Roadrunner 3 is the lab name for the 9133D and 9134D. The 9133D has a double sided Sony floppy drive and a 15 Megabyte Seagate hard disc. The double sided Sony floppy in the 9133D is identical to the floppies in the 9122D.

****

SUBSET/80 PROTOCOL. Communications between Roadrunner 3 and the host controller take place via HP-IB. The protocol used on HP-IB by Roadrunner 3 is the SUBSET/80 protocol. Roadrunner 3 is a SUBSET/80 device. All SUBSET/80 devices must respond to the entire SUBSET/80 command set. For more information on SUBSET/80 see the document "SUBSET/80 FOR SMALL HARD DISCS AND FLEXIBLE DISCS". This document is sufficient to write a driver for all SUBSET/80 devices. The driver firmware should be the same for all SUBSET/80 devices.

The firmware for the floppy on Roadrunner 3 is the same code as used in the 9122D. The "9122D ERS" is a good source of information on the Roadrunner 3 floppy.

****

CONFIGURATION JUMPERS. On the Roadrunner 3 controller board there are 4 jumpers located near the center of the board, close to the Western Digital controller chip. For each jumper there are 3 pins.

```
                -------------
            A   x     x     x  E
                -------------

                -------------
            B   x     x     x  F ------------          CONFIGURATION JUMPERS

                -------------
C   x     x     x  G
                -------------

                -------------
D   x     x     x  H
                -------------
```

If you put the jumper nearer to the A , then the A is selected. If you place the jumper near the E, then the E is selected. At power up the firmware looks at these jumpers to determine whether it should look for a floppy or not and what size hard disc it should expect. The different selections are as follows:

CONFIGURATION JUMPERS

JUMPERS	BINARY	DEC	DRIVE SIZE	FLOPPY?
EFGH	0000	0	ST419 15 MEGABYTE	NO
AFGH	0001	1	ST225 20 MEGABYTE	NO
EBGH	0010	2	ST451 40 MEGABYTE	NO

H = WD10D0 + 1100
D = WD2010

ABGH	0011	3	NOT USED	
EFCH	0100	4	ST419 15 MEGABYTE	YES
AFCH	0101	5	ST225 20 MEGABYTE	YES
EBCH	0110	6	ST451 40 MEGABYTE	YES
ABCH	0111	7	NOT USED	

-- ANY OTHER JUMPER POSITIONS ARE NOT USED--

****

HP-IB ADDRESS SWITCH. The HP-IB address switch is a rotary switch with decimal numbers. This switch setting has the following meanings:

HP-IB switch = 0 through 7   HP-IB address of device is
the same as the switch
setting. Hard disc = unit 0.
Floppy = unit 1.

HP-IB switch = 8 or 9   HP-IB address of device is
0. Hard disc = unit 1. Floppy
= unit 0.

Thus the HP-IB switch is being used for more than just the HP-IB address of the device. It is being used to change whether the floppy is unit 0 or unit 1. This was necessary for the 150 since certain versions could only boot from HP-IB address 0 and unit 0. Note that the actual HP-IB addresses are still numbers from 0 through 7. 8 and 9 map into address 0.

SECTOR SIZES. The floppy can be formatted with sector sizes of 256, 512 or 1024 bytes. See the 9122D ERS for more information. The SET FORMAT OPTIONS command is used by the floppy to allow the host to specify a different format.

The hard disc can be formatted with block sizes of 256 bytes or 1024 bytes. This can not be done by the SET FORMAT OPTIONS command but requires changing a jumper on the controller board.

The hard disc block size jumper is located near the 2K by 8 byte buffer RAM chip. It is labelled 256 on one side and 1K on he other side. Simply place this jumper in the position you desire. If you are changing a drive from one block size to another it will take a couple minutes to first power on. The power on routine is much slower than the normal 30 seconds. After the drive has finally come up, you must then format the drive. The format will be done in the new block size. It is advantageous to format the drive several times since each format may find and spare a less than perfect sector. A list of bad sectors is kept on the drive. The list is cumulative and each format may add to it. However, when you change block sizes, the list is started again without retaining previous spares.

CLEARING A FULL SPARES TABLE. The condition may occur where a controller board goes bad, the user tries a format, and the spare table is rapidly filled up. These are not real media defects but just the result of a bad controller board.

To clear this filled spares table, the service person should do the following:

1. Install a new controller board.
2. Jumper the drive for a block size other than the current one.
3. Power up the drive and wait for the the fault LED to go off.
4. Start a format using any available host.
5. Clear or cancel out of the format, or wait for it to

finish.

6. Power down the drive. Change the jumper back to the original block size.
7. Power up the drive and wait for the fault LED to go off.
8. Start a good format using the available host.

This procedure uses the fact that switching block sizes clears the table of spared sectors.

CONFIGURATION OR VOLUME SWITCH. On the back of the board there is a red clock type switch with the numbers 0 through 9. To change this switch requires a small screw driver. The switch is the configuration switch or volume switch. It allows the user to divide the hard disc into multiple volumes. These volumes are physical volumes which can be formatted independently with different interleaves and can have different file systems on them. To use this feature, your host must support multiple volumes. If the host only supports a single volume, volume 0, then this feature is of no use. The following divisions will be made on the 15 Megabyte drive for Roadrunner 3:

15 MEGABYTE ST419

Configuration Setting	Number of Volumes	Size of Volumes	
		256 bytes/sector	1024 bytes/sector
0	One	14.84 Mbyte/volume	16.64 Mbyte/volume
1	One	14.84 Mbyte/volume	16.64 Mbyte/volume
2	Two	7.37 Mbyte/volume	8.23 Mbyte/volume
3	Three	4.91 Mbyte/volume	5.47 Mbyte/volume
4	Four	3.64 Mbyte/volume	4.03 Mbyte/volume
5	One	12.29 Mbyte/volume	13.76 Mbyte/volume
	One	2.51 Mbyte/volume	2.76 Mbyte/volume
6	Six	2.41 Mbyte/volume	2.65 Mbyte/volume
7	One	9.83 Mbyte/volume	11.00 Mbyte/volume
	Two	2.46 Mbyte/volume	2.70 Mbyte/volume
8	Eight	1.77 Mbyte/volume	1.93 Mbyte/volume
9	One	7.32 Mbyte/volume	8.18 Mbyte/volume
	Three	2.46 Mbyte/volume	2.70 Mbyte/volume

To change the configuration switch, first turn off the power, then select the kind of divisions you want. Now turn the power back on. Next you will have to reformat any volumes you plan to use to get directories written. All old data will be lost when you format the volume.

POWER UP ON THE ROADRUNNER 3

1. YELLOW FAULT LED DURING POWER UP.

When you first turn on the power to the 9134D or 9133 D the red hard disc access LED and the yellow fault LED go on. The yellow fault LED will stay on until the power up self test is finished. The red hard disc access LED goes off when the hard disc testing is done. The floppy red LED will go on during part of its test also. Unit 0 is tested first followed by unit 1.

If the yellow selftest LED stays on after the red drive access LEDs have both gone off, then something failed. If the 9133D or 9134D comes on line it will enable parallel poll and you can read the status. Do a Request Status command to find out what is wrong. The status information tells you what failed. If the Roadrunner 3 does not come on line, and does not respond to commands, the yellow LED will tell what failed.

1. LED on 6 seconds, blinks off 1 time, repeats.
   Rom checksum is wrong. Change the ROMs.

2. LED on 6 seconds, blinks off 2 times, repeats.
   Processor RAM is bad. Change the RAM located near
   the HP-IB switch.

3. LED on 6 seconds, blinks off 3 times, repeats.
   Buffer RAM is bad. Change the RAM located near the
   Western Digital WD1010 or WD2010. It has the 256/1K
   jumper next to it.

4. LED on 6 seconds, blinks off 4 times, repeats.
   The 4 configuration jumpers are set to an illegal
   configuration. Check the settings with the table given
   on an earlier page. These jumpers are labeled ABCDEFGH
   and are located near the center of the board.

5. LED on 6 seconds, blinks off 5 times, repeats.
   The HP-IB chip is bad. Replace the HP-IB chip.(8291A)

6. LED on 6 seconds, blinks off 6 times, repeats.
   The 6809 microprocessor is bad. Replace the 6809.

Of course, if the processor can't execute any code, the above blinking won't work. Also if the microprocessor RAM is totally bad or missing the above blinking also won't work.

2. TESTS DONE DURING POWER UP

1. The 6809 microprocessor is checked.
2. A checksum of the ROM is done and checked.
3. The entire microprocessor RAM is checked thoroughly.
4. The entire buffer RAM is checked thoroughly.
5. The configuration jumpers are read and checked for a valid configuration.
6. Unit 0 is tested. If unit 0 is a hard disc the following tests are done.

HARD DISC POWER UP TESTS.

7. The WD1010 or WD2010 hard disc controller chip is tested.
8. The WD1100 ECC chip is tested if it is there.
9. The controller waits for the hard disc to come up ready and with Seek Complete set. We wait for about 30 seconds before setting an error.

10. The seek command is tested.

11. The restore command is tested.

12. We read in the spares table and the information table. These are put into RAM.

13. The hard disc spindle speed is checked.

14. The track 00 indicator on the drive is checked.

15. We restore the drive to track 0.

16. We write sequential bytes to all the sectors on the selftest cylinder, which is the inner most cylinder. This write is done with an implied seek.

17. We seek back to cylinder 0.

18. We do long reads of these sequential bytes we just wrote, checking each byte and checking that the ECC was computed correctly.

19. Next we write the repeating byte "E5H" into all sectors on the selftest cylinder.

20. All sectors on the selftest cylinder are read with a normal read, checking the ECC bit and checking that each byte is "E5H".

21. Next we intentionally write an error using a long write command and then read it back to check that the ECC chip detects bad sectors.

22. Finally for the hard disc we do a check for actuator hystereses problems. We seek to cylinder 0, do a Scan Id command to check that we are on cylinder 0. Then we seek to the maximum cylinder and again check if we made it. We then seek back to cylinder 0, check, and step out to cylinder 1 and check. We are essentially just stepping the actuator and seeing if we get to where we should be.

Next unit 1 is checked. If unit 1 is a floppy, then the following tests are done:

FLOPPY POWER UP TESTS

23. The floppy disc controller chip is tested.

24. A seek test is done. This steps the head in and out.

25. If there is a piece of media inserted then the index period is measured and checked.

26. If the disc in the drive is not of a HP double-sided format, then the testing is complete. Otherwise, if the disc is write protected, then only a read test is performed. A write, read, compare test is performed if the disc is not write protected. All writing is done in non-data areas. No customer data is at risk.

27. Finally the 8291 HP-IB chip is tested and put on line.

If all the tests pass, then the yellow fault LED goes out and Roadrunner is ready for use. All the tests run at power up can also be run by using the SUBSET/80 Initiate Diagnostic command.

****

SERVICE CHECKS

NOTE!! If you have a 9134D and can't get any service checks to work, jumper the drive as a 9133D and then try the tests. Make sure to change the jumper back when you are done.

NOTE: Since the RAM test will clear all parameters, some of which are needed for other tests, you can't switch arbitrarily from test to test. The best sequence is as follows:

1. Turn off power.

2. Select the RAM test(0,0) with the loop test switch, select switch and HP-IB switch all in the position you want.

3. Turn on power. Device will do all or part of power on selftest and will then start doing RAM test. When the test is completed, the selftest LED should blink 5 times. If it doesn't, the test failed.

4. Turn power off. Select the next test (0,1).

5. Turn on power. Device will do all or part of the power on selftest and will then start doing the test specified. When the test is completed, the selftest LED should blink 5 times. If it doesn't, the test failed.

6. Now select the next test you want. You can test anything but the RAM test. There is a 4-second wait between tests to "debounce" while you are changing the switches. When the test is completed, the selftest LED should blink 5 times. It it doesn't, the test failed.

7. You can do step 6 over and over until all tests are performed except the microprocessor RAM test. The tests are repeated so that you can see if they always fail or fail only occasionally.

LED FUNCTIONING

```
Start   1.  LED ON 4 seconds        -will stay here until test selection
        2.  LED OFF .5 seconds      -is the same for 4 seconds.
        3.  LED ON during test      -test in progress


        TEST PASSES                 TEST FAILS

        4.  LED ON/OFF 5 times      4.  LED stays ON.
             ((.5 seconds each).
        5.  Go to start.            5.  Go to start.
```

We are trying to model our service diagnostics on Sparrow 2 to make training of service personnel easier. Much of the following is taken from Greg Brake's memo on service diagnostics.

Service diagnostics are selftest routines which can be run by service to test definite portions of the Sparrow 2 and Roadrunner 3 board. There are several ways to test these products:

1. At power on a selftest of the processor, ROM, HP-IB chip,microprocessor RAM, buffer RAM and both drives is performed. The hard disc test includes reading and writing sectors, comparing each byte written, seeking back and form from track 0 to the maximum track, checking the spindle speed, and checking that the ECC chip functions correctly by introducing errors and correcting them.

2. With a host, the Initiate Diagnostic command can be given which runs a test similar to those at power on.

3. If no host is available, a switch or jumper can be set which causes Roadrunner 3 to go into the diagnostic test mode. It interprets the 4 bit HP-IB switch as a selftest to perform. The LED is blinked (to show that it works) and then the test as selected by the address switch settings is performed. A successful test is indicated by the LED blinking. A failure causes the LED to stay on. The test results are displayed for 5 seconds. If the jumper is left on, the same test will be performed again. When the service test switch is no longer in the diagnostic test mode, the Roadrunner will go through the power-up sequence and will again be in the peripheral mode of operation, waiting for commands over HP-IB.

The service diagnostic testing ability on Roadrunner 3 consists of being able to select and optionally loop on any one selected test from the following choices:

NOTE

The Test Switch much be "ON" while performing the following tests.

Select Switch	HP-IB Switch		
0	0	RAM	All possible patterns are written in all locations of both RAMS.
0	1	ROM	A checksum calculation is performed.
0	2	HP-IB	Two of the registers on the HP-IB chip are written and their contents verified.
0	3	FDC Chip	Two of the registers on the FDC chip are written and their contents verified.
0	4	Floppy Seek	Commands are given to the FDC to move the head on and off of track 0. The track 0 indicator is checked to see that movement occurred.
0	5	Winchester Seek	Commands are given to the WD1010 to restore to cylinder 0 and then step off of cylinder 0. The track 0 indicator is checked to see that it works.
0	6	Floppy Speed	The head is stepped to track 35 and loaded. The period of the index pulse is measured and compared against the specification. No test is performed if there is no medium in the drive.
0	7	Winchester Speed	The spindle speed of the drive is checked and compared with the allowed range.
1	0	Floppy Write/Verify	Every sector on the disc is written and the data is verified. All user data on the medium is lost.

1	1	Winchester Write/Verify	All sectors on the selftest cylinder are written and read. Each byte including the ECC is checked. Error correction is also checked. No user data is affected.
1	2	Floppy Verify	All sectors in the data area of the disc are checked for CRC errors. No user data is affected.
1	3	Winchester Verify	All sectors in the data area of the disc are checked for CRC errors. No user data is changed.
1	4	Floppy Format	The floppy disc is re-initialized with a 011 data pattern.
1	5	WD1010 Check	All read/write registers on the WD1010 are checked.
1	6	WD1100 Check	Writes data patterns to all four registers in all combinations and verifies them.
1	7	WD data buffer RAM test	Test the WD data buffer RAM near the 256/1K jumper.

The above can also be executed over the HP-IB bus using the Download command. A possible advantage of this mode of testing is that the error results are available and give more information than a blinking LED.

****

READ RECOVERY AND ERROR CORRECTION

When a sector is read and the read is not successful, the following algorithm is used: It is written in pseudo-code.

```
DO UNTIL YOU HAVE RECOVERED THE DATA OR HAVE DONE TWO PASSES
THROUGH THIS LOOP.
    READ  THE SECTOR. (1 TRY ON DATA ERRORS,  8 ON  HEADER
        ERRORS.
    READ THE SECTOR.  (8 TRIES ON DATA ERRORS,  8 ON  HEADER
        ERRORS.
    APPROACH  THE  TRACK  FROM THE INSIDE BY STEPPING IN
        THEN OUT.
    READ THE SECTOR. (8 TRIES ON DATA ERRORS, 8 ON HEADER
        ERRORS.
    APPROACH THE TRACK FROM THE OUTSIDE BY STEPPING OUT
        THEN IN.
    READ THE SECTOR. (8 TRIES ON DATA ERRORS, 8 ON HEADER
        ERRORS.
    TRY ERROR CORRECTION. TWO CONSECUTIVE READS MUST HAVE
        THE  SAME  SYNDROME  FOR ERROR  CORRECTION  TO  BE
        ATTEMPTED. 32 READS ARE DONE.
      RESTORE THE DRIVE TO TRACK 0 TO RECALIBRATE.
    ENDDO
```

Each time a read is done, if the head is on the wrong track, we seek to the correct track.

ERROR CORRECTION ON ROADRUNNER 3

The error correction polynomial which will be used on Roadrunner 3 and 4 is given below:

$$X^{32} + X^{28} + X^{26} + X^{19} + X^{17} + X^{10} + X^{6} + X^{2} + 1$$

If we use the WD1010 chip, this ECC will be in a separate chip, the WD1100-13. If the WD2010 is used, this ECC is included within the WD2010.

The polynomial is a computer generated polynomial. It was discovered by Neil Glover of Data Systems Technology and was selected for its insensitivity to short double bursts, good detection span and eight feedback terms. This polynomial allows for correction of a single burst of length up to 11 bits per sector.

Error correction is the last step taken in data recovery. If the bad sector can be read twice and gives the same syndrome each time, then the syndrome is placed into the reciprocal polynomial and clocked backwards until the register is all zeros except for the left most 11 bits. These 11 bits define the correction pattern. The number of clocks defines where the correction is to be made. We clock backwards through the 4 ECC bytes and into the data. If the polynomial register never has all zeros in the right most 21 bits, then error correction is not possible. In this case the error is too large.

With the WD2010, error correction can be done automatically by the chip so no firmware algorithm is necessary. With both the WD1010 and WD2010, the ECC generation and correction can be totally tested at power-on selftest or during the Initiate Diagnostic command. The reciprocal polynomial used during correction is:

$$X^{32} + X^{30} + X^{26} + X^{22} + X^{15} + X^{13} + X^{6} + X^{4} + 1$$

CORRECTION CAPABILITIES

1. Maximum record length(r) = 1038 bytes including check bits.

2. Maximum correction span (b) = 11 bits.

3. Degree of polynomial = 32

4. Single burst detection span when the code is used for error detection only = 32 bits.

5. Single burst detection span when the code is used for error correction:

   detection span = 20 bits for b=5 and r=270 bytes
   detection span = 13 bits for b=11 and r=270 bytes
   detection span = 11 bits for b=11 and r=1038 bytes

6. Double burst detection span when the code is used for error correction:

   detection span = 4 bits for b=5 and r=270 bytes
   detection span = 2 bits for b=8 and r=270 bytes

7. Non-detection probability = 2.3 E-10

8 Miscorrection probability:

   = 8.00 E-6 for b=5 and r=270 bytes

   = 6.40 E-5 for b=8 and r=270 bytes
   = 5.12 E-4 for b=11 and r=270 bytes
   = 2.01 E-3 for b=11 and r=1038 bytes

The capabilities of this polynomial are much superior to the Abramson 16 bit CRC polynomial used on Roadrunner 1 and 2. It allowed for correction of a single burst of up to 2 bits in length.

****

FORMATTING OR INITIALIZATION PROCESS ON THE HARD DISC

The following algorithm is used during initialization of the hard disc.

```
RESTORE TO TRACK 0
CLEAR THE TRACK STAGGER.
DO FOR EACH TRACK ON THE DISC.
    SET UP THE FORMAT TABLE WITHOUT SPARES.
    DO THE FOLLOWING TWICE.
        FORMAT THE TRACK.
        WRITE "E5H" INTO EACH SECTOR.(SPARE ANY BAD ONES)
        READ EACH SECTOR. (SPARE ANY BAD ONES)
    ENDDO
    STEP THE HEAD IN ONE TRACK THEN BACK OUT.
    DO THE FOLLOWING TWICE.
        FORMAT THE TRACK.
        WRITE "E5H" INTO EACH SECTOR.(SPARE ANY BAD ONES)
        READ EACH SECTOR. (SPARE ANY BAD ONES)
    ENDDO
    STEP THE HEAD OUT ONE TRACK THEN BACK IN.
    DO UNTIL THE TRACK HAS NO ERRORS OR FAILS 4 TIMES.
        SET UP THE FORMAT TABLE WITH SPARES.
        FORMAT THE TRACK.
        WRITE "E5H" INTO EACH SECTOR.(SPARE ANY BAD ONES)
        READ EACH SECTOR. (SPARE ANY BAD ONES.)
    ENDDO
    ADD THE SPARES TO THE SPARE TABLE ON THE DISC.
ENDDO
```

The Spare Table is a 256 byte table which is kept on a system cylinder of the hard disc. It has 3 bytes per entry. The first two bytes are the track number. The third byte is the bad location on that track. The track number in the table is defined as the cylinder times the number of heads plus the current target head number. The table is composed of entries as follows with 0FFFFFFH being in locations not used.

```
SPARE_TABLE        track = 000AH  bad location = 00H
           track = 000AH  bad location = 02H
           track = 0012H  bad location = 00H
           track = FFFFH  bad location = FFH
      . . .
SPARE_TABLE_END    track = FFFFH  bad location = FFH
```

Bad locations are numbered 0 through the maximum sector number. The last entry in the Spare Table will always be 0FFFFFFFH. Entries are aranged in order ;by track and by location. The Spare Table can be read by using special Downloaded commands. The bad locations are number with an interleave of 1 starting at the index with sector 0.

****

P1 TO P10 STATUS ERROR BITS.

The SUBSET/80 document tells you what each of the status bits in the 8 byte field mean. In certain cases, it also specifies a meaning for bytes P1 through P6. The bytes P7 through P10 are always left for the device to put device specific information there. The following lists the meanings of these bytes for Roadrunner 3.

If the Initiate Diagnostic command or power up selftest fails for the floppy, the P1 through P6 bytes have the following meaning:

Meaning	P1	P2	P3	P4
FDC failure	00	00	00	01H
Seek tests failure	00	00	00	02H
Index test failure	A-	-A	00	03H
Write test failure	XX	B	00	04H
Read test failure head 0	XX	B	00	06H
Read test failure head 1	XX	B	00	07H
Read compare error head 0	C-	-C	00	08H
Read compare error head 1	C-	-C	00	09H
Read test failure head 0	XX	B	00	0AH
Read test failure head 1	XX	B	00	0BH

where A*10.5 microseconds +90 msec is the index period.

B reflects the FDC status:

```
     1xxx xxxx -- no disc in drive
     x1xx xxxx -- write protected
     xxx1 xxxx -- ID read error
     xxxx 1xxx -- CRC error
     xxxx x1xx -- hardware failure, and C is the address
     of the bad data.
```

XX means there is no meaningful data in that field.

P5,P6 will be set to the unit that failed.

When you didn't just do an Initiate Diagnostic command or just power up, the floppy uses P7 through P10 for the following information: The P7-P10 bytes are used as a stack, so the most recent error byte is in P7.

P7-P10 ERRORS SET BY FLOPPY

B1H Seek failed in Position_head

B2H Read address failed in Position_head

B3H Read address failed in Position_head

B4H Restore failed in Position_head

B5H,X timeout in Wait_for_FDC, FDC status

B6H Wait_for_FDC failure in seek routine.

B7H Wait_for_FDC failure in Restore

B8H Seek error in Restore

B9H,X,X M_INIT_MEDIA, head 0 tracks spared, head 1 tracks spared.

BAH No system cylinder, no read/write tests performed by M_INIT_DIAG.

BBH A read compare was performed by M_INIT_DIAG.

BCH,X Busy bit set in FDC status in FDC_TERM, FDC status

BDH,X INTER_FDC_ERROR called with no bits set, fdc status.

BEH Unit fault in STP_HD_IN

BFH Unit fault in STP_HD_OUT

P7-P10 ERRORS SET BY THE CONTROLLER:

F0H A command in progress was aborted to do a Clear or CAncel command.

F1H The "ERR" bit was set in the 8291 HP-IB chip.

** WARNING - More lines in range than will fit in terminal memory **SCREEN MODE

F2H The "END" bit was set in the 8291 HP-IB chip.

F3H The "BI" bit was set in the 8291 HP-IB chip.

F4H The Data buffer is bad and failed during the Initiaite Diagnostic command.

P7-P10 ERRORS SET BY THE HARD DISC.

Rather than setting one byte, the hard disc sets all 4 bytes in the P7-P10 area. For hard disc information, the P7-P10 bytes will look like the following:

P7	=	AAH
P8	=	ERROR BYTE LISTED BELOW:
P9	=	WD1010 OR WD2010 STATUS REGISTER
P10	=	WD1010 OR WD2010 ERROR REGISTER

P9= WD2010 OR WD1010 STATUS REGISTER

BITS:	7	6	5	4	3	2	1	0
	BSY	RDY	WF	SC	DRQ	-	CIP	ERR

P10= WD2010 OR WD1010 ERROR REGISTER

BITS:	7	6	5	4	3	2	1	0
	BB	CRC	-	ID	-	AC	RK	DM

HARD DISC ERROR BYTES IN P8

A0H CHECK_WD1010. Some WD1010 Status Register Error is set like NOT BUSY, NOT READY, NOT SEEK COMPLETE, WRITE FAULT, or COMMAND IN PROGRESS. Check P9 Byte.

A1H CHECK_WD1010. There must be a header error. Check P9,P10 for which bits are set.

A2H CHECK_WD1010. There is a data CRC error.

A3H CHECK_WD1010. A WRITE fault error was detected by the WD1010.

A4H SET_TASK_FILE. A register must be bad on the WD1010 chip. What is written to it doesn't compare with what is read.

A5H WAIT_NOT_BUSY. We timed out waiting for the status to be ready, seek complete and not busy. The time we wait is an input to the module.

A6H FAST_WRITE,FAST_READ. The S_D_H register on the WD1010 is bad. What is written to it is not what is read.

A7H WAIT_INTRQ. We timed out after waiting about 2 seconds for intrq to equal 1. It happens at the end of the command.

A8H WAIT_0_BDRQ. We timed out waiting for bdrq to equal 0. We wait about 2 seconds before timing out.

A9H WAIT_1_BDRQ. We timed out waiting for bdrq to equal 1. We wait about 2 seconds before timing out.

AAH H_INIT_MEDIA. We keep finding bad sectors on this track on the final pass. At least 4 times we formatted with spares and then found another bad sector. Because of this a unit fault error bit was set.

ABH ADD_BAD_LOCATION. There are more than 6 bad sectors. This is too many on one track to continue, so the format was aborted.

ACH ADD_BAD_SECTOR, ADD_BAD_LOCATION. There must be a firmware error. The sector spared should have been in the format table, or there is no more room in the bad locations table, respectively.

ADH UPDATE_SPARES. We couldn't read two good sectors which compared when we tried to read the spare table off the disc, so the spare table was cleared.

AEH UPDATE_INFO. We couldn't get two sectors which compared when we tried to read the information cylinders off the disc, so the information table was cleared.

AFH FINAL_SPARES. The spare table is full, something must be wrong. Drives can't have this many bad sectors!

B0H H_EXEC_READ. Data was good after approach from the inside. We had to approach the track from the inside to recover the data.

B1H H_EXEC_READ. We had to approach the track from the outside to recover the data.

B2H TEST_WD1100. The WD1100 ecc chip must be bad,replace it.

B3H TEST_WD_BUFFER. The WD data buffer ram must be bad. It failed a read/write test. It's near the 256/1k jumper.

B4H H_EXEC_READ. The host didn't take all the bytes. A message length error was set.

B5H H_POWER_UP. The drive did not come up (ready,not busy, seek complete) after we waited over 30 seconds.

B6H C_CONFIGURE. The jumper that sets the configuration is set at an illegal configuration.

B7H PUT_VOLUME_PTR. Something is wrong with the configuration choosen. there is an inconsistency. This configuration can not have this many volumes.

B8H DO_ERR_CORRECT. The error was a header or data mark error, so error correction was not possible.

B9H DO_ERR_CORRECT. No error correction was needed as the data was acquired without errors on one of the reads done to get the syndrome.

BAH DO_ERR_CORRECT. The 2 syndromes acquired in this module didn't agree so no error correction was attempted. It is better to do no error correction than risk the possibility of miscorrection.

BBH DO_ERR_CORRECT. Error correction was attempted, but the error was too large and could not be corrected.

BCH DO_ERR_CORRECT. Error correction was used and was successful. Congratulations Neil.

BDH SELF_RW_TEST. We created a bad sector and then read it back. The ECC should have been 1, but it was zero. Something is wrong with the wd1100 or some other hardware on the ecc line.

BEH SELF_WRITE. More than 3 bad headers were found on the selftest cylinder when we tried to write all the sectors. this is too high an error rate and something must be wrong.

BFH SELF_LREAD. More than 5 bad sectors were found during the long read and compare of each sequential byte. This is too high an error rate and something must be wrong.

C0H SELF_READ. In reading all the sectors on the selftest cylinder and checking for data bytes of e5h, there were more than 5 bad sectors. This is too high an error rate and something must be wrong.

C1H TEST_H_TRACK00. There must be something wrong with the track zero indicator. It doesn't register when you restore to cylinder 0.

C2H TEST_H_SPEED. The spindle speed as measured from the leading edge of the index to the next leading edge is out of spec.

C3H TEST_H_SPEED. The index pulse doesn't appear to change. There must be something wrong with the drive, or input driver.

C4H C_CONFIGURE. The volume setting is bad for this device. this will probably keep the microprocessor from going on line.

C5H C_INIT_DIAG. The checksum on the ROM failed.

C6H M_WRITES. There was an ECC or WRITE FAULT error during a write.

C7H C_INIT_DIAG. The data buffer in the microprocessor ram is bad.

C8H PRESET_KEY. During a format, the area of the volume used for the keys is filled with 0FFH's. In writing these sectors, we were unable to write even 4 sectors so this status was set. I bet the head actuator is not moving.

C9H CHK_HYSTERESIS. The drive's actuator has a hysteresis problem. When you step out from cylinder 0, it takes more than one step to get to cylinder 1.

CAH FAST_WRITE. There has been a WRITE fault error or the ECC was not zero after the write was done.

D0-D3H LOCATE AND VERIFY or LOCATE AND WRITE failed.

D0H WRITE FAULT = 0, (NO ERROR) ECC=0 (NO ERROR)

D1H WRITE FAULT = 0, (NO ERROR) ECC=1 (ECC ERROR)

D2H WRITE FAULT = 1, (WF ERROR) ECC=0 (NO ERROR)

D3H WRITE FAULT = 1 (WF ERROR) ECC=1 (ECC ERROR)

CONTROLLER HARDWARE DESCRIPTION

BS 10/5 (ER3B3)

INTRODUCTION

The intent of this document is to discuss the theory and function of the ROADRUNNER III controller board (9133-66510). The document will cover functional basics of the controller as well as some specifics relative to the hardware. All of the theory of operation will be presented in the following three sections: General Overview, Block Diagram and Explanation, and Hardware Description.

GENERAL OVERVIEW

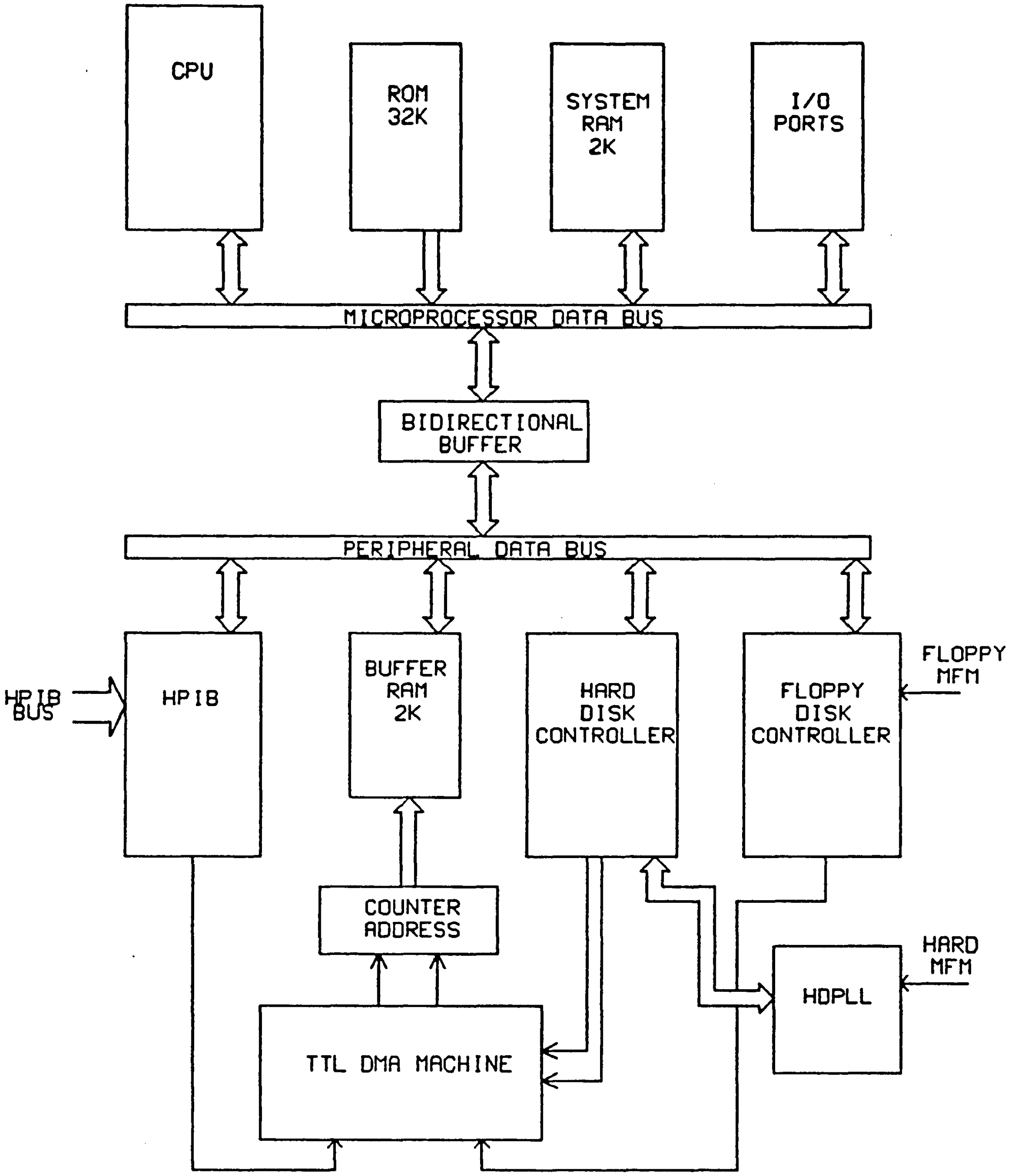
ROADRUNNER III is a mass storage peripheral containing two separate disk drives. The floppy drive is a double sided 3.5 inch drive with a formatted capacity of 730 kbytes. The hard disk is presently a 15 Mbyte Seagate drive. The 9133-66510 controller board is the only circuit board in the product which controls both the floppy and hard disks.

A host computer system is connected to the ROADRUNNER III via the HPIB (IEEE Std. 488) connector. The protocol used in communicating between the host and RRIII is the SUBSET 80 protocol. The various commands defined for this protocol will not be discussed here but can be found in SS80 documents.

The purpose of the controller board is to receive SS80 commands from the HPIB bus, transfer appropriate data between the host and the disks, and report back to the host the status of the execution. Besides the obvious tasks of reading and writing data to the disks, the controller must be able to handle such things as error detection and correction, formatting at various interleaves and staggers, sparing of bad areas, monitoring wear on floppy disks, performing self and user selected diagnostic tests, etc. The specifics of each of these functions will not be presented in this document but they can be found in code listings and software explanations.

BLOCK DIAGRAM AND EXPLANATION

Shown below is a block diagram of the 9133-66510 controller board.



The block diagram can best be explained starting at the upper left hand corner. The microprocessor at the heart of the controller board is the Motorola 68B09. This microprocessor has gained popularity by being used in past products. It is basically an 8 bit processor but it does have the capability to do several 16 bit operations. Internally, it runs at 8 Mhz but its external cycle time is 500 ns. The CPU handles all of the movement of data, handling of bus arbitration, and acting as the master controller of all operations of the controller board.

Connected directly to the microprocessor data bus is the ROM, SYSTEM RAM, and I/O PORTS. The ROM consists of two 16K X 8 eproms and the SYSTEM RAM is 2K bytes of static memory. There are three input and three output ports. The input ports are used to monitor various disk and system signal, jumpers, hpib addresses, etc. The output ports are used to control disk operations, dma operations, and general system signals. The I/O ports are covered in detail in the Hardware Description section.

Upon examining the block diagram it should be noted that the controller board has two data busses. These two data busses are separated by a bi-directional transceiver. The microprocessor can always access the ROM, SYSTEM RAM, and I/O PORTS. However, during certain times the microprocessor can be isolated from the PERIPHERAL data bus. There are two busses so that dma transfers can be taking place between one of the drives and the HPIB, and at the same time the microprocessor can be monitoring activity through the I/O ports. It is essential to have two busses for reasons other than dma. When the HARD DISK CONTROLLER begins transferring data from or to the hard disk drive, it must take over the data bus and read/write control lines so that it can directly transfer data into or out of the BUFFER RAM. The PERIPHERAL DATA BUS is isolated from the MICROPROCESSOR DATA BUS whenever data is being transferred between either of the DISKS and BUFFER RAM or between the HPIB and BUFFER RAM.

The HPIB block shown represents the Intel 8291-a HPIB controller chip and its two transceivers. Commands, data, and status information coming or going pass through this block. This block handles all HPIB handshaking, parallel polling, primary and secondary addressing, and all other miscellaneous HPIB protocol.

The BUFFER RAM is a 2k static ram chip which is used only when data transfers involve either of the DISKS or the HPIB. The CPU can read and write data to the BUFFER RAM but it is normally used for buffering data coming or going to either the hard or floppy disks. As can be seen from the block diagram, the BUFFER RAM's address lines are controlled by a counter. The counter is incremented every time a read or write occurs to the BUFFER RAM. This means that under normal operation a complete sector of data will be written/read into the BUFFER RAM, then that same sector will be read/written out enroute to a different location.

The controller board has a very simple dma machine which can be set up to transfer data between either BUFFER RAM and HPIB or between the FLOPPY disk and BUFFER RAM. This machine simply generates read/write strobes to the devices involved each time a request for transfer is made. This dma machine does not need to handle data transfers between the HARD disk and BUFFER RAM because the Hard Disk Controller chip does this.

The HARD DISK CONTROLLER is the WD1010 chip. This chip handles all interfacing between the controller CPU and the drive. This block also includes an error detection and correction chip. Should any ECC errors occur while the HARD disk is transferring data this chip will flag the error and generate a syndrome from which a correction

can be made. Working in conjunction with the hard disk controller chip is a phase lock loop I.C. which is responsible for locking onto and tracking the mfm data coming off the hard disk.

The FLOPPY CONTROLLER block is the WD2793 chip. This floppy controller acts as the interface between the CPU and the floppy drive. Included in this chip is CRC logic, write precompensation, and an internal phase lock loop and data separator. Commands are given to this chip from the CPU such as "read sector". This chip then controls the signals to locate the desired sector, lock onto the mfm data stream, transfer the data, and return a status upon completion.

The best explanation of how all of these functional blocks work together is by an example. Suppose a user wishes to read one sector of data from the floppy disk. The following scenario applies. Initially the CPU has appropriate bits set in the I/O PORTS to enable access to the PERIPHERAL DATA BUS. Thus, the CPU can continuously poll the HPIB chip to see if any commands have come in. Upon seeing a command come in, the CPU deciphers the command as a "read sector" command. A command is given to the Floppy disk controller to seek to the desired cylinder. Upon successful completion of the command the CPU then sets appropriate bits in the I/O PORTS to enable the dma machine to transfer data from the FLOPPY CONTROLLER to the BUFFER RAM. After giving the "read sector" command to the FLOPPY CONTROLLER the CPU isolates the two data busses and watches an input port to determine when the FLOPPY CONTROLLER is done. Meanwhile the FLOPPY CONTROLLER finds the right header and begins issuing data requests to the dma machine, one for each byte coming in. The DMA machine writes each byte into the BUFFER RAM. When the command is finished the CPU then re-enables the bidirectional buffer so that access to the PERIPHERAL DATA BUS is possible. The CPU then sets the HPIB chip up to send the data read from the disk back to the host. After isolating the two busses again and re-configuring the dma direction the transfers begin with the dma machine moving the data from the BUFFER RAM to the HPIB chip as the HPIB controller issues requests for transfers. This continues until the entire sector of data has been transferred. Then a status is sent back to the host and the command is essentially finished.

Of course this scenario is very simplistic in that no errors were encountered and much of the detail was left out but its intent is to clarify some of the interaction that takes place on the controller board.

HARDWARE DESCRIPTION

The following description is meant to give a fairly detailed explanation of the present hardware for the RRIII controller. This hardware description is for the revision B schematic. The schematic is divided up into 54 zones (6 rows labeled A thru F, 9 columns labeled 1 thru 9). Throughout this explanation references will be made to circuits drawn in the zones mentioned. This description is intended to focus on details unique to this design not on fundamentals common to most all designs. The hardware description will be divided up into the following function-related areas:

1 - MICROPROCESSOR AND CLOCKS
2 - ADDRESS MAPPING
3 - DATA.BUS STRUCTURE
4 - SYSTEM MEMORY
5 - I/O PORTS
6 - HPIB INTERFACE AND CONTROLLER
7 - FLOPPY INTERFACE AND CONTROLLER
8 - HARD DISK CONTROLLER AND ECC
9 - HARD DISK READ CIRCUITRY
10 - HARD DISK WRITE CIRCUITRY
11 - BUFFER RAM AND ADDRESS COUNTER
12 - DMA LOGIC
13 - MEMORY STRETCHING

1 - MICROPROCESSOR AND CLOCKS

The microprocessor used in this design is the Motorola 68B09. It is an 8 bit processor with the capacity to handle many 16 bit manipulations. The processor is connected and operated in the standard way. Our design involves using two of its three interrupt lines. The non-maskable interrupt is never used thus NMIn is tied high. Also tied high are the DMA bus request and the Halt inputs. Each of these three inputs are pulled high through separate resistors so that during testing any or all can be grounded without affecting the others. One might tend to think that with DMABRQn and HALTn tied high, the processor will never tri-state its data and address lines. This assumption is wrong. There is an instruction called "SYNC" which when executed places the processor in the tri-state mode. During certain DMA transfers we use this instruction. Because the addresses are floating during this "SYNC" state, a Valid Memory Address (VMA) must be generated which disables all of the processor address decoding (zones C3-D3) during this condition. Though the processor is not accessing anything during this time, the DMA machine may be transferring data. This will be covered in the "DMA" section. The 68B09 requires an 8Mhz clock and internally it divides this clock by 4 and outputs two system clocks "E" and "Q". These clocks are both 2 Mhz and quadrature in relation to each other. "Q" leads "E" by 90 degrees. Besides providing 8Mhz for the microprocessor, there is a divide down counter in zone E1 which divides the 8Mhz down to 2Mhz for the Floppy Controller chip. It would be nice if we could use one of the 2Mhz outputs of the processor for this clock but this is not possible since we have to stretch these clock cycles during certain operations. The only other frequency needed for the controller is 10Mhz for the Hard Disk. This clock is generated by a clock oscillator in zone B7.

2 - ADDRESS MAPPING

The physical memory map is shown below:

ADDRESS	DEVICE SELECTED
C000 - FFFF	ROM2
8000 - BFFF	ROM1
4800 - 7FFF	RAM (DUPLICATE)

4000 - 47FF	RAM (DUPLICATE)	DUPLICATE means that more than one address selects a device. For instance the addresses 0040,0140,0240,..3F40 all select I/O ONE.
0100 - 3FFF	I/O (DUPLICATE)	
00C0 - 00FF	I/O THREE	
0080 - 00BF	I/O TWO	
0040 - 007F	I/O ONE	
0038 - 003F	NOT USED	
0030 - 0037	NOT USED	
0028 - 002F	BUFFER CLEAR	
0020 - 0027	BUFFER RAM	
0018 - 001F	ECC	
0010 - 0017	HARD DISK CONTROLLER	
0008 - 000F	FLOPPY CONTROLLER	
0000 - 0007	HPIB CONTROLLER	

The memory mapping is organized so that all of the devices on the isolated data bus can be accessed using one byte addresses (this is done using Direct Page addressing and setting the high order address byte to 00). This arrangement allows for faster real time execution of programmed I/O if this mode of transfer is desired. However, in most cases the disk transfers will be done using the TTL DMA machine which will be explained later. All of the address decoding is shown in zones D3 and C3. Perhaps the only decoding that needs special explanation is the BUFFER CLEAR decode. Whenever the microprocessor reads or writes from this address, two things happen: Firstly, the BUFFER RAM address counter is reset to 0, and secondly, the ECC chip is reset. This strobe was directly decoded to speed up execution time.

3 - DATA BUS STRUCTURE

In examining the schematic for this design, it should be obvious that there are two data busses. One of them is the microprocessor DATA BUS. This bus goes to the system RAM and ROM as well as the I/O ports. The other bus is called the ISOLATED DATA BUS because the bi-directional buffer in zone C3 can be tri-stated thus isolating the microprocessor data bus from the isolated data bus. The "peripheral" devices found on the isolated data bus are: HPIB CONTROLLER, BUFFER RAM, HARD DISK CONTROLLER, FLOPPY CONTROLLER, and ERROR CORRECTING CHIP. By having the microprocessor isolated from these peripheral devices, DMA data transfers can be taking place between the DISCS or the HPIB and the BUFFER RAM while the microprocessor can be monitoring system activity through the I/O ports. When the busses are isolated there will be one of two devices controlling data flow on the isolated data bus. Either the HARD DISK CONTROLLER, or the TTL DMA machine will be controlling the chip selects and R/W signals. Being able to isolate the two data busses is mandatory due to limitations (in terms of bus arbitration) of the HARD DISK CONTROLLER chip.

4 - SYSTEM MEMORY

The System Memory includes two 16K X 8 EPROMs and one 2K X 8 RAM. The microprocessor has access to it at all times regardless of whether or not the data bus is split. This allows the processor to continue executing ROM code while the bus is split and DMA is occurring. It should be noted that the 2K ram is selected by a 4K decode. Therefore, the programmer must be consistent in addressing the RAM in either the lower 2K or upper 2K of the 4K address.

5 - I/O PORTS

In zones E6 thru E9 and F6 thru F9 are found the I/O ports. There are three input buffers and three output registers. Each of these ports are explained below:

I/O ONE INPUT PORT:

HDIND	HDINT	BDRQ	BCSn	DC3	DC2	DC1	DC0
D7	D6	D5	D4	D3	D2	D1	D0

DC0-DC3: These 4 bits are used to identify what kind of hard disk is being used. The controller is very versatile in that it will handle varied drive capacities from 15 to 40 Mbytes. These 4 bits are jumper selectable and will be set by the factory depending on what drive is put in.

BCSn: This input lets the processor know when the Hard Disk controller is transferring data between BUFFER RAM and the disk. It is used to monitor activity while the data bus is isolated. A '0' means the Hard Disk controller is transferring data.

BDRQ: This bit is another status bit reflecting certain conditions that are occurring while the Hard Disk controller is in control of the isolated data bus. A '1' means that the Hard Disk is initiating transfers to/from the BUFFER RAM.

HDINT: This bit when set tells the processor that the Hard Disk has finished the command it was last given.

HDIND: This bit is set when an index pulse occurs on the hard disk.

I/O ONE OUTPUT REGISTER:

WDRESETn	BRDY	DISWRPRE	HS2	HS1	HS0	DS1	FLTn
D7	D6	D5	D4	D3	D2	D1	D0

FLTn: This bit when '0' turns on the FAULT l.e.d.

DS1n: When this bit is a '1' the Hard Disk is selected and the ACCESS L.E.D. should be lit.

HS0-HS2: These 3 bits select the head on the hard disk. They are asserted high.

DISWRPRE: When this bit is a '1' write precompensation on the hard disk is disabled.

BRDY: This bit is set to indicate to the Hard Disk controller that the BUFFER RAM is ready and that the Hard Disk controller is free to transfer the data to/from the disk.

WDRESETn: This bit goes directly to the "reset" of the Hard Disk controller. It should be a "1" for normal operation. This bit powers up as a "0". Strobing this bit also clears the WRITE FAULT ERROR bit of I/O TWO. It is important to note that this bit must remain low for 4.8us to meet WD1010-05 specifications.

I/O TWO INPUT PORT:

WRTFLTERR	ECCERR	FDKCHn	FDCINT	HPIB3	HPIB2	HPIB1	HPIB0
D7	D6	D5	D4	D3	D2	D1	D0

HPIB0-HPIB3: These 4 bits are the HPIB address bits. They are asserted high. When in the "SELF TEST" mode the HPIB switch selects various self tests.

FCTINT: This input is generated by the WD2793. When set it signifies that the chip has completed a command.

WRTPROTn: When this bit is a "0" the floppy disk has been write protected.

FDKCHn: This bit being a "0" means that the floppy disk was changed by the user.

ECCERR: This bit is set when after shifting all of the data and ECC bytes through, the ECC chip detects an error.

WRTFLTERR: Whenever the Hard Disk Drive detects a Write Fault, this bit will be set. It will remain set until the processor clears it by writing to D7 in the I/O ONE register.

I/O TWO OUTPUT REGISTER:

RECOVER	FDHDSELn	FDMOTONn	FDHDLDn	DKCHRESn	CLRINDn	nu	FDCRESn
D7	D6	D5	D4	D3	D2	D1	D0

FDCRESN: This output goes solely to the RESETn of the WD2793. This bit must be held low for at least 50 microseconds in order to reset the WD2793 floppy formatter controller chip.

CLRINDn: Resetting this bit to a "0" clears the FDCINDINT signal (see zone F8). When a Floppy Index pulse occurs FDCINDINT is set which creates an interrupt. The interrupt routine should strobe this bit to clear that interrupt. It is important that this bit be pulled low then set again to a "1". If it is left low, Index Pulses coming in will not cause FDCINDINT to be set.

DKCHRESn: This bit must be taken low then back high to clear D5 of I/O TWO (FLOPPY DISK CHANGED signal).

FDHDLDn: When this bit is set to a "0" the floppy disk head is loaded against the media.

FDMOTONn: When this bit is set to a "0" the floppy disk motor is turned on.

FDHDSEL: For double sided floppy drives, this bit selects either one side or the other.

RECOVER: This bit is not currently used but it is for future en- hancements. Seagate Drives will soon be coming out with the ability to be micro-stepped for recovering bad tracks of data. This recovery bit will be used when the drives are capable of doing this.

I/O THREE INPUT PORT:

TEST	SELTST	FASTDON	NU	VOL3	VOL2	VOL1	VOL0
D7	D6	D5	D4	D3	D2	D1	D0

VOL0-VOL3: These bits are switch selectable by the user. They are used to select volume configurations for the Hard Disk.

nu: Not currently used.

FASTDON: This input signals when the TTL dma machine has transferred all of its data in "fast" mode. For instance, if 256 byte blocks were being transferred this signal will go high as the BRAM counter rolls over to 257 (A8 of the BRAM address counter).

SELTEST: This input comes directly from a jumper. When in the SELF TEST mode this jumper is used in conjunction with the HPIB switch to allow the user to select up to 20 different self tests. The HPIB switch is a 10 position switch and this jumper controlled input allows recognition of another bank of 10 tests.

TESTn: This input comes directly from a jumper. When the jumper is put in the SELF TEST position, this input will be low.

I/O THREE OUTPUT REGISTER:

DMA	HPB/FDCn	DMAR/Wn	FAST	LONGn	INTENBLn	INTB	INTA
D7	D6	D5	D4	D3	D2	D1	D0

INTA,INTB: These two bits are used to select which interrupts get through to the FIRQn and IRQn inputs of the microprocessor. The table below shows the multiplexing:

INTB	INTA	FIRQn	IRQn
0	0	HPIB INTRQ	HPIB DRQ
0	1	FLOPPY INDEX	DISABLED
1	0	FDC or HPIB INTRQ	SYNC
1	1	FDC INTRQ	FDC DRQ

INTENBLn: When this bit is a "0" the interrupts are enabled as selected above. When this bit is a "1" the FIRQn and IRQn inputs of the processor are held high.

LONGn: When this bit is a "0" the ECC chip is disabled from writing its ECC bytes at the end of the data field. This allows us to test the ECC chip by writing out erroneous ECC bytes. When these are read back in the ECC chip should detect and correct the errors.

FAST: When this bit is set, the TTL DMA machine transfers data using the "fast" mode of operation. The fast mode does not require consecutive SYNC instructions as does the regular TTL DMA transfer mode. Once the fast bit is set DMA will continue until "FASTDON" (D5 of IO3 INPUT) is a "1".

DMAR/Wn: When this bit is set to a "1" DMA transfers take place in the direction indicated below:

```
DATA READ FROM FLOPPY   AND WRITTEN TO BUFFER RAM
                     or
DATA READ FROM HPIB     AND WRITTEN TO BUFFER RAM
```

When this bit is a "0" the transfers go in the opposite direction as that listed above.

HPB/FDCn: When DMA transfers take place, the data passes between either the HPIB controller or the Floppy disk controller and BUFFER RAM. When this bit is a "1" the HPIB is selected. When it is a "0" the Floppy is selected.

DMA: Setting this bit results in two things. Firstly, the DMA machine is enabled. This means that if either the HPIB or the FLOPPY data request (DRQ) lines are high, one transfer will be done. Secondly, it disables the microprocessor from controlling the R/W lines to those devices on the isolated data bus. So, if this bit is set the processor cannot read or write to any devices on the isolated data bus. When this bit is cleared to a "0", the microprocessor can read or write to the devices on the isolated data bus (providing the WD1010 is not transferring data) and the TTL DMA machine is disabled.

6 - HPIB INTERFACE AND CONTROLLER

The HPIB interface and controller circuitry is found in zones A1-3 and B1-3 of the schematic. The HPIB controller is the 8291-a. The HPIB controller is configured in the standard way. We do use the DMA mode of the chip in transferring data to/from the BUFFER RAM under control of the TTL DMA machine. It should be remembered that it is on the ISOLATED data bus and therefore, cannot be accessed while the bus is split.

7 - FLOPPY INTERFACE AND CONTROLLER

All of the floppy functions are controlled by the WD2793 controller. The interface between this chip and the actual drive is quite well specified by Sony so there is not much flexibity as far as design goes. In connecting this chip to our microprocessor side, there were some things we had to take care of. First of all, the floppy controller must be able to be read by either the processor or the DMA machine. This required an "or"ing of chip selects. Along with that, whenever the DMA machine is accessing the chip we are always accessing the data register. This register requires A0 and A1 of the chip to be high. In zone D7 there is some external circuitry which pulls these two register select lines to "1"s whenever the DMA machine is chip selecting the chip. Aside from that circuitry the interface to the processor side is standard. The 2793 has a built in PLL and data separator. The external pots in zone D8 are used to adjust the write precompensation and read pulse width. The only external discretes needed for this chip besides these pots are for the VCO (1 capacitor) and the PUMP circuitry. In adjusting the VCO cap and the two pots, the "TEST" pin is grounded. The "test" pin has an internal pull up thus eliminating the need for an external one. The WD2793 does have some stringent timing requirements which required stretching the processor read and write cycles. This will be discussed under the MEMORY STRETCHING section.

8 - HARD DISK CONTROLLER AND ECC

The hard disk controller and ECC chip are found in zones A5-6 and B5-6. The hard disk controller accepts commands from the microprocessor and takes care of all interaction with the hard disk drive. The WD1010 acts as a controller of the ISOLATED data bus when data transfers are taking place between the Hard Disk and Buffer Ram. It generates RD/WR strobes and chip selects the BUFFER RAM. Each read or write of the BUFFER RAM increments the address counter. It is important that neither the DMA machine nor the microprocessor try to access the BUFFER RAM during times that the WD1010 is in control.

The ECC chip is connected directly to the hard disk controller's data bus. As data is being written to the disk the ECC chip generates 4 bytes of ECC information. As the address counter rolls over past 256, the ECC chip is selected and these 4 bytes are written at the end of the data field. In reading the data back, these 4 bytes are also shifted through the polynomial and if the result is not zero an ECCERR signal is generated. Resetting the BUFFER RAM address counter also presets the ECC chip. The "LONGn" signal going into the ECC chip can be cleared to a "0" in I/O THREE. This will prevent the hardware from chip selecting the ECC after the address counter goes beyond 256. This feature can be used to write erroneous ECC bytes to test the functionality of the ECC detection and correction.

9 - HARD DISK READ CIRCUITRY

The hard disk read circuitry is found in zones A8-9 and B8-9. The data coming from the disk enters a differential receiver. The output of this receiver goes directly to a one shot adjusted for 50ns. This one shot's purpose is to shape the mfm pulses to ensure they are wide enough. The output of this first one shot feeds a second one shot. It also feeds the data and clock pulses to the phase lock loop. The second one shot is adjusted for just greater than one bit period (250ns). This one shot is necessary to generate the "DRUN" input to the WD1010. Continuous pulses at the bit rate (200ns) will keep re-firing this one shot signifying that perhaps the head is over the sync field.

The phase lock loop is all found within the DP8460-4 integrated chip. This chip accepts mfm data and clock pulses and generates the mfm data synchronized with "READ CLOCK". The phase lock loop needs only a few external discrete components.

10 - HARD DISK WRITE CIRCUITRY

The hard disk write circuitry is found in zones B7 and B8. The function of this circuitry is to provide write precompensation. The Hard Disk controller outputs mfm data and two signals called "EARLY" and "LATE". The delays in the mfm data are provided by a delay line with taps 12ns apart. Should it be desired to operate the controller without engaging write precompensation, it can be done simply by setting bit D5 of I/O ONE (DISWRPRE). If this bit is not set then write precompensation will be done using 12ns delays. After write precompensation is done the mfm data is fed into a differential driver. The output of this driver goes directly to the disk drive.

11 - BUFFER RAM AND ADDRESS COUNTER

The BUFFER RAM and ADDRESS COUNTER are found in zones A3-4 and B3-4. The buffer ram is a 1K X 8 static ram. This combination acts as a FIFO type arrangement. Whenever a byte is written or read from the ram, the address counter is incremented after the transfer. This means that to function correctly all of the data must be written into the BUFFER RAM, then the address must be cleared and all of the data read out. This BUFFER RAM can be accessed by three different devices: The DMA machine, the MICROPROCESSOR, and the WD1010 hard disk controller. Whenever one of these devices is accessing the BUFFER RAM the others must be disabled from accessing it. It must also be remembered that the address to the ram is not preloadable. It can only be cleared, then incremented after each read or write.

12 - DMA LOGIC

The DMA machine is found in zone C4 of the schematic. There are two modes that DMA can be used. I will call one the "FAST" mode and the other the "SYNC" mode.

FAST MODE:

Using this mode data can be transferred at up to 800K bytes per second. To select this mode of DMA, bit D4 of IO THREE OUTPUT PORT must be a high. The limitation with this mode is that only blocks of even multiples of 256 bytes can be transferred. An explanation of how this mode works should make it apparent why only modulo 256 byte blocks can be transferred.

This mode of DMA does not need the microprocessor at all to function. Once the DMA bit is set the DMA machine is enabled and ready to go. Whenever a DRQ is asserted from either the FLOPPY or the HPIB the shift register in zone C4 will begin shifting through a "1". This sequence will generate a DMA chip select strobe as well as the proper READ or WRITE depending on the direction selected. While these signals are being generated the DMA machine is prevented from being retriggered should a second DRQ come through quickly. After each byte is completely read (or written), the DMA machine is re-enabled. This sequence continues, each time incrementing the buffer ram address counter. The only thing that will stop the transfers is when the address counter reaches 256. At this point the DMA machine is disabled and the processor must intervene. The processor knows that all 256 bytes have been transferred by monitoring D5 of I/O THREE. When this bit goes to a "1", all 256 have been transferred and DMA has been automatically disabled. The speed of this "fast" mode is totally dependent on the speed of the DRQ's coming in and the rate at which the hardware can generate the R/W and CS strobes. (This being 800K bytes per second).

SYNC MODE:

Sync mode is used whenever it is necessary to transfer a non-modulo 256 number of bytes between BUFFER RAM and HPIB or FLOPPY. The speed of this DMA mode is dependent upon the processor and is 333K bytes per second. This mode of DMA is performed by executing consecutive "SYNC" instructions. When a "SYNC" instruction is executed, the microprocessor gives up its control and waits for an interrupt. When an interrupt comes the next instruction is fetched and executed. This mode of DMA allows counting each byte transferred since one byte will be transferred for each SYNC instruction executed.

Hardware wise, the way this mode works is as follows. After the microprocessor executes a SYNC instruction it generates a signal called Bus Acknowledge (BA). This signal is used to enable the DMA machine. When a DRQ comes and this signal is asserted, a DMA transfer occurs using the same shift register sequence as explained above. While the transfer is occurring the DMA machine is prevented from being retriggerred by successive DRQ's. As the transfer is finishing the DMA machine asserts an interrupt which tells the processor to take over the bus again and execute the next instruction. This mode of DMA can transfer any number of bytes. The firmware programmer needs only to execute one "sync" instruction per byte.

13 - MEMORY STRETCH CIRCUITRY

Due to some of the chips on the isolated data bus having stringent timing requirements, it was necessary to have some circuitry which stretched the read/write cycles when accessing these devices. Rather than add the additional logic to only stretch when accessing those particular devices, it was decided to stretch the cycle whenever accessing any device on the isolated data bus. The stretch circuitry is found in zones D4 and D5. It is quite simple in operation. Whenever any address lower than 0100H is written to (or read from) the MRDY line of the microprocessor is pulled low. The microprocessor will hold in its present state until the MRDY line is taken back high. A shift register clocked by the 8Mhz signal determines how long the cycle is stretched. Besides just controlling MRDY, the circuitry generates an "enable write" signal in the middle of the stretch cycle. This write signal goes away and MRDY is taken back high. By doing this we could meet the stringent set-up and hold times required by the Floppy. During DMA transfers it is not necessary to stretch any cycles because the DMA machine completely generates the chip selects and read/write strobes; it is designed to meet the particular specifications.

ROADRUNNER III PERFORMANCE ISSUES

BS 6/29 (ER3C)

This portion of documentation is meant to explain some of the performance issues related to RR3. In discussing various performance criteria, it should be understood that the performance varies depending on several things. The speed of the host driving the RR3 is a dominating factor. Things such as interleave, stagger, block size, access times, etc. all become factors which directly affect the performance. The intent of this report is to discuss some of these issues so that maximum performance can be obtained knowing the various parameters and conditions of a given system configuration.

The issues will be presented in the following order:

I. INITIALIZATION
II. DRIVE PERFORMANCE
III. TRANSFER RATES
IV. MAXIMIZING PERFORMANCE

I. INITIALIZATION

Probably the first operation a user will see his new RR3 perform will be that of initializing the hard disk. Users ignorant of what is actually going on during this initialization are often impatient wishing that the initialization procedure was quicker. I observed the following initialization times:

FLOPPY DISK (710 KBYTES) - - - - - 1 minute 23 seconds.
HARD DISK (15 MBYTES) - - - - - - 23 minutes 2 seconds.

(These times were observed using the BASIC 3.0 "INITIALIZE" command on a 9816.)

The question arises, "why does it take so long?". The best answer to this question is to explain what is being done and the algorithm for doing it.

The purpose of the initialize is twofold. Firstly, it is to find out which sectors on the disk are bad (i.e. bad media, or damaged during shipping). When a bad sector is found, it is spared out and never used. Secondly it is to write a specific format on the disk. This involves writing headers, gaps, and a fixed data pattern. Writing this information on the disk involves interleaving, stagger- ing and sparing of bad spots.

The task of determining bad areas on the disk is the task which takes the majority of the initialize time. But it is critical that the bad areas be found and spared. The sparing algorithm works as follows:

Each sector of the entire disk is written and read 5 times. This includes both header and data information. The writing and reading is done by approaching the track from both sides (three times from the outside and twice from the inside). Should any of the reads or writes detect any errors (either hard or soft), the erroneous sector will be entered into a spares table. During the final format of each track, the bad sectors from the spares table will be spared out and that specific area of the disk will never be used. Up to 84 sectors can be spared out on a 15 Mbyte disk.

II. DRIVE PERFORMANCE

The overall system performance of RR3 is mainly dependent on the inherent characteristics of the drives we use. The 15 Mbyte hard disk used is manufactured by SEAGATE. Its speed characteristics are as follows:

DATA TRANSFER RATE - - - - - -	5 Mbits per second (625 Kbytes per second)
AVERAGE ACCESS TIME - - - - -	85 ms
AVERAGE ROT. LATENCY - - - - -	8.33 ms
ROTATIONAL SPEED - - - - - - -	3600 rpm (16.66 ms per rotation)

The floppy disk used is 710 Kbytes and is made by SONY. Its speed characteristics are:

DATA TRANSFER RATE - - - - - -	500 Kbits per second (62.5 Kbytes per second)
AVERAGE ACCESS TIME - - - - -	365 ms
AVERAGE ROT. LATENCY - - - - -	50 ms
ROTATIONAL SPEED - - - - - - -	600 rpm (100 ms per rotation)

III. TRANSFER RATES

In talking of the performance of a mass storage peripheral, the question is often asked, "How fast can it transfer data?". This question is somewhat ambiguous in that how fast a system CAN transfer data is not necessarily how fast it DOES. Each data transfer depends on several factors like: are the sectors of data contiguous?, is the interleave matched for the controlling host's capability?,etc. In executing a typical high level mass storage command such as a "READ" several interactions must occur. I have chosen to break down all of these interactions into time blocks as follows:

SOFTWARE OVERHEAD - This block represents all of the time that the CPU on board the RR3 controller is busy receiving HPIB commands, decoding them, issuing lower level commands to the VLSI chips, monitoring action, reading status, decoding error messages, and reporting via HPIB to the host computer. The time that this block takes for an arbitrary command is very dependent on what things happen during the command. In looking at various commands and timing a conservative "nominal" value is 1.5 ms.

POSITIONING HEAD - This block of time is the time between an actual command given to the drive and the time that the head is positioned over the desired area on the media. Speaking in averages this time would be the AVERAGE ACCESS TIME plus the AVERAGE ROTATION- AL LATENCY. For the SEAGATE drive these times add up to 93.3 ms. For the SONY the average head positioning time is 415 ms.

DISK DATA TRANSFERS - This time is the time it takes to actually transfer the data to/from the drive media from/to the buffer ram on the controller board. For the SEAGATE drive a 256 byte sector can be transferred in 409.6 us (5 Mbits per second). For the SONY it takes 4.096 ms.

HPIB DATA TRANSFERS - This is the time it takes to transfer data between buffer ram and the host computer. The RR3 controller uses two modes of DMA transfers via HPIB. One mode (used in transferring floppy data between HPIB and the host) transfers data at up to 333.3 Kbytes per second. The other mode (used for hard disk transfers) transfers data at up to 800 Kbytes per second. These modes can be used at any time for either disk. Direct leveraging of floppy code from SPARROW necessitated using the slower mode for floppy data. Thus, a 256 byte block of data can be transferred in 320 or 768 us depending on the mode used.

To get a feel of the time breakdown for a command, consider the case of a "READ" of a 256 byte sector from the hard disk:

SOFTWARE OVERHEAD	1.5 ms
POSITIONING HEAD	93.3 ms
DISK DATA TRANSFER	409.6 us
HPIB DATA TRANSFER	320.0 us
TOTAL	95.530 ms

This data shows how significant head positioning time is compared to the other times. In this case it represents nearly 98% of the time. Of course, this percentage decreases as more sectors are read per track access, but it is still very high. Many people talk of getting higher HPIB data rates to improve performance. This data points to that as being very minor. The way to make significant improvements in performance is by decreasing the disk access time.

IV. MAXIMIZING PERFORMANCE

The preceeding paragraphs have attempted to give some background on performance directly related to the RR3 controller. There is still one criteria that directly affects system performance that was not discussed. Though it is feasible that the DMA hardware can operate at 800 Kbytes per second, this condition can only exist if the host computer is fast enough to handshake the data at that rate. System performance can be greatly improved by formatting at an inter- leave appropriate for the host computer being used. Suppose for example that the hard disk was formatted at an interleave of 3. This means that an entire track could be read in 3 revolutions of the disk. The average of these transfers would be 163.9 Kbytes per second. To match this interleave the host computer would have to transfer data at 275 Kbytes per second or faster. If the host transferred less than 275 Kbytes per second it would take 32 revolutions to read the track (one revolution per sector). This would average 15.37 Kbytes per second. Thus, having the wrong interleave for the transfer rate of the host could mean a system that operates 10 times slower than it could if the right interleave were chosen. The following table shows ideal interleave values for selected systems:

HOST COMPUTER CONFIGURATION	MAX BURST TRANSFER RATE	AVERAGE RATE (MULTI TRACK)	HARD DISK INTERLEAVE	FLOPPY INTERLEAVE
SERIES 200	129K	65K	7	2 - ALONE
SERIES 200	304K	108K	4	2 - WITH DMA (98620A)
SERIES 200 WITH SIMON (98625A)	748K	145K	3	2* - WITH DMA (98620A) -
HP150	NO DATA	47K	10	2

*Data transfer rate is limited by floppy DMA mode on controller not on speed of host computer. Otherwise I/L of 1 would be possible with DMA and SIMON card.

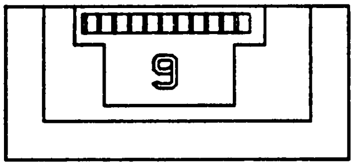
It should be mentioned that IEEE standard 488 bus specifications specify certain handshake timing requirements. When all of these timing requirements are added up using the slowest times permissible the

HPIB transfer rate is around 333 Kbytes per second. Thus, if a user had a 200 series computer with a DMA and SIMON card, the transfer rate could be limited by these worst case handshaking specs. So, the ideal interleave for a very fast system (very fast is anything over 333 Kbytes per second) may best be found by empirical methods.

```
*
*            CONFIGURATION JUMPERS   for REV C board
*
*     JUMPERS ON    BINARY   DEC   MEGABYTES   FLOPPY    CONFIG
*
*     ----------    ----     --    ---------   ------    ------
*     HGFE (EFGH)   0000     0        15        NO       M15
*     HGFA (AFGH)   0001     1        20        NO       M20
*     HGBE (EBGH)   0010     2        40        NO       M40
*     HGBA (ABGH)   0011     3        15        NO       PSEUDO_15        WD1010
*    [✓ HCFE (EFCH)] 0100    4        15        YES     [✓ M15_F] —FOR 15MB DRIVE & FLOPPY
*     HCFA (AFCH)   0101     5        20        YES      M20_F
*     HCBE (EBCH)   0110     6        40        YES      M40_F
*     HCBA (ABCH)   0111     7        15        YES      PSEUDO_15_F
*
*     DGFE (EFGD)   1000     8        15        NO       M15
*     DGFA (AFGD)   1001     9        20        NO       M20
*     DGBE (EBGD)   1010     10       40        NO       M40
*     DGBA (ABGD)   1011     11       15        NO       PSEUDO_15        WD2010
*     DCFE (EFCD)   1100     12       15        YES      M15_F
*     DCFA (AFCD)   1101     13       20        YES      M20_F
*     DCBE (EBCD)   1110     14       40        YES      M40_F
*     DCBA (ABCD)   1111     15       15        YES      PSEUDO_15_F
*
```

SWITCHES

ACCESSIBLE THROUGH BACK PLANE

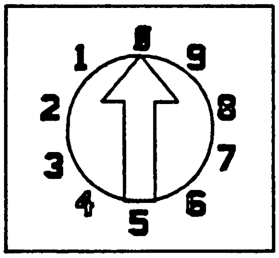


HPIB

Ø-7	HPIB ADDRESSES
8,9	HP15Ø BOOT

WHEN SELF TEST JUMPERED

Ø-9	SELF TESTS



VOLUME CONFIG

Ø	ONE 14.84 MBYTE VOLUME
1	ONE 14.84 MBYTE VOLUME
2	TWO 7.37 MBYTE VOLUMES
3	THREE 4.91 MBYTE VOLUMES
4	FOUR 3.64 MBYTE VOLUMES
5	ONE 12.29 MBYTE VOLUME ONE 2.51 MBYTE VOLUME
6	SIX 2.41 MBYTE VOLUMES
7	ONE 9.83 MBYTE VOLUME TWO 2.46 MBYTE VOLUMES
8	EIGHT 1.77 MBYTE VOLUMES
9	ONE 7.32 MBYTE VOLUME THREE 2.46 MBYTE VOLUMES

HEWLETT PACKARD

FILE A9133089

NOTE: This page provides a running history of changes for a multi-page drawing which cannot conveniently be re-issued completely after each change. When making a change, list for each page all before-and-after numbers (within reason; use judgement, and use 'extensive' revision notes if loss of past history is tolerable, or retype complete page) and associate with each a symbol made up of the change letter and a serial subscript to appear here and on the page involved (there enclosed in a circle, triangle, or other attention-getting outline).

LTR	REVISIONS	DATE	INITIALS
B	REVISED AND RETYPED PER PCO 58-2359	02/13/86	

Model No. 09133-66510	Date JULY 24, 1985
Title MANUAL BOARD ADJUSTMENTS PROCEDURE	
By RAY FRANKLIN	Sheet No. 1 of 3
Supersedes	Drawing No. A-09133-66510-13

<B> 09133-66510 MANUAL BOARD ADJUSTMENTS PROCEDURE

EQUIPMENT NEEDED:

1 ET-17264 Functional test fixture with 2-minute timers installed.
1 Function Generator
1 5335A Counter
1 10041A 10:1 divider probe
1 BNC cable
1 Ribbon cable connector (3M #3421-6020, 20-pin)

GENERAL

Make function generator cable by soldering BNC center lead to Pin 17 of 20-pin connector. Solder shield to Pin 18. If the adjustments are made without the functional test fixture then only J6 and the power connector must be attached to the board. Set the function generator
<B> for a 2 MHZ TTL square-wave. Set the counter to 'COM A' and 'SLOPE B'
<B> negative; this is equivalent to 'PULSE A'.

HARD DISK "DRUN" ADJUSTMENT

1. Plug in all power connectors

2. Plug Function Generator into J6.

3. Turn on power.

4. Start 2-minute timer if not using the hardware timer in the fixture. Press K2 if using functional test station (see A-09133-66510-15 for further instructions).

5. Set Counter to "TIME A -> B".

6. Probe "DRUN" (TP5) test hole.

7. Adjust "DRUN" pot (R10) so that counter reads 250 nSEC +- 5 nSEC.

8. Wait until 2-minutes are up before going on to the next adjustment.

<B> 09133-66510 MANUAL BOARD ADJUSMENT PROCEDURE CONT'D

FLOPPY PHASE LOCK LOOP ADJUSTMENTS:

1. Move JMPO jumper from "FRONT" to "BACK" (HPIB connector is at the "BACK").
2. Set counter to "FREQ A".
3. Probe "VCO" (TP2) test hole.
4. Adjust "VCO" cap (C8) so that counter reads 500 KHZ +- 2 KHZ.
5. Set counter to "TIME A -> B".
6. Probe "RPW" (TP3) test hole.
7. Adjust "RPW" pot (R4) so that counter reads 250 nSEC +- 5 nSEC.
8. Probe "WPW" (TP4) test hole.
9. Adjust "WPW" pot (R2) so that counter reads 125 nSEC +- 5 nSEC.
10. Turn off power and move JMPO jumper back to "FRONT" position (HPIB connector is at "BACK").
11. Disconnect function generator from J6.
12. Follow Functional Test Procedure (A-09133-66510-15) if required.