SIZE	DRAWING NUMBER
A	062-0074-C
SCALE: ———	SHEET 2 OF 6

521

2.7 SERIALIZATION: All supplies shall have a serial number affixed and recorded so that test and failure records can be tracked throughout the life of the product.

2.8 SAFETY REQUIREMENTS: UL 478
UL 1201
CSA 22.2 No. 154
VDE after 9/81

2.9 INDUCTORS

No solenoidal filter inductors should be used in this product.

2.95 INPUT AND OUTPUT CONNECTOR: Molex Connector Pin Designation.

A.C. Input	D.C. Connector
1. AC Neutral	1. Reset Monitor Output
2. Key	2. Key
3. AC Line	3. -12V
	4. +12V
	5. +12V
	6. Common
	7. "
	8. "
	9. "
	10. Common
	11. +5V
	12. +5V
	13. +5V

Mating Molex Connectors:

DC P/N 09-50-3131

AC P/N 09-50-3030

The output connector will be a single, in-line connector combining the above two part numbers.

apple computer inc.

SIZE	DRAWING NUMBER
A	062-0074-C
SCALE: ———	SHEET 3 OF 6

SPECIFICATION: ELECTROMAGNETIC COMPATIBILITY

1. Emissions: Applicable assemblies, subassemblies and peripheral devices shall be 6 dB below limits of (1) Federal Communications Commission (FCC) Part 15 Sections 15.830 (radiation limit) and 15.832 (conduction limit) for units operated from 60 Hz line voltage and (2) VDE 0871/6.78 section 3.2.1 (conducted) and sections 3.2.2 and 3.2.3 (radiated) for units operated from 50 Hz line voltage. Those units rated 50/60 Hz shall meet both requirements.

The upper frequency limit for both FCC and VDE conducted emissions limits is 30 MHz. However, due to radiated emissions from the AC power cord the Apple conducted limit is extended to 60 MHz.

For convenience, limit amplitudes, less 6 dB, are reproduced herein. (Tabulated limits are Apple Computer EMI limits; FCC and VDE limits are 6 dB higher). However, the in-effect version of FCC part 15 or VDE 0871 are the binding documents.

RADIATED

	Frequency Range (MHz)	Field Strengths (uV/m)	(dB uV)	Distance (meters)
1. FCC Part 15	30-88	50	34	3
	88-216	75	38	3
	216-1,000	100	40	3
2. VDE 0871	0.01-30	20	26	30
	30-470	20	26	10
	470-1,000	80	38	10

CONDUCTED

	Frequency Range (MHz)	Voltage (uV)	(dB uV)	LISN Impedance
1. FCC Part 15	0.45-60	125	42	50
2. VDE 0871/6.78	0.01-0.15	*		150
	0.15-0.50	200	46	
	0.50-60	100	40	

*straight line from 10 kHz (3.5mV, 71 dB V) to 150 kHz (300uV, 50 dBuV)

Test and measurement equipment and procedures shall be as specified in applicable specifications. Final acceptance tests are performed with assembly or peripheral installed in system intended to be marketed with; such system to consist of the basic Apple Computer (II, III, etc.) and full memory installed and as many peripheral devices (disk drives, printer, monitors) and optional components (language card, serial card, parallel card, etc.) as possible to simulate worst-case operating conditions as closely as possible. Qualification tests with "remote exercisers", generators or other manufacturer Personal CPUs are unacceptable.

II. SUSCEPTIBILITY (to be determined).

Under consideration: The device shall not have uncorrectable data errors when subjected to the following field strengths or voltages - Irradiated: 0.01-1,000 MHz, 5V/m (100% modulated with 1 kHz square wave).

apple computer inc.

SIZE	DRAWING NUMBER
A	062-0074-C
SCALE: ——	SHEET 4 OF 6

523

II. SUSCEPTIBILITY (to be determined) - continued.

Transient Line Noise:** 1. Class A products: 400V pulse with 100 nonosecond width and 10 nsec risetime.

2. Class B products: 200V pulse (same characteristics)

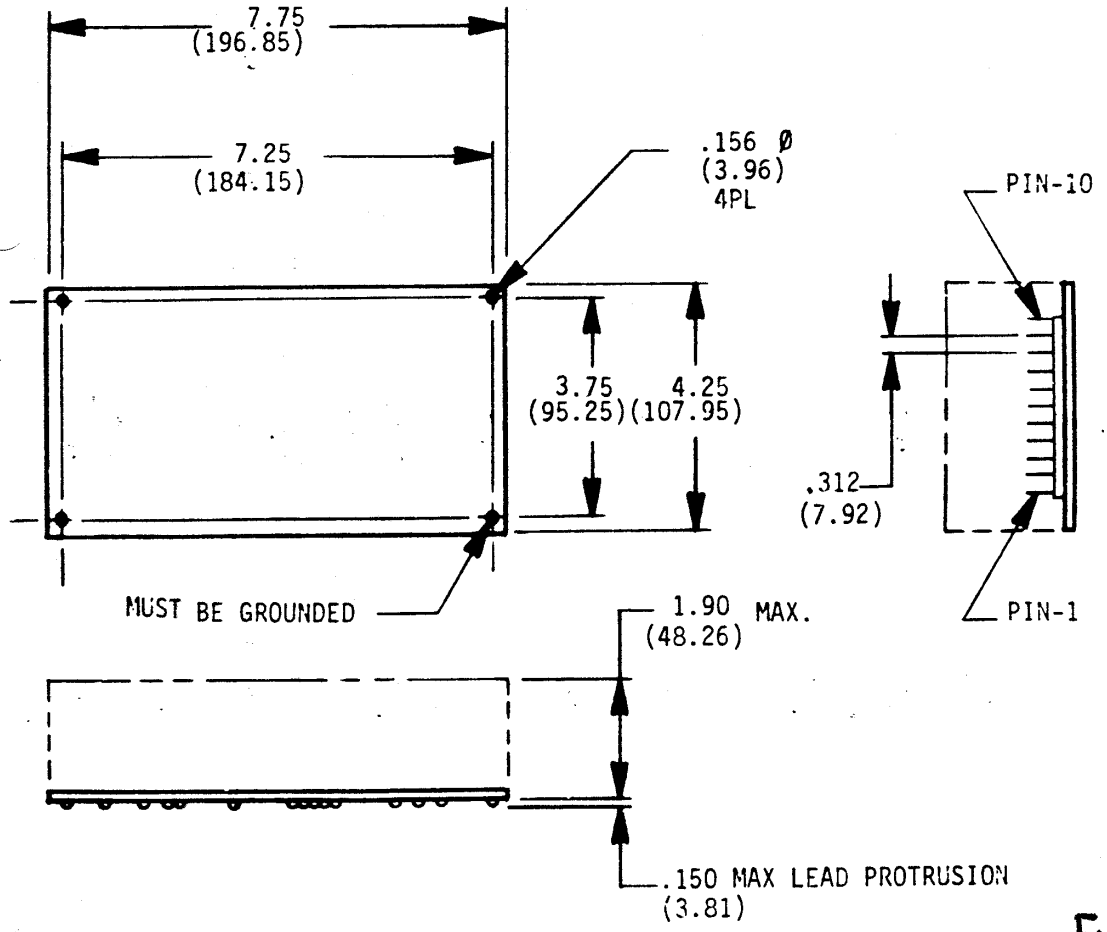
Conducted RF: 0.01-100 MHz: 3V rms.

**no soft errors allowed.

apple computer inc.

SIZE	DRAWING NUMBER
A	062-0074-C
SCALE: ———	SHEET 5 OF 6

524



FINIS

apple computer inc.

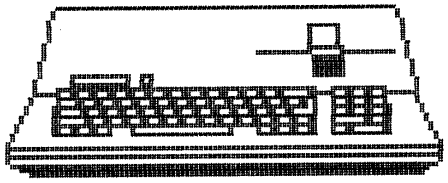
SIZE	DRAWING NUMBER
A	062-0074-C
SCALE: ——	SHEET 6 OF 6

525

24

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk / Drive

DOCUMENT TITLE

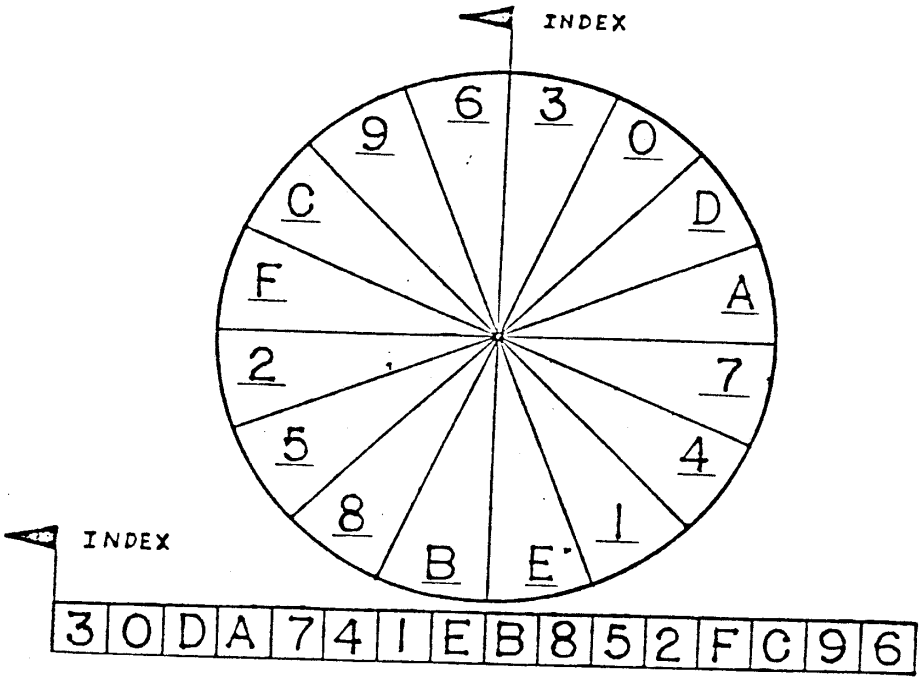
ProFile Sector Interleave Charts

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

524

Pages: 2

Apple /// ProFile HD Sector Interleave Charts

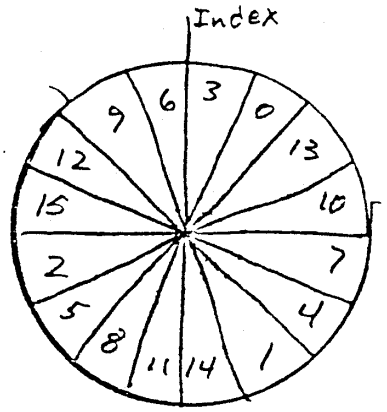


1-2

52X

2-2

0 - 0000	8 - 1000
1 - 0001	9 - 1001
2 - 0010	A - 1010
3 - 0011	B - 1011
4 - 0100	C - 1100
5 - 0101	D - 1101
6 - 0110	E - 1110
7 - 0111	F - 1111



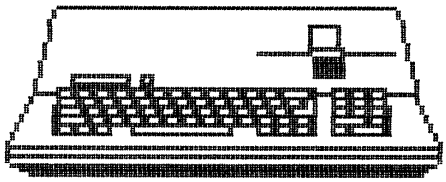
FINIS

528

25

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

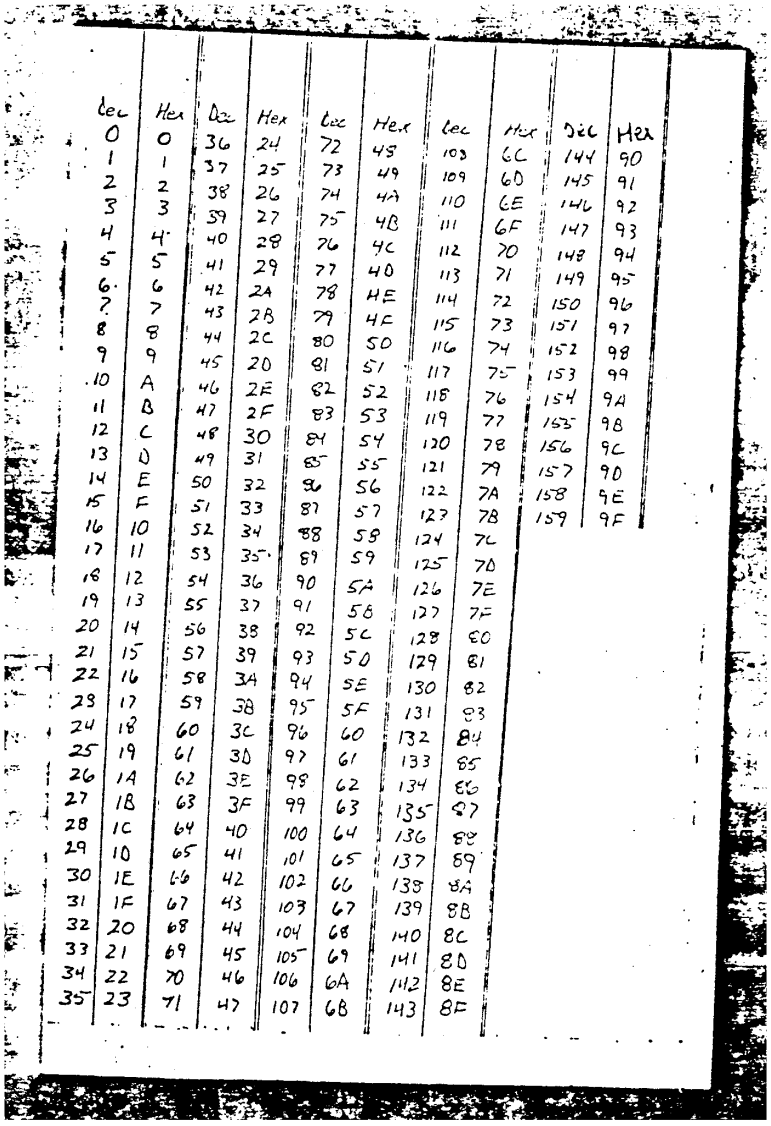
DOCUMENT TITLE

ProFile Status Information

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

529 Apple /// ProFile HD: Status Info

Pages: 2

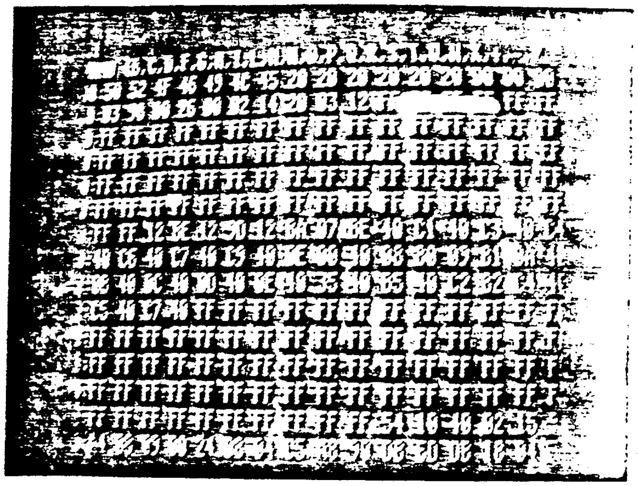


Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
0	0	36	24	72	48	108	6C	144	90		
1	1	37	25	73	49	109	6D	145	91		
2	2	38	26	74	4A	110	6E	146	92		
3	3	39	27	75	4B	111	6F	147	93		
4	4	40	28	76	4C	112	70	148	94		
5	5	41	29	77	4D	113	71	149	95		
6	6	42	2A	78	4E	114	72	150	96		
7	7	43	2B	79	4F	115	73	151	97		
8	8	44	2C	80	50	116	74	152	98		
9	9	45	2D	81	51	117	75	153	99		
10	A	46	2E	82	52	118	76	154	9A		
11	B	47	2F	83	53	119	77	155	9B		
12	C	48	30	84	54	120	78	156	9C		
13	D	49	31	85	55	121	79	157	9D		
14	E	50	32	86	56	122	7A	158	9E		
15	F	51	33	87	57	123	7B	159	9F		
16	10	52	34	88	58	124	7C				
17	11	53	35	89	59	125	7D				
18	12	54	36	90	5A	126	7E				
19	13	55	37	91	5B	127	7F				
20	14	56	38	92	5C	128	80				
21	15	57	39	93	5D	129	81				
22	16	58	3A	94	5E	130	82				
23	17	59	3B	95	5F	131	83				
24	18	60	3C	96	60	132	84				
25	19	61	3D	97	61	133	85				
26	1A	62	3E	98	62	134	86				
27	1B	63	3F	99	63	135	87				
28	1C	64	40	100	64	136	88				
29	1D	65	41	101	65	137	89				
30	1E	66	42	102	66	138	8A				
31	1F	67	43	103	67	139	8B				
32	20	68	44	104	68	140	8C				
33	21	69	45	105	69	141	8D				
34	22	70	46	106	6A	142	8E				
35	23	71	47	107	6B	143	8F				

1-2

530

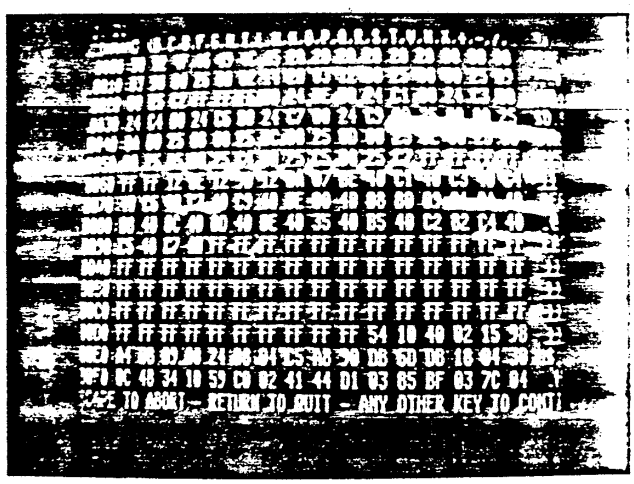
2-2
FINIS



Δ STATUS TABLES READ w/DEBUG ROM

SPARED SECTOR Cly HD Sec
TOTAL 03
LIST
00 25 08 95 00 08
00 25 09 95 00 09
00 25 02 98 00 02

BAD BLOCKS
TOTALS 12
LIST
00 24 0E 93 03 0E
00 24 01 94 00 01
00 24 03 94 00 03
00 24 04 94 00 04
00 24 06 94 00 06
00 24 07 94 00 07
00 24 09 94 00 09
00 25 00 95 00 00
00 25 0A 95 00 0A
00 25 0B 95 00 0B
00 25 0C 95 00 0C
00 25 0D 95 00 0D
00 25 0E 95 00 0E
00 25 35 95 03 05
00 25 D5 97 03 05
00 25 C4 98 00 04
00 25 C5 98 00 05
00 25 C7 98 00 07



Δ STATUS TABLES READ w/PROGRAM ROM

93 = 147
94 = 148
95 = 149
97 = 151
98 = 152

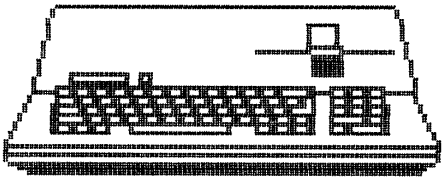
Δ SPARE SECTORS + BAD BLOCKS READ WITH -06 DEBUG SOFTWARE + PROGRAM ROM

531

26

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

DOCUMENT TITLE

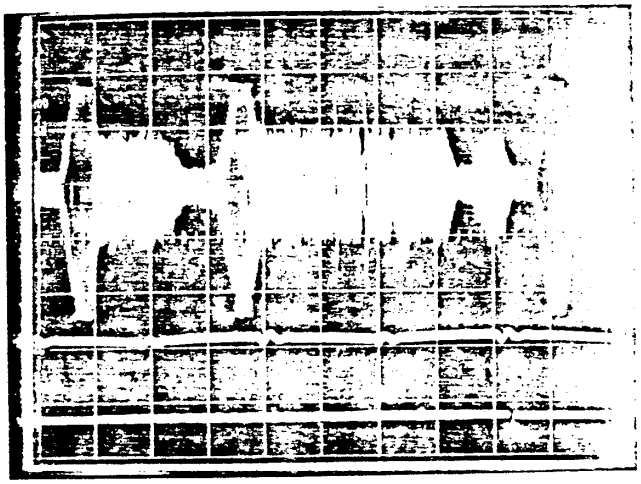
Misc. ProFile Oscilloscope Charts

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 5

WOULD NOT START SCAN
REPLACED Z8 & COMPLETED SCAN WITH TROUBLE
5 1/4" 7 8 1,3,4,6,7,9
151
152

HORZ = 2 MS/DIV



3|0|D|A|7|4|1|E|B|8|5|2|F|C|9|6|

INDEX

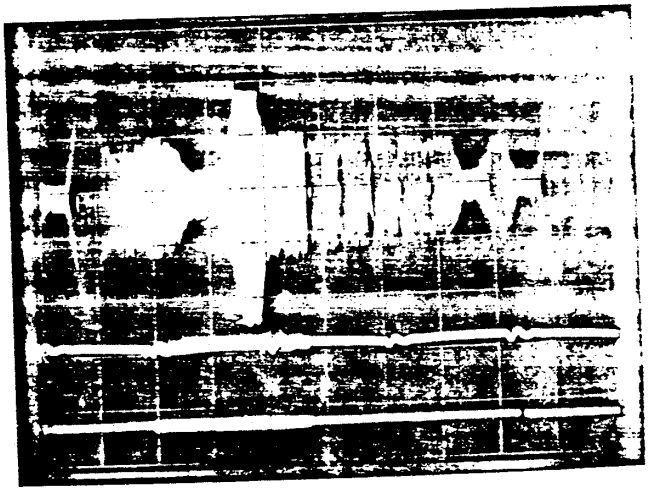
INDEX

CLY IK 148 HD
BAD BLOCKS (Secto
3, 7, 4, 1, 9, 6

ANALOG DATA
1V/DIV

VCO = .5 V/DIV

INDEX = TRIGGER VIEW

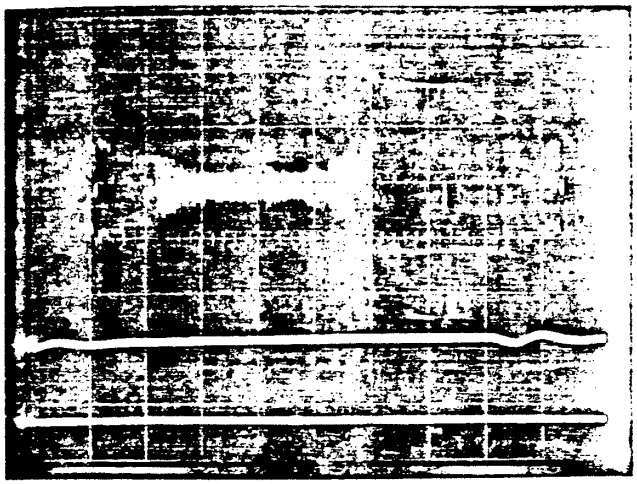


Highlighted AREA with B DELAY at .5 MS/DIV

Apple /// ProFile HD; Oscilloscope Charts - Misc.

1-5

533



HighLighted AREA @ .5ms/DIV

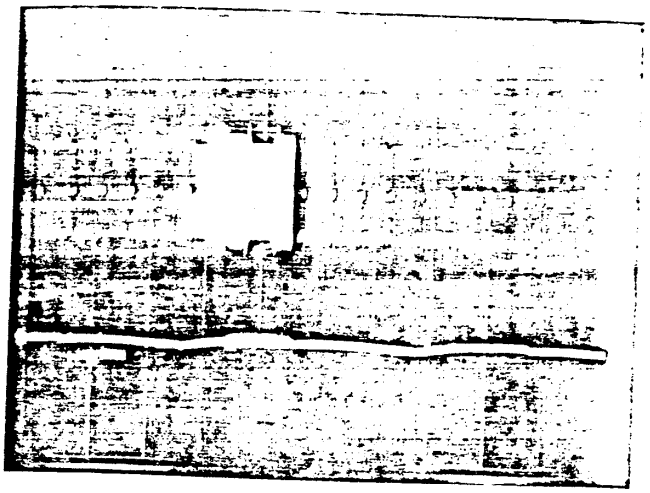
SAME AS ABOVE with INTERRUPTER MOVED SLightly TOWARD TK 147

INDICATES 28 PROBLEM WRITING Between TRACKS -

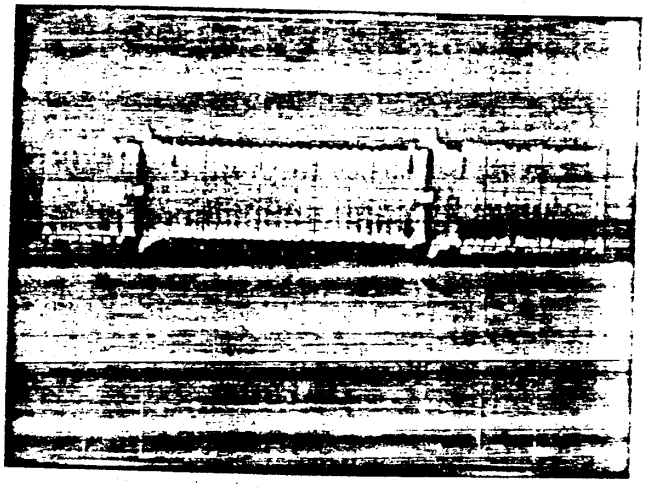
2-5

534

, end of 0's
No Different Pattern - 5



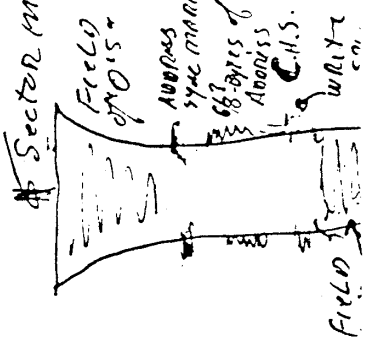
-A INTCH Pushed
-DLAY SWP @ .2ms/DI



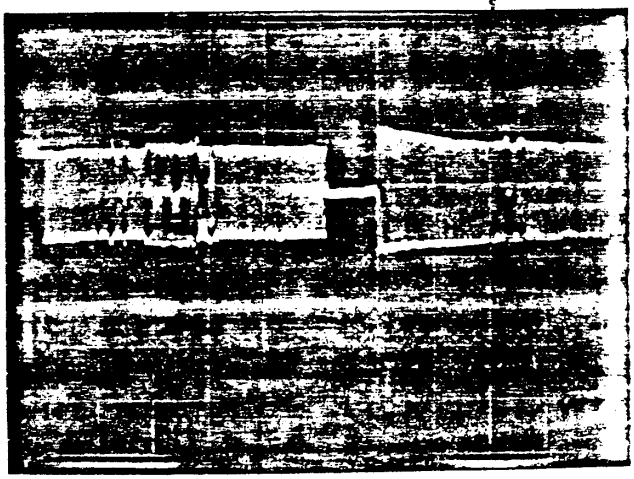
Highlighted Area Above

B DLYD Pushed

DLAY SWP set @ .2ms/DIV



MFM



DLAY SWP set @ 20 µs/DIV

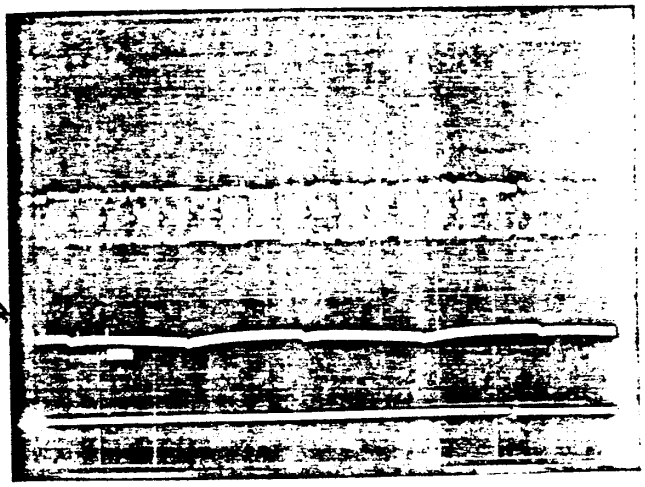
3-5

COULD NOT READ TRACKS CONSISTANTLY — FROM TK 70 TO 152 AT ALL

OBSERVATION - ANALOG DATA SIGNAL APROX. HALF of NORMAL
- OTHERWISE DATA LOOKED OK

535

READING OK ON TK 0

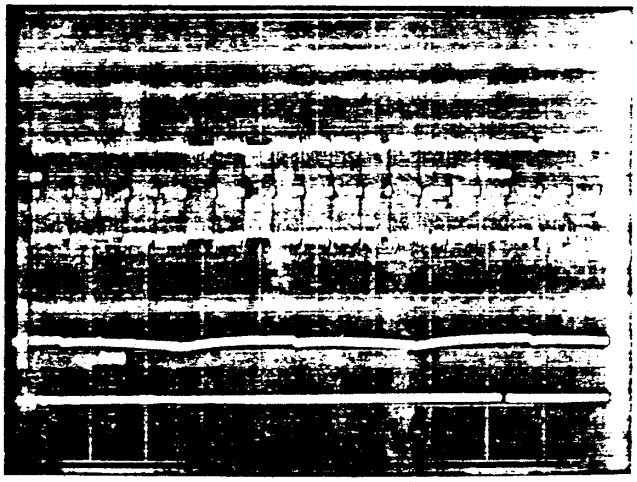


ANALOG DATA = 1 V/DIV

VCO = .5 V/DIV

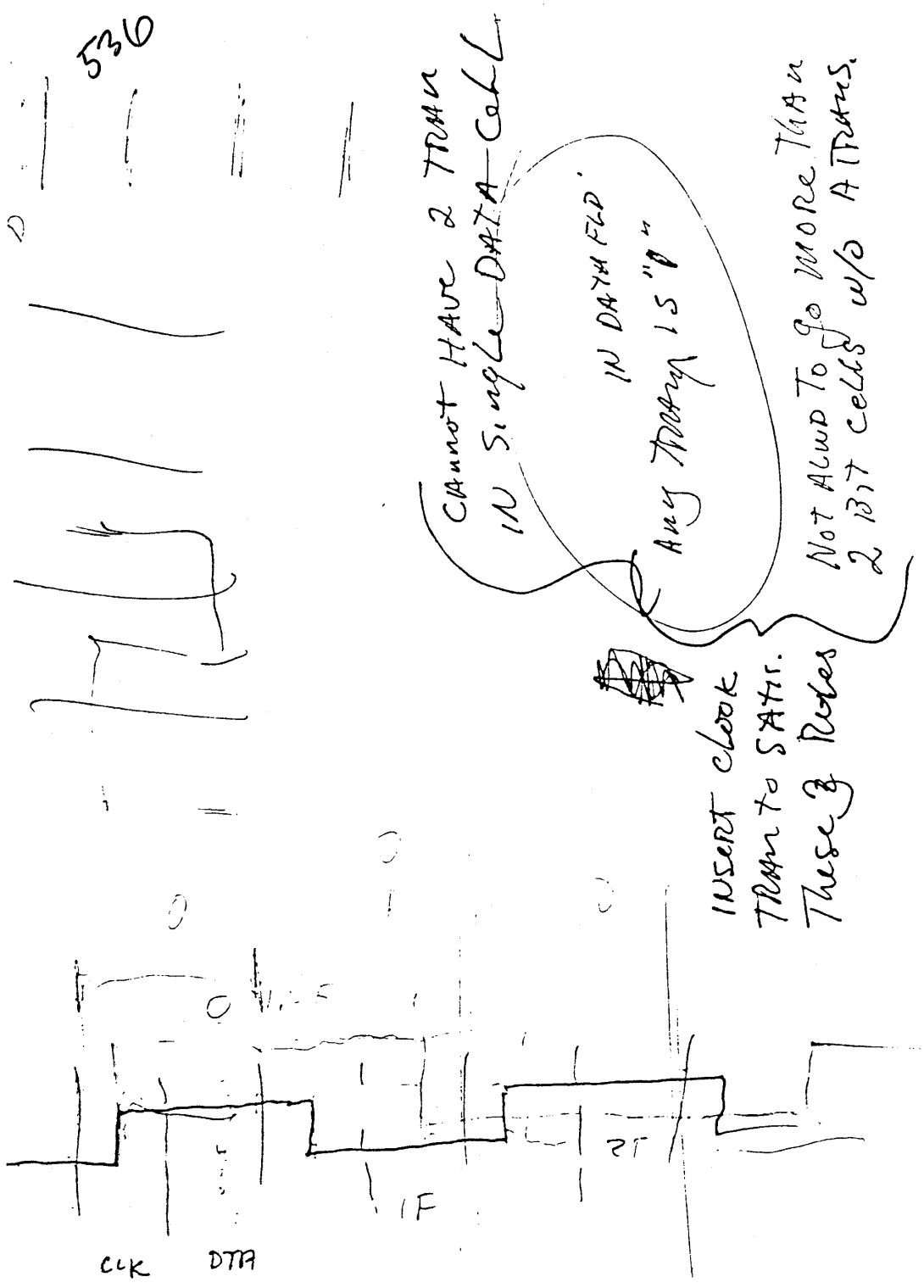
INDEX = TRIGGER

REPLACED ANALOG CARD READS ALL TKS WITHOUT ERROR



SAME AS ABOVE

4-5

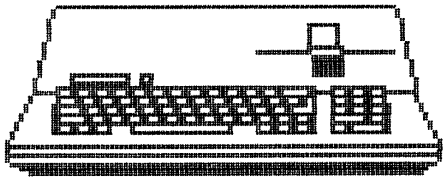


537

27

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

DOCUMENT TITLE

ProFile Controller ROM Misc. Facts

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Pages: 1

```
A/US FOR FD.97 CONTROLLER ROM   538
                                398
BIT
 7=NAK LAST COMMAND
   3 BYTE SEPORATOR BETWEEN WRITE BUFFER AND STATUS TABLES BAD OR 52-3 SET
 SPARE TABLE UPDATE OCCURRED,  BUFFER DATA IS CHANGED
 CAN NOT READ 3 SECTORS AFTER 2 RESEEKS
 3=CRC ERROR (READ ERROR)
 2=CAN NOT FIND TARGET HEADER IN 9 REVOLUTIONS)
 1=N.C.
 0=REQUESTED OPERATION FAILED

BIT
 7=CAN NOT READ 3 SECTORS AFTER SEEK
 6=MORE THAT 32 SPARES (SPARE TABLE OVERFLOW)
 5=N.C.
 4=MORE THAN 100 BAD BLOCKS (BAD BLOCK TABLE OVERFLOW)
 3=CAN NOT READ STATUS SECTOR
 2=SPARING OCCURRED
 1=SEEK TO WRONG TRACK
 0=N.C.

BIT
 7=PROFILE WAS RESET
 6=REQUESTED BLOCK OUT OF RANGE
5-0=N.C.

 READ FAILURE AFTER READ COMMAND
TUS FOR FD3.97 FORMATTER ROM

 =NAK LAST COMMAND
6=NO INDEX FOUND DURING FORMATTING
5=NO SECTOR MARK FOUND DURING FORMATTING
4=SEEK ERROR (UNABLE TO READ 3 CONSECUTIVE HEADERS ON A TRACK)
3=CRC ERROR (ONLY FOR READ WRITE, OR WRITE/VERIFY)
2=TIMEOUT ERROR (UNABLE TO FIND HEADER IN 9 REVS.)
1=N.C.
0=COMPARE ERROR ON A WRITE/COMPARE)

BIT
7=PROFILE HAS BEEN RESET
6=FOUND INVALID TRACK NUMBER WHILE READING OR WRITING
5-2=N.C.
1=SEEK TO WRONG TRACK OCCURRED
0=N.C.


BIT        7     6     5

E TIMEOUT  0     0     0
  TIMEOUT  1     0     0
  CRC      1     1     0
  COMPARE  1     1     1

  UMBER OF ERRORS ENCOUNTERED WHEN FORMATTING DATA FIELDS
  SCANNING
```

Apple /// ProFile HD; Controller ROM Misc. Facts

1-1

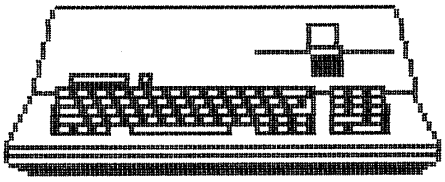
FINIS

539

28

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Misc. ProFile Facts

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

540

Apple /// ProFile HD: Misc. HD Facts

Drive Runs at 3600 RPM.
2 Platers
4 Heads
16 sectors Per Track
305 (131 Hex) Total tracks

Spares table at Cyl 99 Hds 0-3
Sectors 0-F

Hds 0,1 Location were spared data is written
Hds 2,3 Physical Location of spared sector

To Pause on Error - Pause Flag mu
Be inverse (second Line from the top)
Just type "P" (Return). to cancel type ag

Debug Software (10megbyte Debugger) will only work with the Debug Prom, Doesn't work with the System Prom.

Pages: 4

1-4

541

B = Fill Output Buffer
E = Erase Track
H = Hard Reset Drive
K = Write a Header
O = Turn off Stepper
R = Read Block
U = Park arm
X = Do Commands again
? = HELP
C = Compare Buffers
F = Formate
I = Init Spares Tables
L = Set Range Limits
P = Toggle Pause/Print
S = Set Slot Number
V = Scan all Blocks
D = Display Buffers
G = Go Seek
J = Read a Header
M = Write Sector Marks
Q = Quit
T = Test Controller Ram
W = Write Block
+,- = Increment, Decrement Block

DEBUGGER SOFTWARE COMMANDS

2-4

542

Profile Media Description Info

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
	P	R	O	f	I	L	E	_	1	O	M	.	.	.	.	.
0000	50	52	4F	46	49	4C	45	20	31	30	4D	01	00	00	00	10
0010	03	99	00	4C	00	02	14	20	XX	XX	FF	FF	FF	FF	FF	FF

Bytes (0010 row)	Meaning
03 99	Rom Version in Profile Cont. Board is updated Each Time Spare table upd
00	SPACE
4C 00	
02 14	Number of Bytes Per Sector
20	Number of overall spared Sector Available
XX	Number of Actual spares
XX	Number of Actual Bad Blocks
FF FF FF FF FF FF	Depends on Rom Version

34

543

To Read a track RTØ HØ SØ
RTØ
RHØ
RSØ
RØ = TØ HØ SØ

Read Blk Inc. Sect. Repeat.

RØ Return
R, IS1, X
Read, Increment Sect by 1, Execute

J- Read a Header doesn't seem to Work

Range Limits

LT - LH - LS - LR (Limits Reset)
LT Low - T High LT 5 - T 10
LH Low - H High LH 1 - H 2
LS Low - S High LS 5 - S E
L (Block Low) - (Block High)

Must use comas instead of Spaces Between Commands

4-4

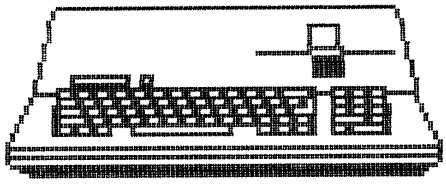
FINIS

29

544

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

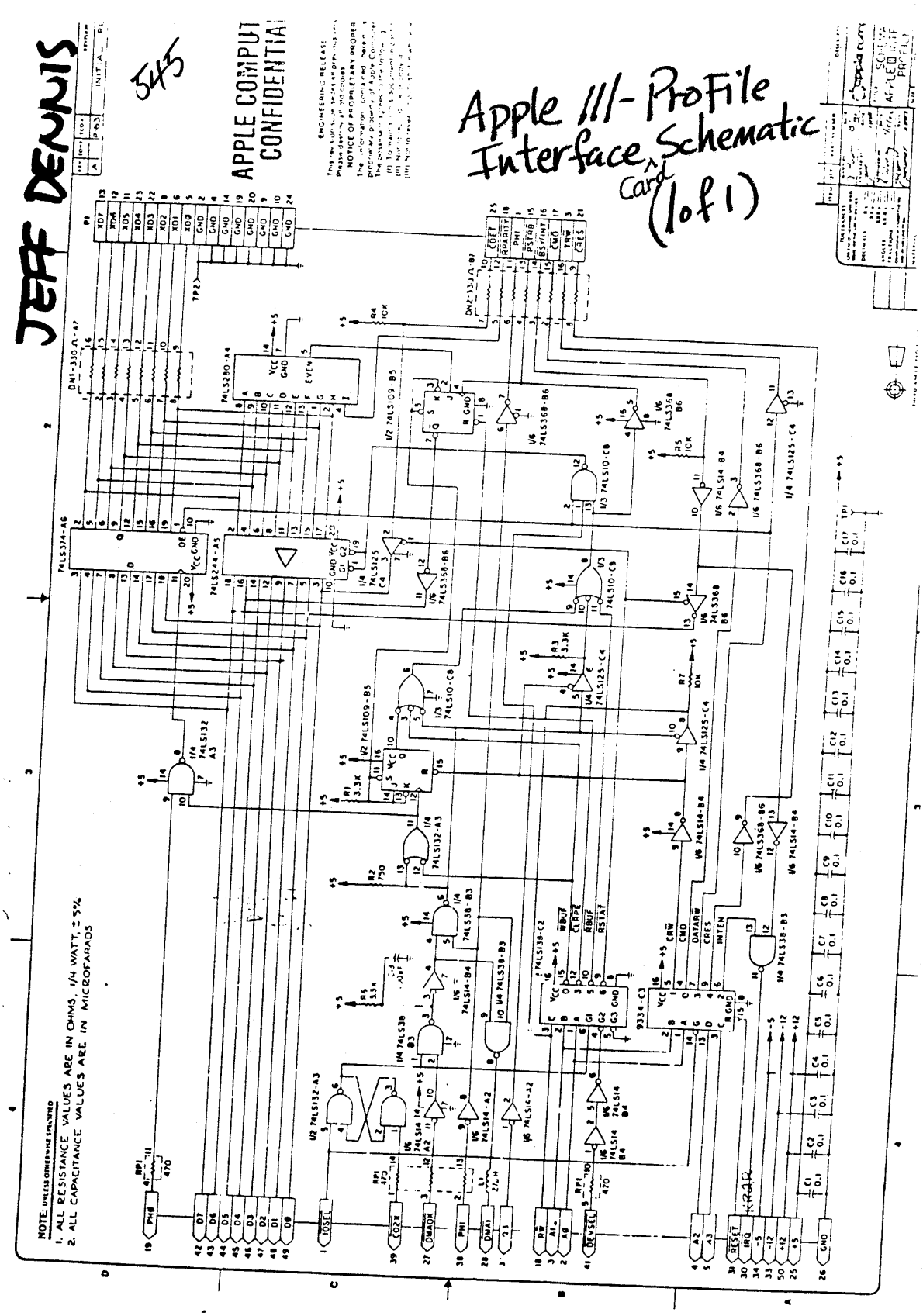
COMPONENT NAME

ProFile Hard Disk

DOCUMENT TITLE

Apple III-ProFile Interface Card Schematic

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

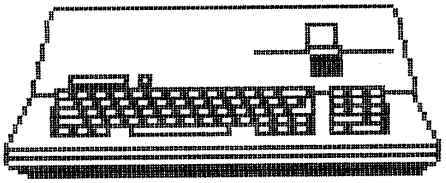


5446

30

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

DOCUMENT TITLE

ProFile Format and Diagnostic Program
Revision 0.11 (Preliminary)

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

Apple /// ProFile HD: ProFile Format and Diagnostic Program Revision 0.11 (Prelim)

SHF

FORMAT - Profile Format and Diagnostic Program Revision 0.11 (Preliminary)

FORMAT allows the Apple III (or any other host computer) to format Profile, to do a scan of all sectors, to read or write any sector (including those sectors not normally accessable because they're reserved for spares and spare tables or are sectors that have already been spared), to initialize Profile's spare tables, to read or write any of Profile's 1024 bytes of RAM, to read or write the internal registers of Profile's Z8 microprocessor, to read any header (including the data field), to loop on reading any header, to loop on formatting sector marks, to loop on formatting headers, to loop on formatting data fields, to loop on formatting sector marks and headers, and to loop on formatting sector marks, headers, and data fields.

The command bytes for each of the commands is shown below.

READ	00	track	head	sector
WRITE	01	track	head	sector
WRITE COMPARE	02	track	head	sector
FORMAT	03			
SCAN	04			
INITIALIZE SPARE TABLES	05			

A read of track FF gets the status table header, which includes the name of the device and the program revision number, preceded by the letter 'D'.

Pages: 6

1-6

48

READ REGISTERS

06	1st register

Even though the Z8 control registers have addresses from F0 through FF, the program expects values from 80 through 8F. They immediately follow the general purpose registers when read by the host.

WRITE A REGISTER

07	register	value

For the write a register command the program expects the control registers to be referenced by their real addresses F0 through FF. Any register can be written to (at the user's own risk). The Ports (registers 0 through 3), the working registers (40 hex through 4F hex), the control registers, and a few others will probably be changed before the user gets a chance to read them. Restoring the registers to their original values is highly recommended.

READ RAM

	beginning address	
08	MSB	LSB

WRITE RAM

	beginning address	
09	MSB	LSB

LOOP ON FORMAT SECTOR MARKS

0A	track	head

LOOP ON FORMAT HEADERS

0B	track	head

LOOP ON FORMAT DATA FIELDS

0C	track	head

2-6

549

READ HEADER (AND DATA)	0B	track	head	sector	control

The sector parameter is actually the physical sector. The first one after index (logical sector 0) is 1A (decimal) and the ones following range from 1 to 15.
The control byte is decremented once, then rotated left after each read, with D7 getting shifted into D0. Profile will loop on reading headers until a zero is shifted out of D7.

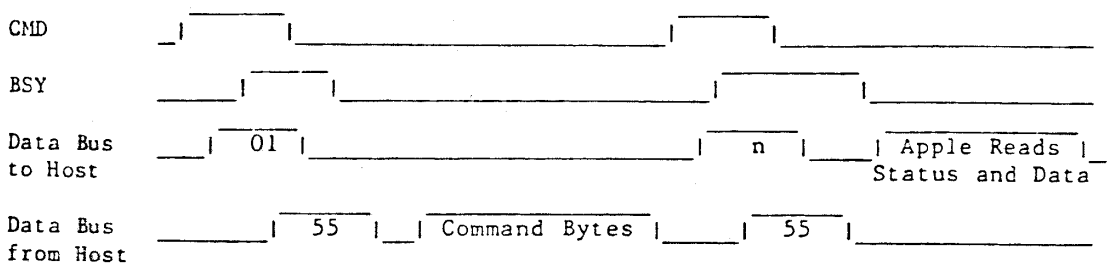
LOOP ON FORMAT SECTOR MARKS AND HEADERS	0E	track	head

LOOP ON FORMAT SECTOR MARKS, HEADERS, AND DATA FIELDS	0F	track	head

TURN OFF STEPPER	10

The read sector, format, scan, initialize status sectors, read registers, and turn off stepper commands use the read protocol shown below.

Read Protocol



In the timing diagram above, n (the Z8's response to the second command high) is always equal to the command byte plus 2. And while the turn off stepper command uses the protocol shown above, the data available after the 4 status bytes is undefined. The data read by the read sector and read registers commands is obviously the sector and register data, respectively. The data read by the format and scan

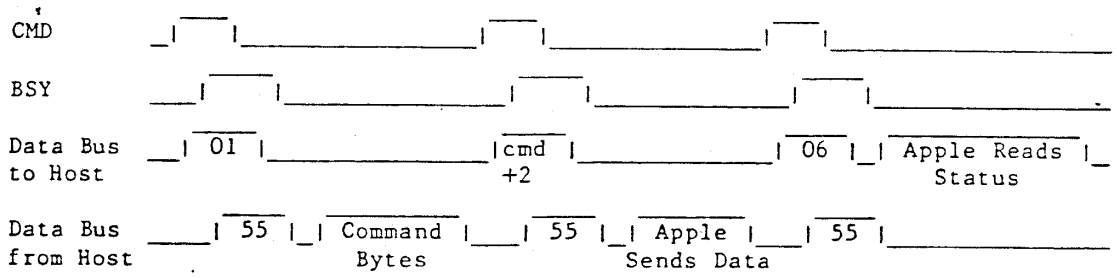
3-6

440

commands is a list of the track, head, and sector where an error occurred, along with the first status byte of the error itself. Only the first 32 errors are listed. The first 255 errors are counted in status4. The scan and format operations will abort if there are more than 255 errors. The list of errors if followed by a delimiter of FF FF FF FF. The data read by the initialize spare table command is a list of the all spare table sectors that Profile was unable to do a write compare on (a write, followed by a read, followed by a data compare), along with some error status information. The organization of the list will be described later in this document. The read RAM command uses the read protocol shown above, except that no status bytes are made available to the host. If the control byte in the read header command is not zero, the read protocol is used but only the first of the four status bytes precedes the data, instead of all four. The write register command uses the read protocol, execept that no status or data bytes are made available to the host.

The write and write compare commands use the protocol shown below

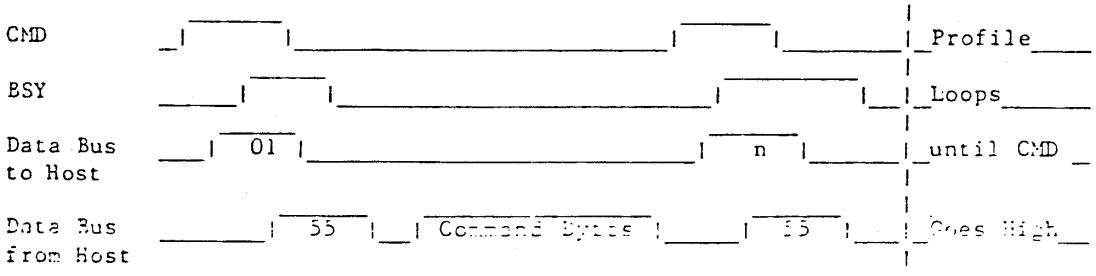
Write Protocol



The write RAM command uses the write protocol shown above, except that no status bytes are made available to the host.

All other commands use the looping protocol shown below.

Looping Protocol



4-6

5-5-1

Following is a description of the bits of the four status bytes.

STATUS 1

7 = 1 if Profile received <> 55 to its last response
6 = 1 if no index found during formatting
5 = 1 if no sector mark found during formatting
4 = 1 if SEEK ERROR - unable to read 3 consecutive headers on a track (one try only)
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 compare error on a write compare

STATUS 2

7 = 1 if Profile has been reset
6 = 1 if track number invalid while reading or writing a sector
5 = N.C.
4 = N.C.
3 = N.C.
2 = N.C.
1 = 1 if seek to wrong track occurred
0 = N.C.

STATUS 3

D7, D6, and D5, along with D4 and D3 from STATUS 1, tell why a write compare operation failed.

	D7	D6	D5
write timeout	0	0	0
read timeout	1	0	0
read CRC	1	1	0
data compare	1	1	1

STATUS 4

7 - 0 = the number of errors encountered when formatting data fields or scanning the disk.

No attempt will be made to initialize the spare tables if Profile is unable to read 3 consecutive sectors when seeking to that track and

5-6

GGN

head. The data after the status bytes following an initialize spare tables commands consists of a list of all the sector that couldn't be written to (one byte each) OR'd with the STATUS 3 result when the error occurred. The D4 bit differentiates between head 2 (0) and head 3 (1).

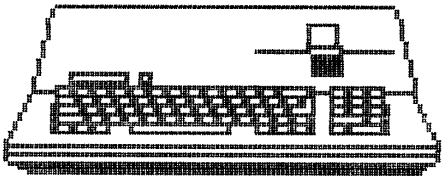
6-6 FINIS

554

31

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

DOCUMENT TITLE

Software Device Driver Info
(for /// SOS)

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

554

Pages: 5

I/O control-block definitions for Peripheral test systems

In general the Z-flag C-flag and accumulator will have most of the useful return status information.

Z=1 then all ok
Z=0 then if C=0 then non-fatal error (we got job done)
Z=0 then if C=1 then Fatal error (we didn't get it done)

Acc will always have the first byte of status information (the same information which is returned in the status buffer)

The length of data given/returned in the data/status buffers is potentially different for different devices and it is up to the programmer to provide enough space for the data for any particular driver.

So far the drivers we have designed have the following specs

Profile	data	532 bytes.
	status	8 bytes.
Duofile	data	536 bytes
	status	9 bytes
Clock	data	18 bytes max (format dependent)(ends with NULL
	status	1 byte
Barcode	data	25 bytes max (format dependent)
	status	1 byte

Apple /// ProFile HD:
Software Driver Info
for 6502 systems
(i.e. /// SOS)

1-5

Profile:

command	Single byte command to instruct the profile on what to do
Buf-lo	Pointer to the location in memory to get/give the requested data
Buf-hi	Hi byte of pointer
Stat-lo	Pointer to the location in memory to give status information
Stat-hi	Hi byte of pointer
Slot	Slot number of the device (0= no slot)
Device	For drivers that will talk to more than 1 device per slot (0=1)
Blk-lo	For block orented devices this is the lo byte of the block number
Blk-med	The medium byte (it is a 3-byte field)
Blk-hi	The hi byte
Spf=lo	Pointer to the location in memory to give/get special functions
Spf-hi	For devices that have special functions or controls (0000=none)

2-5

Duofile: 6540

command	Single byte command to instruct the profile on what to do
Buf-lo	Pointer to the location in memory to get/give the requested data
Buf-hi	Hi byte of pointer
Stat-lo	Pointer to the location in memory to give status information
Stat-hi	Hi byte of pointer
Slot	Slot number of the device (0= no slot)
Device	For drivers that will talk to more than 1 device per slot (0=1)
Blk-lo	For block orented devices this is the lo byte of the block number
Blk-med	The medium byte (it is a 3-byte field)
Blk-hi	The hi byte
Spf=lo	Pointer to the location in memory to give/get special functions
Spf-hi	For devices that have special functions or controls (0000=none)

Mode	Duofile mode control
Addr hi	Hi address of execute local pgm command
Addr lo	Lo address
Dld Lo	Lo address of location in APPLE ramspace to start downl
Dld Hi	Hi address of start location
Dldl lo	Lo and hi length of dataa going into the ramspace
Dldl Hi	Hi byte of length

3-5

SGX

Command Summary

00 - Null command. Do nothing.

01 - Status command. Return the extended status information from the controller.

02 - Execute Run program placed in ram at location specified in the special function table for the device

03 - Modify Modify the registers in the controllers CPU according to device dependent requirements (messy sigh)

04 - Unload Unload the data transfer device from the media and prepare to release the media, (this includes turning off spindle moto or shutting down laser ect. as required)

05 - Write Write data onto (into) the media at location specified (Do an automatic seek as neccessary). For block devices the length of the data will usually be 512 or 536 bytes per block

06 - Format Write header (address/data) information onto the media in pre for use as a block device

07 - Seek Position the transfer device onto (into?) the correct area of the media for the requested block.

08 - Read Read data from the media at location specified (Do an automatic seek as neccessary). For block devices the length of the data will usually be 512 or 536 bytes per block

09 - Load Prepare to engage the media, (this includes turning on spindl motor or powering up the laser ect. as required). Then Load the data transfer device onto (into) the media (if neccessary

0A-17 Reserved for future expansion.

These are specalized commands available to a few controllers

18 - Stproff Turn off power to the device which moves the transfer device.

19 - Spmoff Turn off power to the device which moves the media.

1A - 1F Reserved for future expansion

4-5

496

These are commands for just the I/O driver

20 - Reset Reset the device by hardware switch if possable else command the device to do a software reset.

21 - Init Initalize the internal defaults of the driver (including the special functions)

28 - Version Return the Version number and Driver I.D. in the form

Enn.nn-ddmmyy-xxxxxx<NULL>

Where E is E for experimental or R for released

nn.nn is the Version number (E.G. 02.31)

dd is the day
mm is the month
yy is the year of the revision/or minor update

xxxxxx is the 6 digit Identifier

NULL is a null ($0) character (its a handy terminator)

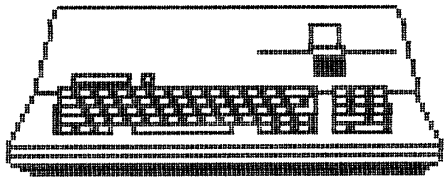
FINIS 5-5

559

32

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///

Apple ///+

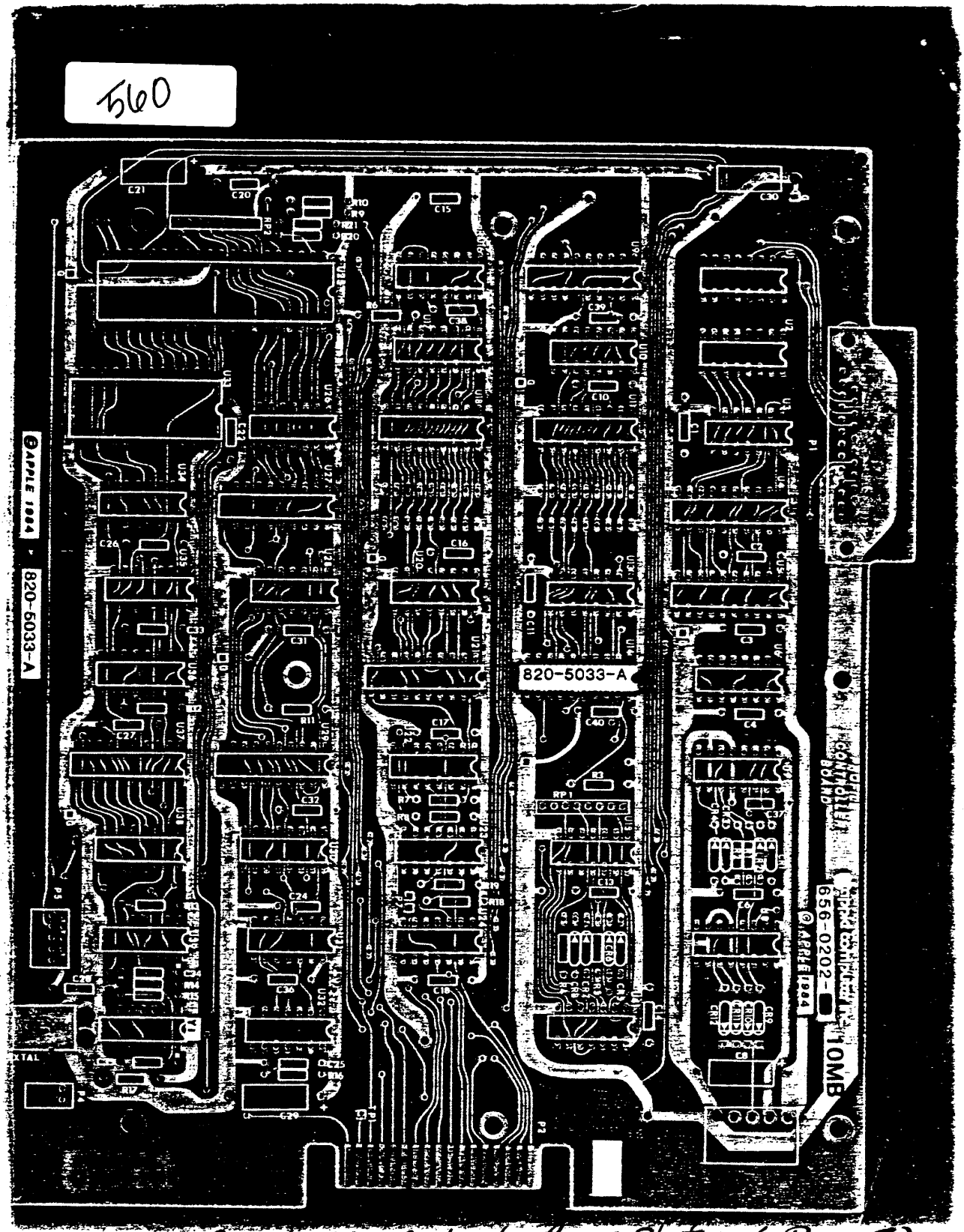
COMPONENT NAME

ProFile Hard Drive

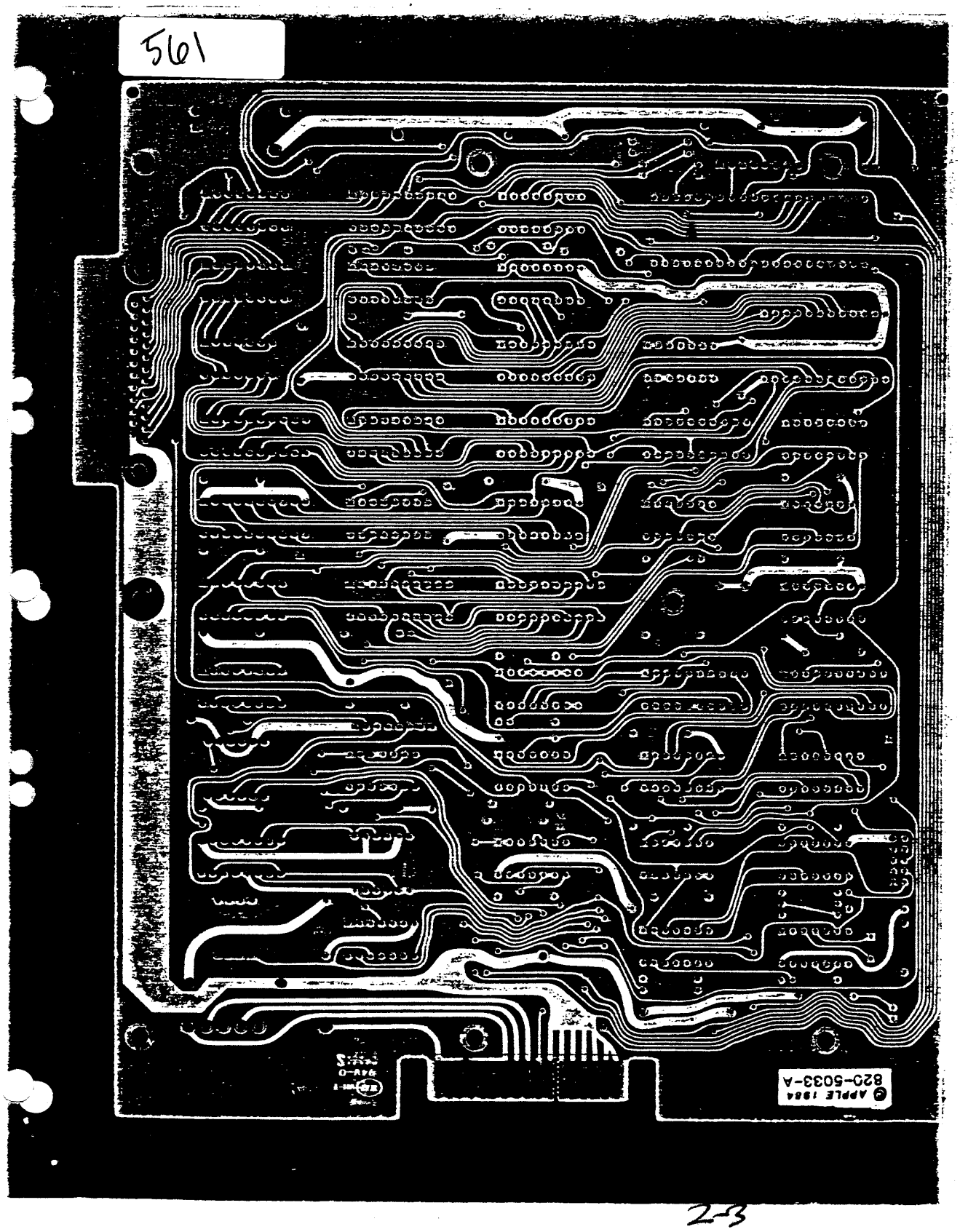
DOCUMENT TITLE

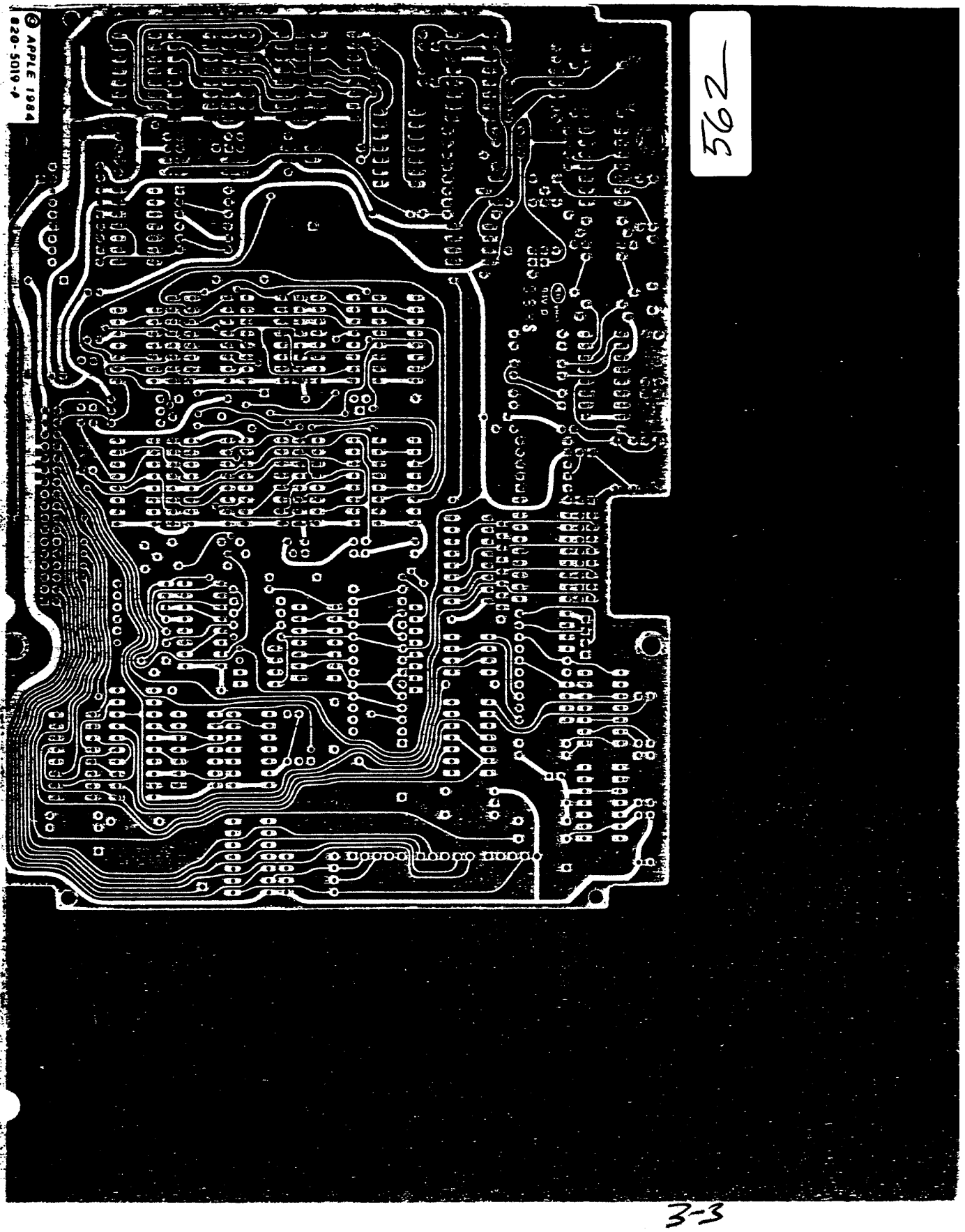
ProFile Controller Board Photos

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]



Apple /// ProFile HD: Controller Board Photos (Page 3)
1-3



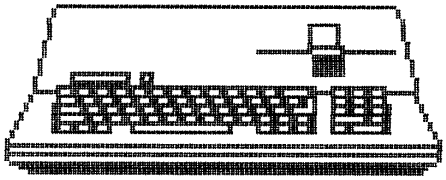


563

33

Apple Computer, Inc.
Service Engineering Department

Apple /// Computer Repair Document



Apple ///
Apple ///+

COMPONENT NAME

ProFile Hard Drive

DOCUMENT TITLE

Functional Description of the ProFile Analog Board

DAVID T. CRAIG
736 EDGEWATER, WICHITA, KANSAS 67230 [USA]

564 Apple /// ProFile Hard Drive

FUNCTIONAL DESCRIPTION OF THE PROFILE ANALOG BOARD

(Refer to the six page schematic for reference details)

(rough draft)

The following document has been written with the purpose of describing the circuitry on the Profile Analog Board in detail. It is not meant to be a troubleshooting guide, but is aimed at informing the technician of the different circuits and operating parameters of the board. This document should be used with the Profile Analog Board Tester, Apple Part # 890-0184, in testing and repairing any problem on the board.

In the event that it is desired to troubleshoot the board onboard the Profile unit itself, the board may be accessed using the following method:

Turn the Profile off and let it sit for a minute until the drive stops rotating. Access the Profile Analog board by turning the Profile over on its top, providing the top cover is on the unit. Remove the metal cover under the cabinet which accesses the Analog board. Power the drive back up and refer to the schematic for the test points and waveforms. No harm can come to the drive while it is running in this position, but random data errors will probably occur. Make sure that the drive cannot fall over or receive any sharp jolt or shock as this will damage the HDA media.

223 direct.

0 & 1 - Spare or 0 & 1

Pages: 16

1-16

565

PAGE ONE

The connector J2 on the left connects the Analog board to the Controller board. The signals coming into the board are the write data (NRZWDTA), the 10mhz write clock (2F), the write precompensation and current control signal (PCOMP-LOCUR), and the write gate signal (WTGT). Outputs to the Controller board are the TRK 0 detection and Index pulse signals coming from the HDA assembly. Power and ground inputs to the board are also shown for reference.

The track 0 signal indicates when the head stepper motor assembly is at track 0 or the outer track location, while the index pulse occurs once per drive revolution. The index pulse can be monitored at test point TP-9 and used as an oscilloscope trigger for viewing the track analog data at test point TP-1.

The +/- 0A and +/- 0B stepper motor control phases pass through the board and are routed to the HDA stepper motor at connector J7.

The Track 0 signal is obtained from the photosensor mounted on the HDA stepper motor. The sensor is tripped by the rotating arm assembly on the motor shaft when the head array is at the track 0 position, and the sensor output comes in the board at connector J6. The signal is detected and amplified by device UC30 and passed to the Controller board at J2 pin 2.

The Index signal comes from the photosensor mounted under the HDA. The sensor is tripped by a small metal tab fastened to the bottom plastic cover of the drive spin motor and comes in the board at connector J8. The signal is detected and amplified by device UC30 and passed to the Controller board at J2 pin 4.

The write encoding and precompensation circuitry consists of devices UC27, 28, 29, and 30. The 10mhz write clock is divided by UC2 to create a 5mhz clock (1F) for VCO reset control purposes. The data and control signals are decoded by the PAL logic array UC28, and DLY 2 provides delay references of 10 and 20nsec for data signal precompensation.

2-16

566

This is done to compensate for the magnetic fluxes on the disk media trying to repel each other because of their polarity differences. If this happens, the flux reversals on the media will all have more or less the same distance between them, resulting in a sine wave pattern on the disk with no 1F or 2F signal component margin between flux reversals.

The signal outputs of the PAL indicate Early, On Time, or Late data occurances as detected in the PAL data register. These are generated by comparing the data bit pattern and defining the three potential transition points for every flux reversal with the write timing. These signals then shift the write data signal by selecting the proper amount of delay (usually 20 nsec) to over-write any potential bit shift on the media surface. This precompensates the data on the disk surface, allowing a higher bit density without loss of resolution.

3-16

56X

On this page are the write amplifier, write current control, head select, and read detection circuitry. The write data (WXS) is gated through UC2 into the write current driver array UC1 (MPQ6700) and UC31 (MPQ2907). The data through UC2 will be blocked if the WRTSM' write sector mark signal is true, as this is used to write blank areas on the track for disk sector formatting identification.

The current level of the final driver transistor array is set by a voltage reference developed across zener CR1. This diode has a value of 1.2 to 1.25 VDC, which sets the drive level of UC1 to 24ma of write current (48ma peak-to-peak).

This level is used to write tracks 0 through 128, while tracks 129 to 152 use less current because of the thinner media coating and smaller disk radial area per bit on the inner tracks. Remember that the drive spin is a constant speed, and the head covers more area on an outer track than it does an inner track during the same time period. In addition, the media surface coating on the disk is thinner towards the center. The smaller available area requires less current to effectively write the media than the larger.

To set the lower current value, the PCOMP-LOCUR' (Precompensation- Low Current) signal goes active, causing UC3 to "steal" 5ma from the write current value through the 1.78Kohm resistor R10. This results in a lower head write current level of 19ma (38ma peak-to-peak) for the inner tracks.

The write current passes through diodes CR4 and CR13 to four connector pins ABCD, and from there directly to the heads. If desired, a pin to pin wire loop can be used at pins AB or CD to monitor the actual write current with a current probe, as the pins are shunted with 22 ohm resistors to the head matrix.

4-16

568

Signals HS-1 and HS-0 are decoded by 1 of 4 decoder UC4. The output selects one of the four heads for either reading or writing by correctly biasing the center tap of the head coil. The other heads are held in an off condition allowing only one head to be selected at a time.

The POWEROK signal is a monitor function of the power supply and instantly terminates any write signal if the main power to the Profile unit is lost. The POWEROK line is dropped low by the power supply if the incoming AC is lost, causing the voltage drop across R16 to turn on UC13. This cuts off UC1 while there is still power out of the supply maintaining the system electronics. The POWEROK signal also holds the stepper motor, Z8 controller, and write amplifiers off for a brief moment after initial power on. This delay action can be seen by the READY lamp on the front panel lighting for 1 to 1 & 1/2 seconds when Profile power is first turned on.

The WRTSM' signal is used to "write" (actually erase) the track sector marks. The Profile uses 16 blank sector marks per track, radially aligned on the disk, and this signal inhibits any write action to the head during the formatting of the these areas on the disk.

The sector marks are erased as 20usec long blank spaces and the disk read logic needs a minimum of 10usec to identify the mark. If any sector mark on a track is written over, the data in that sector is effectively destroyed and cannot be used.

For read operation the forward biased diode matrix detects the differential analog transitions (-X and -Y) of the selected head. The other heads are biased so that there is no current flow in the coil, preventing any signal detection. These analog signal transitions are extremely small and attempting to view them in this diode network is impossible.

5-16

569

PAGE THREE

The -X and -Y analog signals enter the ECL 10114 preamplifiers, which are power supply isolated by RG1 for noise immunity. The 10114 was selected because of its excellent input geometry with very little noise. The signals then pass through a low pass filter which is set for 7.5Mhz, or 3 times the maximum read signal frequency of 2.5Mhz (2F).

This filter attenuates the major error element of the detected signal, which is the third harmonic of the flux reversal. If viewed on a scope, this is the "knee" part of the waveform transition midway between signal peaks. The signal then enters a 592 video amplifier UC6, which acts as the AGC (Automatic Gain Control) integrator and controller.

The AGC level is developed by device Q8, a VCR2N junction whose bias level is controlled by UC8, a 353 AGC detector and amplifier. The AGC controlled output of the 592 passes through an emitter follower and buffer network UC7 and is available for monitoring purposes at test points TP1 and TP2 at the bottom of the page. These are differential signals and can be shown individually or compared with each other.

The signal levels at these test points are still analog in form and should be around 1.5 to 2 volts peak-to-peak. Anything outside of these levels is indicative of a bad AGC network and renders the data unreadable to the system. If the AGC is lost, the Profile cannot perform the initial scan function after power up.

If this should happen, suspect anything in this network. However, remember that the signals into the circuit are far too small to be monitored directly. The best way to troubleshoot anything in this area is to swap out the ICs in the circuit. Troubleshooting to a bad component other than an IC is almost impossible and the board may have to be scrapped out if any other component is at fault.

To the right of UC8 pin 7 is a jumper used to set the AGC at a fixed level referenced to -12VDC. This is done only in board manufacture test and the jumper should never be found in the field. Remove it if installed.

6-16

570

The AGC controlled analog signals enter two 10116 ECL schmitt level circuits (UC9) which make up the zero crossing detector. If any signal dropout occurs because of poor AGC, it will be detected by this circuit and interpreted as data.

The upper portion of the circuit sets the gain for a minimum amplitude symmetrical signal level. The lower circuit has a delay line in parallel which acts as a 1/4 wavelength delay line through the sum and cancel effect on the signal. In addition, the delay line acts as a good low pass filter to eliminate "hash". The output of the OR gate UC10 is the detected signal pulse. This is allowed to happen only during the "signal allowed" period set by the delay circuit.

The detected signal is then gated through UC11 which outputs a low going pulse for every signal that is detected by the zero crossing detector circuit. The output of UC11 is developed by an RC network consisting of C20, R65, and R66, which forms a 50nsec pulse for every flip-flop transition of UC11. If the pulse width is less than 40nsec, capacitor C20 should be replaced to increase the time factor and corresponding pulse width.

As previously outlined, Profile uses blank areas on the track for sector identification. These marks indicate the 16 individual track sectors and are placed on the track when it is initially formatted. If any sector mark is over-written or lost, that one sector cannot be read by the system and the data within it is lost.

Another schmitt trigger device (UC13) coupled with a slow response time RC circuit is used to detect these blank marks and will not detect any "no signal" period less than 10usec long. The input develops the signal across CR18, CR19, and CR20 with the RC circuit of C23 and R74 delaying the response of one side of UC13.

This means that sec r cannot be detected if any noise or off track data has over-written the blank disk area to the point where it is less than 10usec long. The sectors are written in 20usec periods during formatting, so some degradation can occur before the sector is completely lost.

The AM HOLD or address m k hold is a function of the Write Sector Mark and Write Gate signals on page 2. It is used to clamp the circuit during write operations, which prevents the output from indexing any data to the controller card.

7-16

57

PAGE FIVE

On this page are the VCO (Voltage Controlled Oscillator) phase comparator control and data output circuits.

The read gate RDGT provides the control element for the circuit. The VCO is reset during any "no read" condition by the 1F 5Mhz signal. This allows the VCO to be triggered almost instantaneously by the RDGT signal instead of recovering from an extreme compliance condition. The circuit at the top of the page sets the time constant of the reset signal (usually a few microseconds).

When the read gate goes active it triggers the VCO. The circuit then acts to lock the leading edge of the VCO pulse to the leading edge of the 50nsec signal data pulse. This is done by phase comparing the data pulse with the VCO pulse and generating a corresponding +/- INC or +/- DEC control signal. This pulse goes to the charge pump on page 6 which controls the VCO operating frequency. It takes about 20 read data pulses to properly control the VCO as the pulse train consists of data pulses at different intervals due to MFM criteria.

The net effect of the VCO control is to compensate for any difference between the read circuit timing and the speed of the disk. Any speed difference creates an error in the data decoding as the data has to occur at 200nsec cycle periods. The VCO compensates for this, allowing a motor speed deviation of up to 3%.

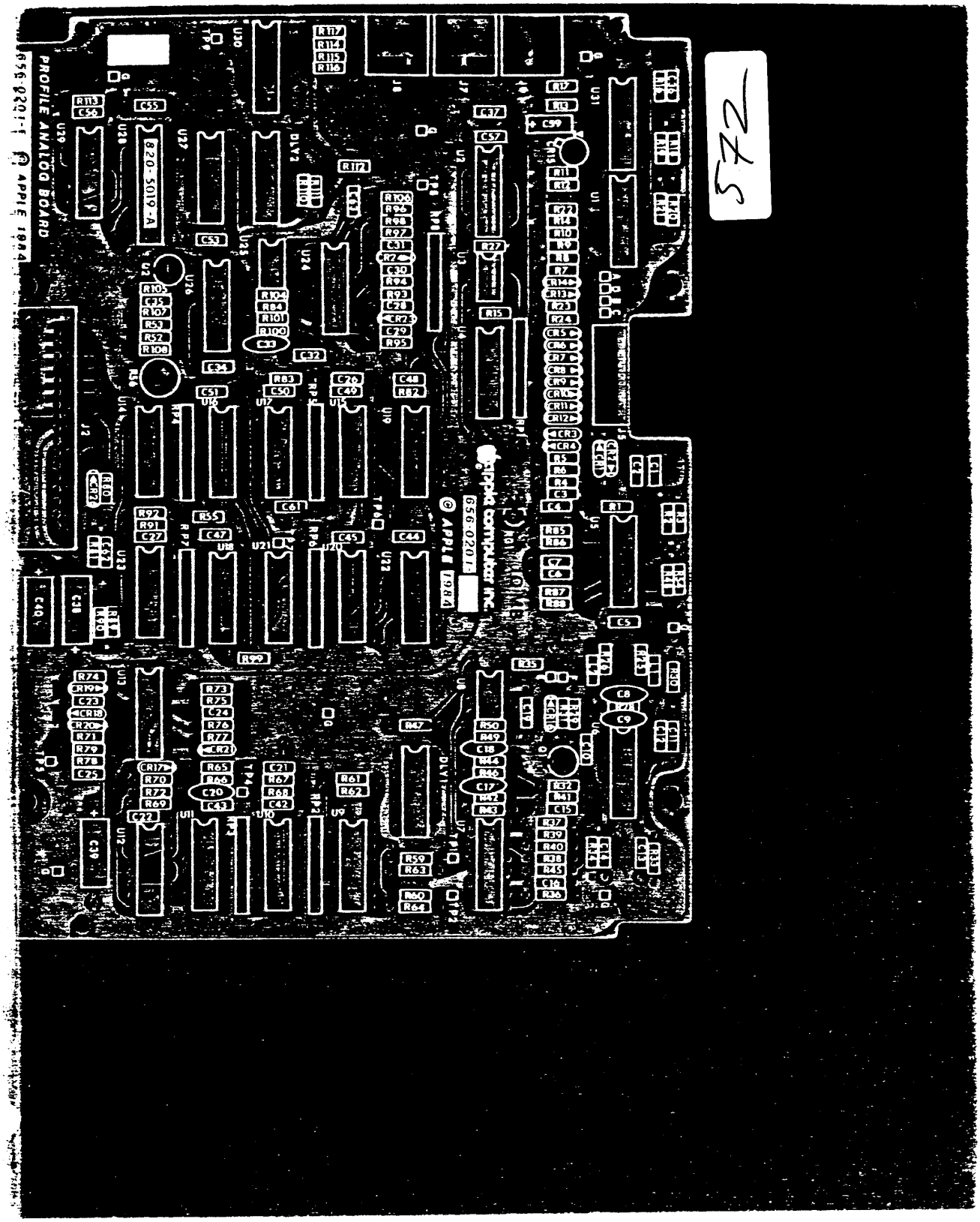
The circuit does this by acting as a "lock", which keeps the data referenced to the center of the VCO pulse "window". The lock lines up the data pulse in the center of the window for stability and proper detection. The VCO output and the data pulse can be monitored at (TP6 and TP7, respectively) to verify the leading edge of both signals occurring simultaneously.

The last portion of the circuit is the ECL to TTL level transition network at the bottom of the page. The detected NRZ data and clock signals are fed to the Controller board through connector J2.

8-16

9-16

572



573

PAGE SIX

On this page is the VCO circuit itself. The main portion of the circuit is the "charge pump" formed with UC24 and the network around it. The voltage level of the circuit is controlled by the INC -/+ and DEC -/+ signals from page 5. The input of the network acts as a "sample and hold" circuit, setting the current level of the ouput transistors.

The output passes through a low pass RC filter to a buffer amp UC25. This serves to buffer the current that sets the voltage control level at the MV104 varactor, controlling the VCO frequency. The output of the charge pump can be monitored at TP8.

The VCO output is approximately 20Mhz, and is routed to UC22 on page 5. This is a divide by 4 circuit and the resulting 5Mhz signal is the 200nsec VCO "window" into which the incoming read data is locked.

10-16