

## SPECTRA LOGIC DISK & TAPE EMULATOR

The Spectra Logic Disk & Tape Emulator is essentially two boards in one. The Disk Emulator is separate from the Tape Emulator with the single exception that they share a 20MHz oscillator.

This manual is therefore split into 2 parts

Part A Disk Emulator  
Part B Tape Emulator

## PART A DISK EMULATOR

The Disk Emulator is designed to use dynamic memory to emulate an SMD disk drive. It is intended to be used in Manufacturing alleviating the need to purchase expensive high performance disk drives.

The design includes 16 memory ICs, which may be either 64k or 256k giving a simulated disk capacity of 128k or 512k bytes. This memory is divided up to emulate 2 heads and either 2 or 8 cylinders with a track capacity of 32k bytes. A rotating disk is emulated by using a free running 14 bit counter to determine the track position. When this counter is zero the Index signal is generated. Other values will generate hard Sector pulses. This 14 bit counter together with the cylinder and head addresses are used to determine which word of the dynamic memory to address.

The Disk Emulator has 2 oscillators installed to allow it to operate at 2 different data transfer speeds. The first version of the emulator has a 20MHz and a 25MHz oscillator installed to allow the drive to transfer data at 2.5 MBs or 3.125 MBs respectively.

## SWITCHES

There are 11 switches provided by the disk emulator as follows:

### Unit Address 0, Unit Address 1, Unit Address 2

These 3 switches allow the drive address to be in the range 0-7

### On-line, Write Protect

These 2 switches allow the user to set the status of the drive

### Sector Switch 1, Sector Switch 2

These 2 switches address a PROM which generates the index and sector pulses. The switches allow one of 4 different hard sector sizes to be selected. The values depend on the contents of the PROM, but the initial version of this PROM give the following results

Sector Switch 1	Sector Switch 2	
OFF	OFF	32 sectors per track <i>(512 bytes/sector)</i>
OFF	ON	64 sectors per track <i>(512 bytes/sector)</i>
ON	OFF	35 sectors per track (last sector long)
ON	ON	Not Used <i>(493 bytes)</i>

### 3 MBs Switch

This switch is used to select the data transfer speed.

### Dual Port

Although the Disk Emulator is not capable of being attached to 2 different controllers, a dual port function may be simulated by setting this switch. In this case whenever the controller releases and de-selects the disk emulator, it will go "busy" for 10ms. During this time the disk emulator will not respond to any further controller commands.

### Correctable ECC Error

Setting this switch inhibits one bit of one 16 bit word in the second sector of all tracks with odd cylinder addresses. This simulates a single bit defect.

### Uncorrectable ECC Error

Setting this switch inhibits one bit of two consecutive 16 bit words in the second sector of all tracks with odd cylinder addresses. This simulates a defect of 17 bits.

### Reset

This switch resets the disk emulator.

## LEDS

There are 8 LEDs on the disk emulator as follows

### Unit Selected, Unit Ready

These display the status of the disk emulator.

### Off Cylinder

Although there is no head to move, the disk emulator will return a status of "off cylinder" for a period of 3 MS following a seek, recalibrate or offset command.

### Fault

The disk emulator will return a fault condition whenever

- a) More than one TAG line has been asserted
- b) Read and write gate have been asserted simultaneously
- c) A head address greater than 1 has been selected
- d) Read gate has been asserted and the drive is "off cylinder"
- e) Write gate has been asserted and the drive is "off cylinder"
- f) Write gate has been asserted and the drive is write protected
- g) Write gate and offset have been asserted simultaneously
- h) An illegal cylinder address where more than one bit is set has been selected. This enables software to detect when two high order bus bits are shorted together since the high order bits are not normally tested due to the disk emulator only having a maximum of 8 cylinders.

### Seek Error

Seek error is set whenever an illegal cylinder address is selected.

### Read Gate, Write Gate

These LEDs are lit whenever the respective commands are asserted by the controller.

## THEORY OF OPERATION

The Disk Emulator is drawn on the first 5 sheets of logic.

### Sheet 1

This page shows the SMD connectors, the SMD terminating resistors, the switch connector, the LEDs and their drivers, the power supply connector and the decoupling capacitors.

J1 attaches the SMD A cable and J2 the SMD B cable. RP1, RP2, RP3, and RP4 are the terminating resistors for the A cable. These resistor packs are normally installed in sockets so that they may be removed for daisy chain drives. RP5 has the terminating resistors for the B cable and is normally soldered directly into the board.

J8 is the connector for switches. This connector enables the switches to be mounted on the front panel and connected to the PCB. The LS244 at 7T is the driver for LEDs DS1 thru DS8. These LEDs are normally installed on the front panel and connected to the PCB via a cable. The LEDs should have current limiting resistors.

### Sheet 2

This page shows the SMD drivers and receivers. An ECL driver and an ECL receiver are used on part of the B cable interface to allow reliable operation at higher data transfer rates.

### Sheet 3

This page contains the SMD interface control. The PROM and two PALs at the top of the page are used to decode the SMD bus and TAG lines. The PROM at 2X decodes the most significant 9 bits of the SMD bus to determine illegal cylinder or head addresses. The PAL at 2W decodes the Read and Write gates and the majority of fault conditions. The PAL at 2V decodes a Read or Write Gate status and also decodes a seek, offset, or recalibrate command to fire the one-shot at 3BB taking the emulator "off cylinder". This PAL at 2V also holds the current cylinder and head addresses of the Disk Emulator. These addresses are latched by using the feedback capability of the PAL. These addresses and the signals R+WGATE and WGATE are then synchronised by the 74LS174 in 2U so that they do not change in the middle of a dynamic memory cycle.

The 74S85 at 3X in the bottom left hand side of the page is used to compare the disk address being selected by the controller to the address set on the switches. If they compare the flip/flop at 3AA is set, selecting the unit when Unit Select Tag is asserted. This flip/flop remains set until Unit Select Tag is de-asserted. Providing the Pick and Hold lines are asserted, the Off-line switch is not set, and there is no Busy or Fault condition, Unit Ready is returned to the controller and the TAGEN signal is asserted by the gates of the LS00 IC at 3Z on the right hand side of the page. The TAGEN signal enables the tri-state TAG line receivers which are otherwise held inactive by pulling them high.

The 10 MS one-shot in the middle of the page is used to simulate dual port operation. It is held inactive if the dual port switch is not on. If the switch is on the latch made up of 2 LS02 gates may trigger the one-shot. This latch is set whenever the controller sends a release command. When the unit is subsequently de-selected the latch is reset firing the one-shot. If the unit is then selected again before 10 MS has expired, Busy is asserted, and Unit Ready and TAGEN are inhibited. This will prevent the Disk Emulator responding to any controller TAG lines.

The 3 MS one-shot on the right hand side of the page is used to simulate a moving head. It is fired whenever a seek, recalibrate, or offset command is decoded by the PAL at 2V.

#### Sheet 4

This sheet contains the dynamic memory address and the timing controls. The 2 basic oscillators are shown at the bottom left hand side of the page at 5R and 3R. The 3 MBs switch decides which oscillator is selected to be the source of the Servo Clock signal which is returned to the controller. The controller will turn this signal around and send it back as Write Clock for write commands. Either the Servo Clock or the Write Clock is then selected to be the basic Bit Clock to be used to generate the timing for the Disk Emulator. The selection between the 2 clocks is performed by the 2 74F174 ICs and some associated gates at the bottom left hand side of the page. This design ensures a smooth crossover of the selection such that no very short glitches are produced on the Bit Clock line that may upset the operation of other parts of the logic.

The Bit Clock is then used to clock the 74AS163 counter at 3V. This 4 bit counter is used to divide the Bit Clock by 16 and generate timing signals for a 16 bit word. The Row and Column Address strobes, RAS and CAS, for the dynamic memory array are derived from this counter on the bottom right hand side of the page. The Q3 output of the counter is used to clock the Word Counter on the top left hand side of the page and also select the Row or Column Address via the 2 74LS257 ICs and the 74LS253 IC at the top of the page.

The Word Counter is a free running 14 bit counter that is used to address 16k words of the dynamic memory array. This simulates a rotating disk with a track capacity of 32k bytes. The Word Counter and the current Cylinder and Head addresses are then multiplexed to generate the Row and Column address for the dynamic memory. The least significant bits of the Word Counter are used to generate the Row address and since it is a free running counter no independent refresh cycles are required of the dynamic memory. The Word Counter is also decoded via the PROM at 6Z and some gates on the right hand side to produce the index and sector pulses and the INHDAT signal. The Index and Sector pulses are 2 words or 4 bytes wide. The INHDAT signal is used to inhibit data from being written into one of the dynamic memory ICs. It is present for one word if the Correctable ECC Error switch is set and for two words if the Uncorrectable ECC Error switch is set.

## Sheet 5

This sheet contains the dynamic memory array itself. There are 16 ICs drawn on the right hand side representing a 64k x 16 bit or a 256k x 16 bit memory array. Two bidirectional register ICs, 2952s, are used to buffer the memory array from the 16 bit serialiser/deserialiser made up of 2 74F299 ICs.

When writing to the Disk Emulator the serial Write Data is fed into the 2 74F299s and deserialised. When 16 bits have been shifted in the word is transferred to the 2952s. While the next 16 bits are being accumulated in the deserialiser, the first word is written to the dynamic memory.

When reading from the Disk Emulator a word is read from the dynamic memory every 16 bit clocks. This word is clocked into the 2952s and then into the 74F299s in parallel. The word is then serialised while the next word from memory is being read. The serial data is shifted through and out of the 74F299s as Read Data (RDDATA).

## PART B TAPE EMULATOR

The Tape Emulator is designed to use dynamic memory to emulate a magnetic tape and drive. It is intended to be used in Manufacturing alleviating the need to purchase and maintain tape drives.

The design includes twelve 64K memory ICs, which are used to emulate a very short tape 64K bytes long. 9 of the data bits are used to for data and parity, and the other three bits are used for file marks, end of block marks and overall parity.

This is a bit-slice 2901 design and has many similarities to other Spectra Logic designs. The firmware is written into five 1K PROMs giving a code space of 1K x 40 bits. It is clocked by the 20 MHz oscillator shared with the Disk Emulator.

## SWITCHES

There are 16 switches provided by the tape emulator as follows:

### Formatter Address, Transport Address 0, Transport Address 1

These 3 switches allow the tape drive address to be in the range 0-7

### On-line, Off-line, Write Protect

This switch control the status of the tape drive.

### Rewind

This switch causes the firmware to logically rewind the tape.

### High Density, Low Density, Remote Density

The three position switch sets the logical density of the tape drive.

### Internal Parity

This switch causes the tape drive to generate its own data parity when writing to tape.

### Hard Error, Correctable Error, Read Parity Error

These switches force error status or inhibit read data parity on the interface so that error status may be checked in the adaptor.

### Disable EOT

This switch disables the End of Tape status. The memory address will continue to be incremented when writing to the tape and will wrap around when it overflows. The purpose of this switch is to allow write commands to be diagnosed without having to issue rewinds, thus increasing the intensity of an oscilloscope display.

### 100KB, 365KB, 455KB

These switches control the data transfer speed as follows :-

100KB	365KB	455KB	Transfer speed
OFF	OFF	OFF	556 KB/sec (Default)
OFF	OFF	ON	455 KB/sec
OFF	ON	X	365 KB/sec
ON	X	X	100 KB/sec

### Reset

This switch reset the tape emulator and causes the firmware to run its own internal micro-diagnostics.

## LEDS

There are 8 LEDs on the tape emulator as follows

### Diagnostic Fail

This red LED is lit when the tape emulator is reset, and is put out by the firmware when it has successfully run its internal diagnostics.

### Diagnostic Pass

This green LED is lit by the firmware when it has successfully run its internal diagnostics.

### Memory Error

This LED is lit when an overall parity error is detected in the dynamic memory.

### Tape Unit Selected

This LED indicates the tape unit is currently selected.

### On-Line

This LED indicates the tape unit is on-line.

### Write Protect

This LED indicates the write protect switch is on.

### End of Tape

This LED indicates the tape unit has reached a logical end of tape.

### Beginning of Tape

This LED indicates the tape unit is positioned at the beginning of the logical tape.

## THEORY OF OPERATION

The Tape Emulator is drawn on the last 5 sheets of logic.

### Sheet 6

This page shows the interface connectors, the interface terminating resistors, the switch connector, the LEDs and their drivers, the unit selection logic and the tape status decode PAL.

J3 and J4 are edge connectors to attach the two tape interface cables. The terminating resistors packs in 1C and 1F are normally installed in sockets so that they may be removed for daisy chain drives.

J7 is the connector for switches. This connector enables the switches to be mounted on the front panel and connected to the PCB. The LS244 at 7J is the driver for LEDs DS9 thru DS16. These LEDs are normally installed on the front panel and connected to the PCB via a cable. The LEDs should have current limiting resistors.

The tape unit address is compared by the S85 comparator in 2F. If the address selected is equal to the address on the switches the status signals are enabled to the interface. The tape status decode PAL in 2E controls the general status of the tape unit. This PAL also detects tape motion commands sent on the interface and flags the firmware by setting the FBY, Formatter Busy, or RWD, Rewinding, signals.

### Sheet 7

The write data lines sent from the host adaptor and the read data lines returned are controlled by the logic across the top of this page. The write data lines come in on the left hand side of the page and are clocked into the LS374 registers at 2B and 2C. If the internal parity switch is not set, the write parity is checked and the error flip-flop at 3A is set if it is wrong. If the internal parity switch is not set, the write parity of the 8 data lines is generated. The function of the internal parity switch is accomplished by the LS257 multiplexor at 2A. The LS374 registers hold the last word signal as well as the data lines and are read by the firmware onto the 2901 source bus. On the right hand side of the page the firmware can load the 2901 destination bus into the two LS174 ICs at 2G and 2H. The outputs of these registers are used to return the read data and file mark signals to the interface.

The LS374 register at 2D at the bottom of the page is used to receive the command lines from the host adaptor. This register also may be read by the firmware onto the 2901 source bus.

The logic on the bottom right hand side is the reset logic for the tape emulator. The diagnostic fail latch is made by cross-coupling two LS02 gates.

## Sheet 8

This sheet contains the condition code, sequencer, control store and pipeline registers for the firmware of the tape emulator. The test panel connector J6, for use with the S02 test panel is also shown here.

The next address control PAL 2K at the top left decodes the next address control and the condition codes to generate the controls for the 2911s. This method of next address control is a little different from previous Spectra Logic designs in as much as the 2911 controls are decoded in the previous cycle and clocked into the register of the PAL for use with the next cycle. This PAL also controls the flag enable signal for flag instructions.

The three 2911 ICs generate a 10 bit microcode address. The direct inputs to the multiplexor within the 2911s are not used since all jumps use the internal register, whose clock is always enabled and thus holds the jump address for the current instruction being obeyed.

The microcode itself is contained in five 1Kx8 PROMs giving a code space of 1K by 40 bits. The 40 bit instruction is decoded as follows :-

Bits 39,38	Next address control
Bits 37,36	2901 source bus select
Bits 35,34	2901 destination bus select
Bits 33-25	2901 instruction bits 8-0
Bit 24	2901 carry in to the least significant bit
Bits 23-20	2901 A & B register address
Bit 19	5N Flag inhibit bit for non-jump instructions or Most significant condition code select bit for jumps
Bits 18-16	Least significant condition code select bits for jumps
Bits 15-12	Flag data and address bits for 5P flags
Bits 11-0	5N Flag address or constant for non-jump instructions or
Bit 10	Condition code polarity and
Bits 9-0	Jump address for jump instructions

## Sheet 9

This sheet contains the constant buffer, the ALU, the flags, the clock divider, and the source and destination controls. The ALU is made from three 2901s giving a width of 12 bits.

There are two types of flags. The flags in the IC at 5P may be changed in any instruction. The flags in the IC at 5N may only be changed by non-jump or non-constant instructions since the flag address is shared with other fields.

The clock divider is made from two S113 flip-flops, dividing the 20MHz clock by four to yield a basic clock cycle of 200ns.

## Sheet 10

This sheet contains the dynamic memory, the memory controller, and the memory datapaths. The dynamic memory is drawn as twelve 256K dynamic rams but since address bit 8 is held at ground only 64K will be used. The PCB may contain either 64K or 256K dynamic rams.

The dynamic memory controller is a 4500 IC. This IC controls all the dynamic memory timing and has an internal refresh counter. It accepts a 16 bit memory address, loaded by the firmware in two parts. The most significant 4 bits are pre-loaded by the firmware into the LS174 IC at 4A and the other 12 bits are loaded directly.

The two S280 parity ICs at 5B and 6B on the middle left hand side of the page are used to both generate and check the overall parity bit of the 12 bit word. When writing to memory the data is loaded into the two LS374 ICs at 5A and 6A. The parity bit input here is grounded to ensure the two parity ICs properly generate the parity. When reading from memory the data is read into the two LS373 ICs at 5C and 6C. The parity is checked at this time and, if it is wrong, the LS74 flip-flop at 3A is set. The firmware may invert the parity written to memory by raising the PARTEST signal and so check the operation of the parity error flip-flop in micro-diagnostics.



## S03 Master Diskette

The S03 Master Diskette contains the following files : -

S03P1.SCH - S03P10.SCH

These files are the PCAD schematic files for each of the 10 pages.

S03P1.PLT - S03P10.PLT

These files are the PCAD plot files for each of the 10 pages in HP format.

S03PANEL.SCH

This file is the PCAD schematic file for the front panel.

S03PANEL.PLT

This file is the PCAD plot file for the front panel in HP format.

S03A.REW

This file contains the rework instructions for revision A of the artwork of the S03 PCB.

S03.BOM

This file is the Bill of Material file generated by running the PCNETS extract program on the 10 schematic sheets.

S03.TEL

This file is the package and net list file generated for the TELESIS system by running the PCNETS extract program on the 10 schematic sheets.

S03.MAN

This file is the manual describing the product. It has been generated using WORDSTAR and must be printed using WORDSTAR.

S03\_2EA0.PAL, S03\_2KA0.PAL, S03\_2VA0.PAL, S03\_2WA0.PAL

These files are the PALASM source files for the 4 PALs.

S03\_2EA0.PIN, S03\_2KA0.PIN, S03\_2VA0.PIN, S03\_2WA0.PIN

These files are the pin files generated by PALASM for the 4 PALs.

S03\_2EA0.JED, S03\_2KA0.JED, S03\_2VA0.JED, S03\_2WA0.JED

These files are the JEDEC files generated by PALASM for the 4 PALs. They can be downloaded directly to the DATAIO programmer to blow the fuses in the 4 PALs. If transferred to the PDP11 first, the files must be transferred using XMODEM since the files contain special control characters.

S03\_2XA0.PLE, S03\_6ZA0.PLE

These files are the PLEASM source files for the two miscellaneous PROMs.

S03\_2XA0.HEX, S03\_6ZA0.HEX

These files are the HEX format files generated by PLEASM for the two PROMs. They can be downloaded directly to the DATAIO programmer to blow the fuses in the two PROMs.

S03TA0.MAC

This file is the source for the firmware of the tape emulator. This source may be used by the meta-assembler on the PDP11.

S03TA0.LST

This file is the list file of the firmware of the tape emulator. It is generated by running the meta-assembler on the PDP11.

S03TDEF.MAC

This file is the definition source file for the tape emulator.

S03TDEF.LST

This file is the list file of the firmware definitions of the tape emulator. It is generated by running the meta-assembler on the PDP11.

S03TA0.PRM

This file is the PROM object file generated by running the meta-assembler on the PDP11. It can be used by the DATAIO program to download the PROM information for the tape emulator to the DATAIO programmer.

READ.ME

This file.

S03 Rework Revision A PCB

Deletions

1. Cut trace 3J-5 (Component side)
2. Cut trace 3J-9 (Solder side)

Additions

1. Add wire 3R/7 - 3S/7
2. Add wire 3R/14 - 4R/14
3. Add wire 5R/7 - 5S/7
4. Add wire 5R/14 - 5P/16
5. Add wire 4M/6 - 4M/1
6. Add wire 4M/7 - 2K/7
7. Add wire 1D/9 - 6M/13
8. Add wire 7J/11 - 6N/1
9. Add wire 6N/2 - 6M/12
10. Add wire 6M/11 - 3J/5
11. Add wire 4N/11 - CR4/2 (Square pad on CR4 between 4L and 4M)
12. Add wire 4N/9 - 3J/9
13. Add wire 4N/19 - 4P/10

\* Must change wld <sup>IN</sup>CA1 (DIAGS) to Run Mag Reli & ECL/N3  
MOLT

MT RELI PATCH: 1337/24243

\* Must add wire to 17+27 @

Set CONFIG switch for Emulator 1-7-ON/8 OFF  
Replace 3R to change xfer rate

(Material list created from S03.CMP and S03.WRL)

(S03.BOM created )

(Date: 12-18-1986 Time 18:43:07)

Upper Assembly

2	1100013	2KX4PRM;27S185	2X;6Z
5	1100017	1KX8PRM;27S281	3K;3L;3M;3N;3P
3	1100019	16L8;16L8	2E;2V;2W
1	1100036	16R6;16R6	2K
28	1100050	256KDRM;150NS	4V;4W;4X;4Y;5D;5E; 5F;5G;5V;5W;5X;5Y; 6D;6E;6F;6G;6V;6W; 6X;6Y;7D;7E;7F;7G; 7V;7W;7X;7Y
1	3500001	JUMPER2;.1"	W1

Lower Assembly

3	1000008	7416;7416	1G;1H;1K
1	1000018	74139;74S139	4J
2	1000032	74251;74S251	5K;6K
3	1000042	7400;74LS00	3S;3Z;4AA
2	1000043	7402;74LS02	2AA;5J
2	1000044	7404;74LS04	5H;6N
1	1000046	7408;74LS08	3J
1	1000048	7411;74LS11	3Y
1	1000050	7432;74LS32	6M
3	1000052	7474;74LS74	3A;3AA;3W
6	1000060	74174;74LS174	2G;2H;2U;4A;4K;4M
4	1000064	74240;74LS240	1B;1D;1E;7J
2	1000065	74244;74LS244	4N;7T
1	1000067	74253;74LS253	3T
3	1000068	74257;74LS257	2A;4Z;5Z
1	1000069	74273;74LS273	4L
2	1000073	74373;74LS373	5C;6C
6	1000074	74374;74LS374	2B;2C;2D;3H;5A;6A
2	1000076	74393;74LS393	5AA;6AA
5	1000082	3450;3450	1BB;1U;1W;1Y;1Z
3	1000083	3453;3453	2T;2Y;2Z
2	1000087	74259;74LS259	5N;5P
1	1000090	26S02;26S02	3BB
1	1000094	74113;74S113	5L
3	1000104	74280;74S280	1A;5B;6B
2	1000113	2952;2952	5U;6U
2	1000128	74299;74F299	5T;6T
1	1000129	7421;74H21	5BB
2	1000133	7485;74S85	2F;3X
1	1000138	7474;74F74	3U
1	1000139	7408;74F08	4S
2	1000143	7438;7438	1J;2J
2	1000147	74174;74F174	2S;4T
1	1000152	7404;74F04	4U
2	1000153	7402;74F02	4R;5S
1	1000160	74273;74F273	4P
1	1000162	74163;74AS163	3V
1	1000163	10192;10192	1P
1	1000165	10125;10125	1S
1	1000169	10124;10124	2R
1	1000173	4500;4500	3B
3	1100004	2911;2911A	2L;2M;2N
3	1100015	2901;2901	3C;3E;3F
4	1300003	DIODE;1N914	CR1;CR2;CR3;CR4
2	1400009	RES;470	R3;R4
1	1400011	RES;1K	R9

1	1400017	RES;10K	R5
2	1400019	RES;20K	R1;R2
1	1400020	RES;33K	R6
3	1500011	9RSIP;1K	RP19;RP20;RP26
12	1500013	4RSIP;470	RP10;RP11;RP12;RP13;RP14;RP15; RP16;RP17;RP18;RP25;RP8;RP9
2	1500022	9RSIP;3.3K	RP23;RP24
2	1500024	9RSIP;4.7K	RP6;RP7
5	1500039	15RDIP;56	RP1;RP2;RP3;RP4;RP5
2	1500040	14RNDIPA;220/330	1C;1F
8	1600003	SSIDELED;RED	DS10;DS11;DS14;DS15;DS3;DS4; DS5;DS6
8	1600004	SSIDELED;GREEN	DS1;DS12;DS13;DS16;DS2;DS7; DS8;DS9
1	1700003	OSC;25.00MHZ	3R
1	1700011	OSC;20.00MHZ	5R
2	1800001	CAPTANTR;1.0UF	C4;C5
2	1800005	CAPTANTR;6.8UF	C3;C7
2	1800006	CAPTANTA;10UF	C1;C2
67	1900006	CAPDIP;.01UF	C70;C74;C[71-73];C[8-69]
28	3100001	256KDRM;150NS	4V;4W;4X;4Y;5D;5E; 5F;5G;5V;5W;5X;5Y; 6D;6E;6F;6G;6V;6W; 6X;6Y;7D;7E;7F;7G; 7V;7W;7X;7Y
2	3100002	2KX4PRM;27S185	2X;6Z
4	3100003	16L8;16L8	2E;2V;2W;2K
5	3100006	1KX8PRM;27S281	3K;3L;3M;3N;3P
3	3200001	26PHDR;5002	J2;J7;J8
1	3200003	60PHDR;5202	J1
1	3200010	20PHDR;6002	J6
1	3200033	CONN6;	J5
2	3500002	JUMPER2;.1"	W1[1-2]

(Netlist created from S03.CMP and S03.WRL)

(S03.TEL created )

(Date: 12-18-1986

Time 18:42:25)

\$PACKAGES

!74LS00;	3S	3Z	4AA					
!74F02;	4R	5S						
!74LS02;	2AA	5J						
!74F04;	4U							
!74LS04;	5H	6N						
!74F08;	4S							
!74LS08;	3J							
!74LS11;	3Y							
!7416;	1G	1H	1K					
!74H21;	5BB							
!74LS32;	6M							
!7438;	1J	2J						
!74F74;	3U							
!74LS74;	3A	3AA	3W					
!74S85;	2F	3X						
!74S113;	5L							
!74S139;	4J							
!74AS163;	3V							
!74F174;	2S	4T						
!74LS174;	2G	2H	2U	4A	4K	4M		
!74LS240;	1B	1D	1E	7J				
!74LS244;	4N	7T						
!74S251;	5K	6K						
!74LS253;	3T							
!74LS257;	2A	4Z	5Z					
!74LS259;	5N	5P						
!74F273;	4P							
!74LS273;	4L							
!74S280;	1A	5B	6B					
!74F299;	5T	6T						
!74LS373;	5C	6C						
!74LS374;	2B	2C	2D	3H	5A	6A		
!74LS393;	5AA	6AA						
!2901;	3C	3E	3F					
!2911;	2L	2M	2N					
!2952;	5U	6U						
!3450;	1BB	1U	1W	1Y	1Z			
!3453;	2T	2Y	2Z					
!4500;	3B							
!10124;	2R							
!10125;	1S							
!10192;	1P							
!256KDRAM;	4V	4W	4X	4Y	5D	5E	5F	5G
!256KDRAM;	5V	5W	5X	5Y	6D	6E	6F	6G
!256KDRAM;	6V	6W	6X	6Y	7D	7E	7F	7G
!256KDRAM;	7V	7W	7X	7Y				
!60PHDR5202;		J1						
!50PECON/100;		J3	J4					
!26PHDR5002;		J2	J7	J8				
!26S02;	3BB							
!20PHDR6002;		J6						
!16L8;	2E	2V	2W					
!16R6;	2K							
!15RDIP;	RP1	RP2	RP3	RP4	RP5			
!14RNDIP;	1C	1F						
!9RSIP;	RP19	RP20	RP26					
!9RSIP;	RP23	RP24						

19RSIP;	RP6	RP7						
14RSIP;	RP10	RP11	RP12	RP13	RP14	RP15	RP16	RP17
14RSIP;	RP18	RP25	RP8	RP9				
12KX4PRM;	2X	6Z						
11KX8PRMS;	3K	3L	3M	3N	3P			
1CAPDIP;	C70	C74	C[71-73]	C[8-69]				
1CAPEL/750;	C1	C2						
1CAPEL/125;	C4	C5						
1CAPEL/125;	C3	C7						
1CONN6;	J5							
1DIO500;	CR1	CR2	CR3	CR4				
1JUMPER2/100;		W1						
1OSC;	5R							
1OSC;	3R							
1RES;	R5							
1RES;	R9							
1RES;	R1	R2						
1RES;	R6							
1RES;	R3	R4						
1SSIDELED;	DS1	DS12	DS13	DS16	DS2	DS7	DS8	DS9
1SSIDELED;	DS10	DS11	DS14	DS15	DS3	DS4	DS5	DS6
\$NETS								
OPENCBL+;	J1.27	RP2.3	RP8.1					
INDEXB+ ;	J2.22	RP5.13	2T.4					
DUMMY+ ;	RP5.14	1P.3						
DUMMY- ;	RP5.4	1P.4						
RESET* ;	CR1.2	C3.1	RP7.6	J8.10	2W.16	2V.13,		
	3Y.1							
GND ;	J2.1	J2.7	J2.13	J2.4	J2.10	J2.16,		
	J2.21	J2.24	C[8-69].2		RP5.16	C70.1,		
	C1.2	C2.1	C3.2	7T.1	7T.19	J8.1,		
	J8.19	RP1.16	RP2.16	RP4.16	RP3.16	C[71-73].2,		
	J5.3	C74.1	1BB.4	1Z.4	1Y.4	1W.4,		
	2R.11	2X.10	3X.2	3X.4	3BB.12	3BB.4,		
	5AA.2	6AA.12	6AA.2	5AA.12	4Z.15	5Z.15,		
	3T.13	3T.12	3T.1	W1.1	4R.8	4R.12,		
	5S.6	6Z.10	J3.1	J3.3	J3.5	J3.7,		
	J3.9	J3.11	J3.13	J3.15	J3.17	J3.19,		
	J3.21	J3.23	J3.25	J3.27	J3.29	J3.31,		
	J3.33	J3.35	J3.37	J3.39	J3.41	J3.43,		
	J3.45	J3.47	J3.49	J4.5	J4.7	J4.9,		
	J4.11	J4.13	J4.15	J4.17	J4.19	J4.21,		
	J4.23	J4.25	J4.27	J4.29	J4.31	J4.33,		
	J4.35	J4.37	J4.39	J4.41	J4.43	J4.45,		
	J4.47	J4.49	J7.1	1C.8	1F.8	1E.19,		
	1E.1	2F.15	2F.1	2F.2	2F.4	7J.1,		
	7J.19	1B.19	1B.1	2A.13	2A.2	2A.5,		
	2A.14	2A.15	2C.3	2C.4	2C.7	2C.8,		
	2C.13	2C.14	1D.1	1D.19	2D.14	2D.17,		
	2D.18	C7.1	4N.19	2K.11	2L.3	2L.16,		
	2M.3	2M.16	2N.3	2N.16	5K.4	5K.7,		
	6K.7	3P.21	3P.20	3N.21	3N.20	3M.21,		
	3M.20	3L.21	3L.20	3K.21	3K.20	J6.1,		
	J6.11	J6.15	RP25.8	RP25.3	RP25.1	RP25.6,		
	3C.40	3E.40	3F.40	4J.1	4A.13	3B.3,		
	3B.4	3B.38	3B.37	3B.36	5B.13	6B.8,		
	6B.9	6B.10	6A.8					
+5V ;	C[8-69].1	C1.1		CR1.1	DS1.1	RP6.1,		
	DS3.1	DS2.1	DS5.1	DS4.1	DS7.1	DS6.1,		
	DS8.1	RP7.1	J5.5	J5.6	RP20.1	R2.2,		
	R5.1	R4.2	R3.2	R6.2	RP23.1	RP24.1,		

	1C.16	1F.16	DS16.1	DS15.1	DS14.1	DS13.1,
	DS12.1	DS11.1	DS9.1	DS10.1	CR4.1	J6.19,
	RP26.1	R9.1				
TAG1-	; J1.1	RP1.1	RP15.5			
TAG2-	; J1.3	RP1.2	RP14.5			
TAG3-	; J1.5	RP1.3	RP15.3			
TAG2+	; J1.4	RP1.4	RP14.7			
BUS1+	; J1.10	RP1.5	RP13.7			
BUS3-	; J1.13	RP1.6	RP13.3			
BUS2-	; J1.11	RP1.7	RP12.5			
BUS1-	; J1.9	RP1.8	RP13.5			
-5V	; C70.2	C2.2	C[71-73].1	J5.1	J5.2,	
	C74.2	R1.2	RP19.1			
BUS3+	; J1.14	RP1.10	RP13.1			
BUS2+	; J1.12	RP1.11	RP12.7			
BUS0+	; J1.8	RP1.12	RP14.1			
TAG3+	; J1.6	RP1.13	RP15.1			
BUS0-	; J1.7	RP1.14	RP14.3			
TAG1+	; J1.2	RP1.15	RP15.7			
BUS9-	; J1.25	RP2.1	RP9.5			
BUS6-	; J1.19	RP2.2	RP10.5			
PICK*	; J1.57	CR2.1				
OPENCBL-	J1.28	RP2.5	RP8.4			
BUS7-	; J1.21	RP2.6	RP11.3			
BUS9+	; J1.26	RP2.7	RP9.7			
BUS5-	; J1.17	RP2.8	RP11.5			
BUS8-	; J1.23	RP2.9	RP10.4			
BUS7+	; J1.22	RP2.10	RP11.1			
BUS6+	; J1.20	RP2.11	RP10.7			
BUS4-	; J1.15	RP2.12	RP12.4			
BUS5+	; J1.18	RP2.13	RP11.7			
BUS8+	; J1.24	RP2.14	RP10.1			
HOLD*	; J1.58	CR3.1				
BUS4+	; J1.16	RP2.15	RP12.1			
FAULT-	; J1.29	RP3.1	2Y.5			
INDEX+	; J1.36	RP3.2	2Z.4			
SEEKERR+	J1.32	RP3.3	2Y.3			
RDY+	; J1.38	RP3.4	2Y.12			
ONCYL-	; J1.33	RP3.5	2Y.13			
BUSY+	; J1.42	RP3.6	2Z.3			
SECSW2	; RP6.2	J8.9	6Z.15			
RDY-	; J1.37	RP3.8	2Y.11			
BUSY-	; J1.41	RP3.9	2Z.2			
USTAG-	; J1.43	RP3.10	RP9.1			
USTAG+	; J1.44	RP3.11	RP9.3			
SEEKERR-	J1.31	RP3.12	2Y.2			
INDEX-	; J1.35	RP3.13	2Z.5			
ONCYL+	; J1.34	RP3.14	2Y.14			
FAULT+	; J1.30	RP3.15	2Y.4			
UNITS1-	; J1.47	RP4.1	RP16.5			
OFFCYL*	; 7T.8	2Y.15	3Y.6			
SERVOCK-	J2.3	1P.2				
RDDATA-	; J2.5	1P.14				
USELD*	; 7T.2	2AA.8	3AA.6			
RDCLK-	; J2.9	1P.12				
WRDATA+	; J2.14	RP5.5	RP18.5			
WRPROT*	; RP6.7	7T.6	J8.5	2Z.15	2W.15	
BUS10+	; J1.60	RP4.2	RP8.7			
USELD+	; J2.17	RP5.8	2T.2			
FAULT*	; 7T.11	2Y.7	2V.14	2U.1	3Z.1	
SERVOCK+	J2.2	1P.1				

UNCRECC*	RP7.4	J8.6	3S.5	3S.10		
RDDATA+	J2.6	1P.15				
RDCLK+	J2.8	1P.13				
WRCLK+	J2.12	RP5.15	RP18.1			
SECTOR+	J1.50	RP4.3	2Z.12			
UN001000	DS1.2	7T.18				
USELD-	J2.18	RP5.10	2T.3			
UNITS0-	J1.45	RP4.4	RP17.5			
WRDATA-	J2.15	RP5.1	RP18.8			
UN001001	7T.16	DS2.2				
SECTORB+	J2.26	RP5.12	2T.12			
UN001002	7T.14	DS3.2				
INDEXB-	J2.23	RP5.9	2T.5			
SECTORB-	J2.25	RP5.11	2T.11			
UN001003	7T.12	DS4.2				
UN001004	7T.9	DS5.2				
UN001005	7T.7	DS6.2				
UN001006	7T.5	DS7.2				
UN001007	7T.3	DS8.2				
OFFLINE*	RP6.6	J8.3	3Y.11			
WRPROT+	J1.56	RP4.5	2Z.13			
WRCLK-	J2.11	RP5.3	RP18.3			
SECTOR-	J1.49	RP4.6	2Z.11			
UNITS2+	J1.52	RP4.7	RP17.1			
RDY*	7T.4	2Y.10	3Z.3			
SEEKERR*	7T.13	2Y.1	2W.18	3Y.3		
RGATE*	7T.15	2W.12	5U.15	6U.15		
WGATE*	7T.17	6T.2	6T.3	5T.2	5T.3	4U.6
UNITS3-	J1.53	RP4.8	RP16.4			
SECSW1	RP6.8	J8.7	6Z.8			
WRPROT-	J1.55	RP4.9	2Z.14			
UNITA2*	RP6.3	J8.11	3X.13			
UNITA1*	RP6.4	J8.13	3X.12			
BUS10-	J1.59	RP4.10	RP8.5			
CORRECC*	RP7.5	J8.8	3S.9			
DUALPORT	RP7.3	J8.12	3BB.3			
3MBSW	RP7.7	J8.14	5S.2	4S.5		
UNITA0*	RP7.8	J8.16	3X.10			
UNITS3+	J1.54	RP4.11	RP16.1			
UNITS1+	J1.48	RP4.12	RP16.7			
UNITS2-	J1.51	RP4.14	RP17.3			
UNITS0+	J1.46	RP4.15	RP17.7			
256K*	RP20.6	2X.5	3T.2	3T.15	W1.2	
UN002000	1BB.6	RP9.6				
UN002001	1BB.7	RP9.8				
UNITS2*	1BB.13	3X.14				
UNITS3*	1BB.3	3X.1				
BUS10*	1BB.11	2X.8				
BUS9*	1BB.5	2X.15	2AA.5			
UN002002	1Y.2	RP10.3				
UN002003	1Y.1	RP10.2				
UN002004	1Y.10	RP10.6				
UN002005	1Y.9	RP10.8				
UN002006	1Y.14	RP11.4				
UN002007	1Y.15	RP11.2				
UN002008	1Y.6	RP11.6				
UN002009	1Y.7	RP11.8				
BUS8*	1Y.3	2X.16				
BUS7*	1Y.13	2X.17				
BUS6*	1Y.11	2W.1	2V.1	2X.1		
BUS5*	1Y.5	2X.2				

UN002010;	1W.2	RP12.3				
UN002011;	1W.1	RP12.2				
UN002012;	1W.10	RP12.6				
UN002013;	1W.9	RP12.8				
UN002014;	1W.14	RP13.4				
UN002015;	1W.15	RP13.2				
UN002016;	1W.6	RP13.6				
UN002017;	1W.7	RP13.8				
BUS4*	; 1W.3	2V.6	2X.3			
BUS3*	; 1W.13	2W.2	2V.2	2X.4		
BUS2*	; 1W.11	2W.3	2V.3	2X.7		
BUS1*	; 1W.5	2W.4	2V.4	2X.6		
UN002018;	1U.14	RP14.4				
UN002019;	1U.15	RP14.2				
UN002020;	1U.6	RP14.6				
UN002021;	1U.7	RP14.8				
UN002022;	1U.2	RP15.4				
UN002023;	1U.1	RP15.2				
UN002024;	1U.10	RP15.6				
UN002025;	1U.9	RP15.8				
TAGEN*	; 1U.4	3Z.8				
TAG3*	; 1U.3	RP20.2	2W.11	2V.11	2AA.6	
TAG2*	; 1U.5	RP20.3	2W.9	2V.9		
TAG1*	; 1U.11	RP20.4	2W.8	2V.8		
PUL	; RP20.10	6T.18	6T.1	5T.18	5T.1	
UN002026;	1Z.6	RP17.6				
UN002027;	1Z.7	RP17.8				
WRDAT	; 1S.5	4R.9				
UNITS0*	; 1Z.5	3X.9				
OPENCBL*	; 1Z.3	3Y.13				
UNITS1*	; 1Z.11	3X.11				
USTAG	; 1Z.13	3Z.12	3Z.13	3AA.3		
WRCLK	; 1S.13	5S.12	2S.9			
BUSY*	; 2Z.1	3AA.8	3Y.9			
SECTOR*	; 2T.10	2Z.10	4AA.8			
UN002028;	1Z.1	RP8.2	R2.1			
INDEX*	; 2T.7	2Z.7	4AA.11			
UN002029;	RP18.6	1S.7				
UN002030;	RP18.7	1S.6				
USELD	; 2T.1	2Z.6	2Y.6	3Z.5	3AA.11	3Z.10,
		3AA.5				
UN002031;	RP18.2	1S.15				
UN002032;	RP18.4	1S.14				
SERVOCLK;	2R.10	5S.4				
RDCLK	; 2R.7	4U.10				
RDDATA	; 2R.5	5T.17				
UN002034;	2R.15	1P.6	RP19.3			
UN002035;	2R.1	1P.11	RP19.7			
BUS0*	; 1U.13	RP20.5	2W.5	2V.5		
UN002036;	2R.2	1P.10	RP19.6			
UN002037;	2R.14	1P.5	RP19.4			
UN002038;	1Z.14	RP9.4				
UN002039;	2R.12	RP19.5				
PU2	; RP20.8	3U.4	3U.13	3U.10	2S.1	4T.1,
		3V.3	3V.4	3V.5	3V.6	3V.9
		3V.10	3V.1	3W.10	3W.2	3V.7,
					3W.4	
UN002040;	1Z.15	RP9.2				
UN002041;	2R.3	RP19.8				
UN002042;	2R.4	RP19.2				
UN002043;	2R.13	RP19.10				
PU3	; 2T.6	RP20.7	2R.6	3AA.10	3X.15	3X.3,

	3BB.13	3AA.4				
UN002044;	1BB.10	RP8.6				
UN002045;	1BB.9	RP8.8				
UN002046;	1BB.14	RP17.4				
UN002047;	1BB.15	RP17.2				
UN002048;	1BB.2	RP16.3				
UN002049;	1BB.1	RP16.2				
UN002050;	1Z.10	RP16.6				
UN002051;	1Z.9	RP16.8				
UN002052;	1Z.2	RP8.3	R1.1			
ONCYL ;	2W.17	3Y.4	3Y.5	3BB.9		
UN003000;	3Z.11	3Z.4				
WGATE ;	2W.13	2U.6	3S.1	3S.2	2S.6	4U.5
R+W ;	2V.19	2U.4				
UN003001;	3X.6	3AA.2				
UN003002;	3Y.12	3AA.13	3AA.1			
ILLCYL ;	2W.6	2X.14				
ILLHD ;	2W.7	2X.13				
MULTIBIT;	2W.14	2X.12				
CADD2 ;	2V.18	2U.14				
UN003003;	3Y.2	3Z.6				
UN003004;	3Y.10	2AA.1				
CADD1 ;	2V.17	2U.13				
CADD0 ;	2V.16	2U.11				
HADD0 ;	2V.15	2U.3				
FLT* ;	2W.19	2V.7				
CYLADD2 ;	2U.15	3T.11				
CYLADD1 ;	2U.12	3T.5				
CYLADD0 ;	2U.10	4Z.3	3U.1			
HDADD0 ;	2U.2	4Z.6				
UN003005;	C4.1	3BB.1				
WRT ;	2U.7	4U.13				
UN003006;	R6.1	C4.2	3BB.2			
SSEEK* ;	2V.12	3BB.11				
R+WGATE ;	2U.5	6T.9	5T.9			
WRLOAD* ;	2U.9	3U.8	5U.11	6U.11		
UN003007;	C5.1	3BB.15				
UN003008;	R5.2	C5.2	3BB.14			
UN003009;	3AA.12	3BB.6				
UN003010;	2AA.12	2AA.4				
UN003011;	2AA.13	2AA.9				
UN003012;	2AA.11	2AA.10	3BB.5			
UN003013;	3Z.2	3Z.9	3Y.8			
UN003014;	R4.1	2AA.3	CR3.2			
UN003015;	R3.1	2AA.2	CR2.2			
MA0 ;	3T.7	4X.5	4V.5	5V.5	5X.5	4W.5,
	4Y.5	5W.5	5Y.5	7W.5	7Y.5	6W.5,
	6Y.5	7V.5	7X.5	6V.5	6X.5	
MA3 ;	5Z.4	4X.12	4V.12	5V.12	5X.12	4W.12,
	4Y.12	5W.12	5Y.12	7W.12	7Y.12	6W.12,
	6Y.12	7V.12	7X.12	6V.12	6X.12	
UN004000;	3W.5	4U.9				
UN004001;	5BB.6	5BB.10	4AA.13	4AA.10		
20MHZ ;	5R.8	5S.3	5L.13			
UN004002;	5BB.8	3U.2				
MA2 ;	5Z.7	4X.6	4V.6	5V.6	5X.6	4W.6,
	4Y.6	5W.6	5Y.6	7W.6	7Y.6	6W.6,
	6Y.6	7V.6	7X.6	6V.6	6X.6	
MA8 ;	3T.9	4X.1	4V.1	5V.1	5X.1	4W.1,
	4Y.1	5W.1	5Y.1	7W.1	7Y.1	6W.1,
	6Y.1	7V.1	7X.1	6V.1	6X.1	



WC10	;	6AA.9	5Z.6	6Z.2			
UN004025;		4R.10	4R.11				
WRDATA	;	4R.13	6T.11				
WC4	;	6AA.3	4Z.14	5BB.1			
WC3	;	6AA.1	5AA.8	5Z.2	5BB.2		
WC2	;	5AA.9	5Z.5	5BB.4			
UN004026;		4AA.12	6Z.14				
UN004027;		4AA.9	6Z.13				
BITCT15*;		4U.4	5U.13	6U.13			
D11	;	7V.2	7V.14	6U.6			
WRT*	;	4X.3	4V.3	5V.3	5X.3	4W.3	4Y.3,
		5W.3	5Y.3	7W.3	7Y.3	6W.3	6Y.3,
		7V.3	7X.3	6V.3	6X.3	4U.12	5U.9,
		6U.9					
UN005000;		4V.2	4S.3				
D16	;	6X.2	6X.14	6U.1			
D12	;	6V.2	6V.14	6U.4			
D15	;	7X.2	7X.14	6U.7			
D14	;	6Y.2	6Y.14	6U.3			
D10	;	6W.2	6W.14	6U.2			
D13	;	7Y.2	7Y.14	6U.8			
D9	;	7W.2	7W.14	6U.5			
D8	;	4X.2	4X.14	5U.1			
D4	;	4V.14	4S.2	5U.6			
D7	;	5X.2	5X.14	5U.8			
D3	;	5V.2	5V.14	5U.4			
D6	;	4Y.2	4Y.14	5U.3			
D2	;	4W.2	4W.14	5U.2			
D5	;	5Y.2	5Y.14	5U.5			
D1	;	5W.2	5W.14	5U.7			
UN005001;		6T.17	5T.11				
UN005002;		5U.23	5T.7				
UN005003;		5U.16	5T.13				
UN005004;		5U.21	5T.6				
UN005005;		5U.19	5T.14				
UN005006;		5U.18	5T.5				
UN005007;		5U.20	5T.15				
UN005008;		5U.22	5T.4				
UN005009;		5U.17	5T.16				
UN005010;		6T.7	6U.23				
UN005011;		6T.13	6U.17				
UN005012;		6T.6	6U.21				
UN005013;		6T.14	6U.16				
UN005014;		6T.5	6U.20				
UN005015;		6T.15	6U.18				
UN005016;		6T.4	6U.22				
UN005017;		6T.16	6U.19				
DISEOT*	;	J7.13	RP23.10	6K.15			
IRP*	;	J4.1	1J.3				
IR1*	;	J4.3	1G.4				
SFAD	;	J7.8	RP24.7	2F.14			
100KB*	;	J7.11	RP23.9	6K.14			
IGO*	;	J3.8	1F.1	1E.2			
ILOL*	;	J3.16	1F.2	1E.4			
IDEN*	;	J4.50	1F.7	1E.11			
SMRES*	;	J7.3	RP24.5	CR4.2	C7.2	4N.11	
IREV*	;	J3.18	1C.15	1D.2			
SRDEN*	;	J7.18	RP23.6	2E.15			
STAD1	;	J7.6	RP24.6	2F.11			
IONL*	;	J4.44	2J.8				
UN006000;		1E.9	2E.2				

UN006001;	1E.18	2E.3				
UN006002;	1E.16	2E.4				
UN006003;	1E.14	2E.5				
UN006004;	1E.12	2E.6				
IWFM*	; J3.42	1F.10	1D.13			
SFPT	; J7.20	RP23.8	7J.13	1J.5		
ITAD1*	; J4.46	1F.5	1E.15			
UN006005;	1E.7	2F.13				
UN006006;	1E.5	2F.12				
ITAD0*	; J3.46	1F.4	1E.17			
DIAGPASS;	7J.2	5N.12				
IR2*	; J3.48	1G.2				
IR3*	; J3.50	1G.12				
IR0*	; J4.2	1G.6				
SOFL*	; J7.12	RP24.10	2E.11			
IR4*	; J4.6	1G.10				
IR7*	; J4.8	1H.4				
IR6*	; J4.10	1H.6				
IHER*	; J4.12	1K.2	1J.11			
IFMK*	; J4.14	1H.12				
ICCG*	; J4.16	1K.8				
DIAGFAIL;	7J.4	5J.4	5J.3			
IR5*	; J4.20	1H.2				
PU6	; 2F.3	RP26.8				
MMEMERR	; 7J.6	6M.2	3A.9			
IDBY*	; J4.38	1K.4				
FBY	; 1G.9	3J.3	3A.1	2D.11	2G.1	2H.1,
	5K.14					
UN006007;	2E.18	3J.1	1K.3			
IRSTR*	; J4.34	1K.10				
IWSTR*	; J4.36	1K.12				
INRZ*	; J4.26	J4.40	1H.10			
SONL*	; J7.10	RP24.2	2E.9			
ICER*	; J4.42	1J.8				
365KB*	; J7.9	RP24.9	6K.13			
EOT	; 7J.15	2J.5	5P.11			
IEOT*	; J4.22	2J.6				
LDP	; 2E.8	7J.17	2J.2	5P.12		
IEDIT*	; J3.38	1C.14	1D.17			
IREW*	; J3.20	1F.14	1E.6			
IOFL*	; J4.24	1F.6	1E.8			
IW6*	; J3.28	1C.13	1B.6			
IERASE*	; J3.40	1C.12	1D.15			
IWP*	; J3.22	1C.11	1D.6			
STAD0	; J7.4	RP24.3	2F.9			
455KB*	; J7.7	RP24.8	6K.12			
IWRT*	; J3.34	1C.10	1D.8			
IFAD*	; J4.48	1F.11	1E.13			
IW3*	; J3.26	1C.9	1B.11			
SPARERR*;	J7.5	RP24.4	1J.1			
ILDLP*	; J4.4	2J.3				
MSEL	; 2F.6	2E.1	7J.8	2J.4	2J.1	3J.2,
	2J.9	2J.13	1J.4	3J.13		
SREW*	; J7.14	RP23.4	2E.13			
SINTPAR*;	J7.19	RP23.7	2A.1			
SHER	; J7.17	RP23.5	1J.12			
SCER	; J7.15	RP23.3	1J.9			
UN006008;	1E.3	2F.10				
SHDEN*	; J7.16	RP23.2	2E.14			
IFEN*	; J4.18	1F.9	1D.11			
IW2*	; J3.30	1C.7	1B.13			

IW7*	; J3.24	1C.5	1B.8			
UN006009;	7J.18	DS16.2				
UN006010;	7J.16	DS15.2				
UN006011;	7J.14	DS14.2				
UN006012;	7J.12	DS13.2				
UN006013;	7J.9	DS12.2				
UN006014;	7J.7	DS11.2				
UN006015;	7J.5	DS10.2				
UN006016;	7J.3	DS9.2				
IW1*	; J3.12	1C.6	1B.15			
ILWD*	; J3.4	1C.4	1D.4			
IW0*	; J3.10	1C.3	1B.17			
IW5*	; J3.32	1C.2	1B.4			
IW4*	; J3.6	1C.1	1B.2			
UN006017;	2E.19	1H.9				
MRESFBY*;	2E.7	5J.10				
IRDY*	; J4.28	1H.8				
IFBY*	; J3.2	1G.8				
ONL	; 2E.17	7J.11	2J.10	6N.1		
RWD	; 2E.16	2J.12	5K.15			
NRZ	; 2E.12	1H.11	5C.13			
IRWD*	; J4.30	2J.11				
IFPT*	; J4.32	1J.6				
UN007000;	1B.3	1A.11	2B.8			
UN007001;	1B.5	1A.10	2B.4			
UN007002;	1B.7	1A.12	2B.17			
UN007003;	1B.9	1A.8	2B.18			
UN007004;	2A.3	2A.10	1D.14			
UN007005;	1B.16	1A.4	2B.14			
UN007006;	1B.18	1A.9	2B.13			
UN007007;	1B.12	1A.1	2B.3			
UN007008;	1A.2	2A.4				
UN007009;	1A.5	2A.6	2A.11			
UN007010;	2H.10	1J.2				
UN007011;	2C.17	1D.16				
UN007012;	2A.7	3A.2				
UN007013;	2A.9	2B.7				
UN007014;	2H.12	1H.13				
DATAIN*	; 2C.1	2B.1	4J.4			
UN007015;	1D.18	2D.13				
SRC0	; 2B.9	4N.18	3C.25	5C.5		
SRC1	; 2B.5	4N.16	3C.24	5C.6		
SRC2	; 2B.16	4N.14	3C.23	5C.9		
SRC3	; 2B.19	4N.12	3C.22	6C.2		
SRC4	; 2B.12	3H.12	3E.25	5C.15		
SRC5	; 2B.15	3H.9	3E.24	6C.6		
SRC6	; 2C.19	3H.15	3E.23	6C.15		
MRES*	; 3J.8	5H.13	2K.4	4M.6	4M.1	4K.1,
	4L.1	4P.1	5N.15	5P.15	4A.1	
RDBAD	; 5J.11	5J.5	5N.10			
UN007016;	3A.5	6M.1				
UN007017;	1D.12	2D.4				
UN007018;	1D.7	2D.7				
UN007019;	1D.3	2D.8				
UN007020;	1D.5	2D.3				
RSTBSY	; 5J.8	5P.7				
RDCMD*	; 2D.1	4J.5				
WSTR	; 5H.1	5P.9	1K.13			
UN007021;	2G.2	1G.5				
UN007022;	2G.5	1G.3				
UN007023;	2G.7	1G.1				

UN007024;	2G.10	1G.13				
UN007025;	2G.12	1G.11				
UN007026;	2G.15	1H.1				
UN007027;	2H.2	1H.5				
UN007028;	2H.5	1H.3				
UN007029;	5J.6	5J.1				
DEST0	; 2G.3	3C.36	4A.4	3B.11	5A.14	
DEST1	; 2G.4	3C.37	4A.14	3B.16	5A.17	
DEST2	; 2G.6	3C.38	4A.3	3B.17	5A.8	
SRC11	; 2C.12	2D.12	3H.5	3F.22	5C.12	
DEST4	; 2G.13	3E.36	3B.26	5A.13		
DEST3	; 2G.11	3C.39	4A.11	3B.23	6A.18	
DEST5	; 2G.14	3E.37	3B.29	6A.13		
DEST6	; 2H.3	3E.38	3B.30	6A.14		
DEST7	; 2H.4	5K.2	3E.39	3B.35	6A.17	
DEST8	; 2H.6	6K.4	3F.36	3B.12	5A.3	
DEST9	; 2H.11	6K.3	3F.37	3B.15	5A.18	
DEST10	; 2H.13	6K.2	3F.38	3B.18	5A.4	
DEST11	; 2H.14	6K.1	3F.39	3B.22		
DATAOUT*;	2G.9	2H.9	4J.10			
UN007030;	5H.12	5J.9	5J.12	5J.2		
UN007031;	3A.6	3A.4				
UN007032;	3A.3	2C.11	2B.11	5H.2		
SRC10	; 2C.15	2D.5	3H.19	3F.23	5C.16	
SRC9	; 2B.6	2D.6	3H.16	3F.24	5C.2	
SRC8	; 2C.16	2D.9	3H.6	3F.25	5C.19	
SRC7	; 2B.2	2D.2	3H.2	3E.22	6C.19	
UN007033;	1B.14	1A.13	2C.18			
PARERR	; 6M.3	5K.3				
UN007034;	3J.5	6M.11				
EXTRES*	; 3J.4	J6.18	RP26.10			
UN007035;	3J.9	4N.9				
RMMERR*	; 5J.13	3A.13				
UN007036;	3J.6	3J.10				
UN007037;	1D.9	6M.13				
UN007038;	6M.12	6N.2				
PU8	; 2N.17	3P.19	3P.18	3N.19	3N.18	3M.19,
	3M.18	3L.19	3L.18	3K.19	3K.18	RP26.3
PU9	; RP26.7	5L.10	5L.3	5L.2	5L.4	3B.9
PUREF*	; RP26.2	3B.39	1K.6			
MRES D*	; 2K.7	4M.7				
UN008000;	3K.15	4K.13				
UN008001;	3K.14	4M.14				
UN008002;	3K.13	4M.4				
UN008003;	3K.11	4M.13				
UN008004;	3K.10	4M.3				
UN008005;	3K.9	4K.3				
UN008006;	3L.17	4L.18				
UN008007;	3L.16	4L.17				
UN008008;	3L.15	4L.3				
UN008009;	3L.14	4L.8				
UN008010;	3L.13	4L.13				
UN008011;	3L.11	4L.7				
UN008012;	3L.10	4L.4				
UN008013;	3L.9	4L.14				
UN008014;	3M.17	4K.4				
UN008015;	3M.16	4K.6				
UN008016;	3M.15	4K.11				
UN008017;	3M.14	4K.14				
UN008018;	3N.17	4P.14				
UN008019;	3N.16	4P.8				

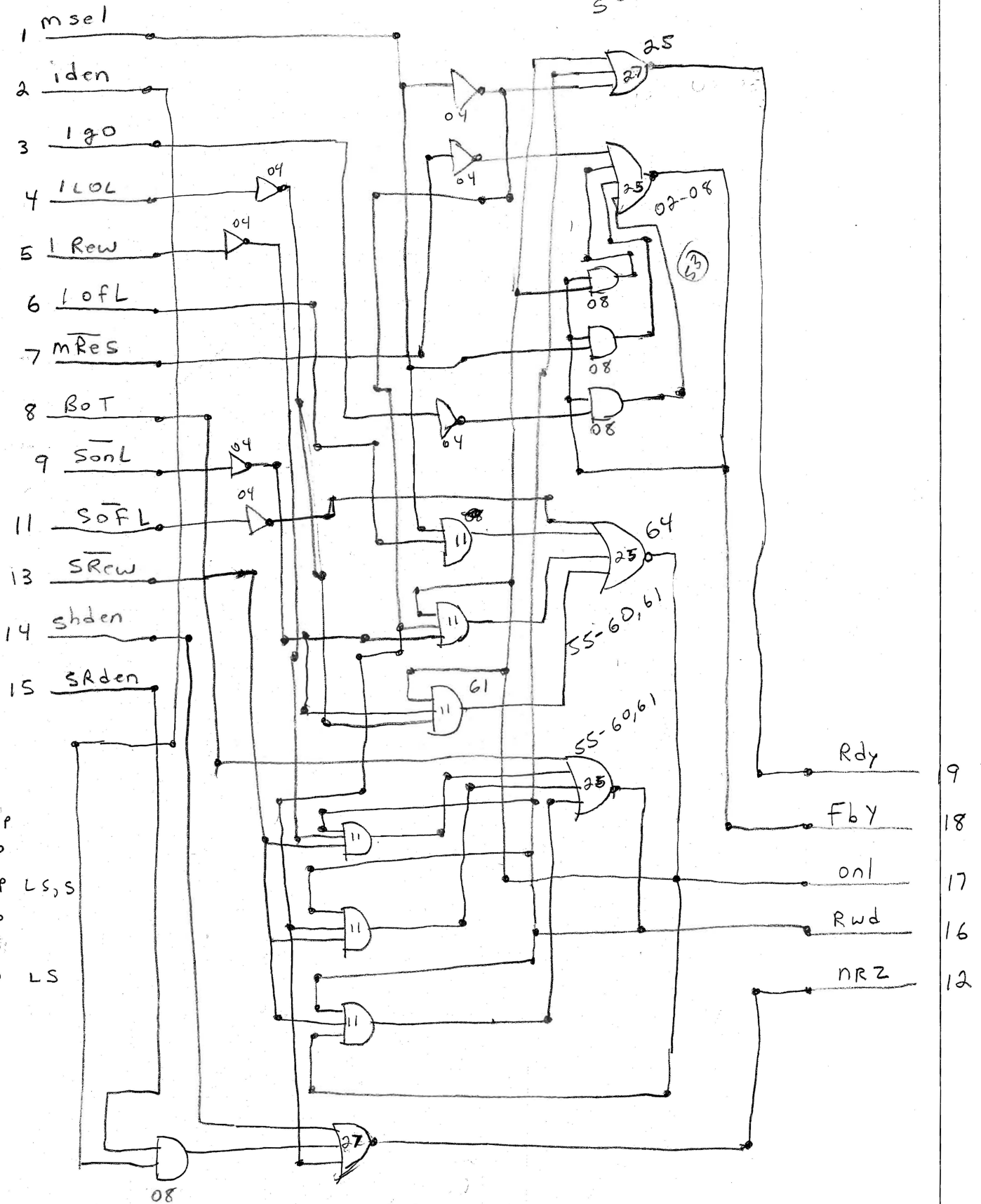
UN008020;	3N.15	4P.18					
UN008021;	3N.14	4P.17					
CSD11	; 3N.13	3H.4					
CSD9	; 2L.6	3N.10	3H.17				
CSA9	; 2L.13	3P.22	3N.22	3M.22	3L.22	3K.22,	
	J6.14						
CSA8	; 2L.12	3P.23	3N.23	3M.23	3L.23	3K.23,	
	J6.13						
CSA7	; 2M.15	3P.1	3N.1	3M.1	3L.1	3K.1,	
	J6.12						
CSA6	; 2M.14	3P.2	3N.2	3M.2	3L.2	3K.2,	
	J6.9						
CSA5	; 2M.13	3P.3	3N.3	3M.3	3L.3	3K.3,	
	J6.8						
CSA4	; 2M.12	3P.4	3N.4	3M.4	3L.4	3K.4,	
	J6.7						
CSA3	; 2N.15	3P.5	3N.5	3M.5	3L.5	3K.5,	
	J6.6						
CSA2	; 2N.14	3P.6	3N.6	3M.6	3L.6	3K.6,	
	J6.5						
CSA1	; 2N.13	3P.7	3N.7	3M.7	3L.7	3K.7,	
	J6.4						
CSA0	; 2N.12	3P.8	3N.8	3M.8	3L.8	3K.8,	
	J6.3						
CSD8	; 2L.7	3N.9	3H.7				
UN008022;	2L.17	2M.18					
CSD19	; 2K.9	3M.13					
CSD10	; 2K.8	3N.11	3H.18				
UN008023;	2K.5	5K.6					
UN008024;	2K.6	6K.6					
CSD7	; 2M.4	3P.17	3H.3				
CSD6	; 2M.5	3P.16	3H.14				
CSD5	; 2M.6	3P.15	3H.8				
CSD4	; 2M.7	3P.14	3H.13				
CSD3	; 2N.4	3P.13	4P.13				
CSD2	; 2N.5	3P.11	4P.7				
CSD1	; 2N.6	3P.10	4P.4				
CSD0	; 2N.7	3P.9	4P.3				
UN008025;	2M.17	2N.18					
CSD18	; 5K.9	6K.9	3M.11				
CSD17	; 5K.10	6K.10	3M.10				
CSD16	; 5K.11	6K.11	3M.9				
ENFLG*	; 2K.18	6M.10					
S0	; 2K.16	2L.10	2M.10	2N.10			
PUP	; 2K.14	2L.20	2M.20	2N.20			
S1	; 2K.17	2L.11	2M.11	2N.11			
FE*	; 2K.15	2L.19	2M.19	2N.19			
ZE*	; 2K.13	2L.9	2M.9	2N.9			
UN008026;	2K.2	3K.17					
UN008027;	2K.3	3K.16					
TPGO	; J6.17	RP26.4	5L.11	5L.12			
MCLK	; 2K.1	2L.1	2M.1	2N.1	J6.16	4M.9,	
	4K.9	4L.11	4P.11	3H.11	3C.15	3E.15,	
	3F.15	5P.14	3J.12	5L.5	4J.15	3B.1	
SRCADD1	; 4K.12	4J.3					
SRCADD0	; 4M.15	4J.2					
DESTADD1;	4M.5	4J.13					
DESTADD0;	4M.12	4J.14					
I8	; 4M.2	3C.6	3E.6	3F.6			
I7	; 4K.2	3C.7	3E.7	3F.7			
I6	; 4L.19	3C.5	3E.5	3F.5			

I5	;	4L.16	3C.27	3E.27	3F.27		
I4	;	4L.2	3C.28	3E.28	3F.28		
I3	;	4L.9	3C.26	3E.26	3F.26		
I2	;	4L.12	3C.14	3E.14	3F.14		
I1	;	4L.6	3C.13	3E.13	3F.13		
I0	;	4L.5	3C.12	3E.12	3F.12		
CN	;	4L.15	3C.29				
AB3	;	4K.5	3C.1	3C.20	3E.1	3E.20	3F.1,
		3F.20					
AB2	;	4K.7	3C.2	3C.19	3E.2	3E.19	3F.2,
		3F.19					
AB1	;	4K.10	3C.3	3C.18	3E.3	3E.18	3F.3,
		3F.18					
AB0	;	4K.15	3C.4	3C.17	3E.4	3E.17	3F.4,
		3F.17					
FLGD	;	4P.15	5P.13				
FLGA2	;	4P.9	5P.3				
FLGA1	;	4P.19	5P.2				
FLGA0	;	4P.16	5P.1				
PIPE3	;	4P.12	4N.8	5N.13			
PIPE2	;	4P.6	4N.6	5N.3			
PIPE1	;	4P.5	4N.4	5N.2			
PIPE0	;	4P.2	4N.2	5N.1			
MRDY	;	5K.1	3B.2				
EQUAL	;	5K.13	3C.11	3E.11	3F.11	R9.2	
COUT	;	5K.12	3F.33				
ENCON*	;	3H.1	4N.1	4J.7			
MEMRD*	;	4J.6	5H.5	5C.1	5C.11	6C.1	6C.11
MEMWR*	;	4J.11	6A.11	5A.11			
CA*	;	4J.9	4A.9				
MFLGCLK*	;	5N.14	6M.8				
ALE	;	5P.10	5H.11				
REFREQ	;	5P.6	1K.5				
UN009000	;	5P.5	1K.11				
DE	;	5N.11	5H.9				
MWR	;	5N.9	5H.3				
UN009001	;	5N.7	1K.9				
PARTEST	;	5N.6	6B.1				
UN009002	;	1J.13	1J.10	3J.11			
UN009003	;	5N.5	1K.1				
UN009004	;	5L.9	5L.1				
UN009005	;	5L.6	6M.5				
UN009006	;	5L.8	6M.4				
UN009007	;	6M.6	6M.9				
UN009008	;	3E.33	3F.29				
UN009009	;	RP25.4	3C.21				
UN009010	;	RP25.7	3C.9				
UN009011	;	3C.33	3E.29				
UN009012	;	3C.8	3E.9				
UN009013	;	3C.16	3E.21				
UN009014	;	RP25.2	3F.16				
UN009015	;	3E.8	3F.9				
UN009016	;	3E.16	3F.21				
UN009017	;	RP25.5	3F.8				
MQPAR	;	6B.4	6A.9	5G.14			
MDWP	;	6B.13	5A.19	5C.3	5E.2	5E.14	
MMA8	;	4A.12	6D.1	6E.1	6F.1	6G.1	7D.1,
		7E.1	7F.1	7G.1	5D.1	5E.1	5F.1,
		5G.1					
UN010000	;	3A.11	5H.6				
MDPAR	;	6B.5	3A.12	5G.2			

UNØ1ØØØ1;	3A.8	3A.1Ø				
MDLWD	; 6B.12	5A.2	5C.18	5D.2	5D.14	
MDFMK	; 6B.2	5A.5	5C.17	5F.2	5F.14	
MMA7	; 3B.33	6D.9	6E.9	6F.9	6G.9	7D.9,
	7E.9	7F.9	7G.9	5D.9	5E.9	5F.9,
	5G.9					
MMA6	; 3B.32	6D.13	6E.13	6F.13	6G.13	7D.13,
	7E.13	7F.13	7G.13	5D.13	5E.13	5F.13,
	5G.13					
MMA5	; 3B.27	6D.1Ø	6E.1Ø	6F.1Ø	6G.1Ø	7D.1Ø,
	7E.1Ø	7F.1Ø	7G.1Ø	5D.1Ø	5E.1Ø	5F.1Ø,
	5G.1Ø					
MMA4	; 3B.24	6D.11	6E.11	6F.11	6G.11	7D.11,
	7E.11	7F.11	7G.11	5D.11	5E.11	5F.11,
	5G.11					
MMA3	; 3B.21	6D.12	6E.12	6F.12	6G.12	7D.12,
	7E.12	7F.12	7G.12	5D.12	5E.12	5F.12,
	5G.12					
MMA2	; 3B.19	6D.6	6E.6	6F.6	6G.6	7D.6,
	7E.6	7F.6	7G.6	5D.6	5E.6	5F.6,
	5G.6					
MMA1	; 3B.14	6D.7	6E.7	6F.7	6G.7	7D.7,
	7E.7	7F.7	7G.7	5D.7	5E.7	5F.7,
	5G.7					
MMAØ	; 3B.13	6D.5	6E.5	6F.5	6G.5	7D.5,
	7E.5	7F.5	7G.5	5D.5	5E.5	5F.5,
	5G.5					
MRAS*	; 3B.6	6D.4	6E.4	6F.4	6G.4	7D.4,
	7E.4	7F.4	7G.4	5D.4	5E.4	5F.4,
	5G.4					
MCAS*	; 3B.1Ø	6D.15	6E.15	6F.15	6G.15	7D.15,
	7E.15	7F.15	7G.15	5D.15	5E.15	5F.15,
	5G.15					
MWR*	; 5H.4	6D.3	6E.3	6F.3	6G.3	7D.3,
	7E.3	7F.3	7G.3	5D.3	5E.3	5F.3,
	5G.3					
MDØ	; 5B.11	5A.15	5C.4	6D.2	6D.14	
MD4	; 5B.1	5A.12	5C.14	7D.2	7D.14	
MD1	; 5B.1Ø	5A.16	5C.7	6E.2	6E.14	
MD5	; 5B.8	6A.12	6C.7	7E.2	7E.14	
MD2	; 5B.12	5A.9	5C.8	6F.2	6F.14	
MD6	; 5B.4	6A.15	6C.14	7F.2	7F.14	
MD3	; 5B.9	6A.19	6C.3	6G.2	6G.14	
MD7	; 5B.2	6A.16	6C.18	7G.2	7G.14	
UNØ1ØØØ2;	5B.5	6B.11				
UNØ1ØØØ3;	5H.8	6A.1	5A.1			
UNØ1ØØØ4;	4A.1Ø	3B.34				
UNØ1ØØØ5;	4A.2	3B.31				
UNØ1ØØØ6;	4A.15	3B.28				
UNØ1ØØØ7;	4A.5	3B.25				
UNØ1ØØØ8;	3B.5	3B.8	5H.1Ø			
§END						

04-2 53-1 08-1  
 27-1 64-1 55,60,61-1

52+04 53  
 50,51,54,55,60,64,65



42-381 50 SHEETS 5 SQUARE  
 42-382 100 SHEETS 5 SQUARE  
 42-389 200 SHEETS 5 SQUARE  
 NATIONAL  
 MADE IN U.S.A.

2-04-14P  
 08-14P  
 4-11-14P LS,S  
 4-23-16P  
 4-25-14P  
 4-27-14P LS

11 = 10 + 04

## DESCRIPTION

S03 - Disk & Tape Emulator  
Mag Tape Status Control PAL @ 2E

This PAL controls the status reporting for the mag tape interface on the emulator. There are 5 outputs used by this PAL. The other 3 bidirectional pins on the PAL are used for extra inputs.

The 5 outputs are :

- RDY      Ready  
Set when the device is selected, on-line, and not rewinding.
- FBY      Formatter Busy  
This output is latched when the device is selected, on-line, and a GO pulse is received. The latch is reset by the firmware sending MRESFBY.
- ONL      On Line  
This output is latched when the device is selected and a LOL pulse is received, or when the On Line switch is depressed. The latch is reset when the device is selected and a OFL pulse is received, or when the Off Line switch is depressed.
- RWD      Rewinding  
This output is latched when the device is selected, BOT is not set, and a REW pulse is received, or when the Rewind switch is depressed. The latch is reset when BOT is set by the firmware.
- NRZ      Non Return to Zero  
Set when the device is selected and  
either 1. The low density switch is set  
or        2. The remoted density switch is set and  
          the IDEN signal is not asserted.

MSEL IDEN IGO ILOL IREW IOFL /MRESFBY BOT /SONL GND  
 /SOFL NRZ /SREW /SHDEN /SRDEN RWD ONL FBY RDY VCC

/RDY = /ONL + RWD + /MSEL

/FBY = MRESFBY + /FBY\*/ONL + /FBY\*/MSEL + /FBY\*/IGO

/ONL = SOFL + MSEL\*IOFL + /ONL\*/MSEL\*/SONL + /ONL\*/ILOL\*/SONL

/RWD = BOT + /RWD\*/IREW\*/SREW + /RWD\*/MSEL\*/SREW + /RWD\*/ONL\*/SREW

/NRZ = SHDEN + SRDEN\*IDEN + /MSEL

FUNCTION TABLE

MSEL IDEN IGO ILOL IREW IOFL /MRESFBY BOT /SONL  
 /SOFL /SREW /SHDEN /SRDEN RDY FBY ONL RWD NRZ

```

;           /   /   /   /   /
; M I I I I I M B S S S S S           R F O R N
; S D G L R O R O O O R H R           D B N W R
; E E O O E F E T N F E D D           Y Y L D Z
; L N   L W L S   L L W E E
;           F           N N

```

---

H X L H L L L H H H H H H	H L H L H	Set On-line
H X L L L L H H H L H H H	L L L L H	Switch Off-line
H X L L L L H H L H H H H	H L H L H	Switch On-line
H L H L L L H H H H H H H	H H H L H	Send command
H H L L L L H L H H H H H	H H H L H	Test FBY hold
H H L L L L L L H H H H L	H L H L L	Reset FBY, Change DEN
H L L L L H L H L H H H L	L L H H H	Rewind, Change DEN
H L L L L L H L H H H L H	L L H H L	Test RWD hold, HDEN
H L L L L L H H H H H L H	H L H L L	Reach BOT
H H L L L L H L H H H L H	H L H L L	Drift off BOT
H H L L L L H L H H L L H	L L H H L	Switch Rewind
L L L L L L H L H H H L H	L L H H L	Wait BOT
L L L L L L H H H H H L H	L L H L L	Reach BOT
L L H H H H H L H H H H H	L L H L L	Test /MSEL

---

\* and

+ or

/ not

S03 - Tape Status @ 2E Rev A0

```

*****
*
MSEL  ****
* 1*          P A L          *20*  VCC
*  ****
*          1 6 L8          *
*  ****
IDEN  * 2*          *19*  RDY
*  ****
*          *
*  ****
IGO   * 3*          *18*  FBY
*  ****
*          *
*  ****
ILOL  * 4*          *17*  ONL
*  ****
*          *
*  ****
IREW  * 5*          *16*  RWD
*  ****
*          *
*  ****
IOFL  * 6*          *15*  /SRDEN
*  ****
*          *
/MRESFBY * 7*          *14*  /SHDEN
*  ****
*          *
*  ****
BOT   * 8*          *13*  /SREW
*  ****
*          *
*  ****
/SOVL * 9*          *12*  NRZ
*  ****
*          *
*  ****
GND   *10*          *11*  /SOFL
*  ****
*          *
*****

```

PAL16L8

PAL DESIGN SPECIFICATION

PAL S03 Disk & Tape Emulator @ 2E Revision A0

S03 - Tape Status @ 2E Rev A0

Steve Blightman 10/15/86

\*

G0\*F0\*

L0000 11111111111111111111111111111111\*

L0032 11111111110111111111111111111111\*

L0064 11111111111111011111111111111111\*

L0096 11101111111111111111111111111111\*

L0256 11111111111111111111111111111111\*

L0288 1111111111111111111111011111111111\*

L0320 11111101110111111111111111111111\*

L0352 11101110111111111111111111111111\*

L0384 11111010111111111111111111111111\*

L0512 11111111111111111111111111111111\*

L0544 11111111111111111111111111111110\*

L0576 11011111111111110111111111111111\*

L0608 11101111111011111111111111110111\*

L0640 1111111101011111111111111110111\*

L0768 11111111111111111111111111111111\*

L0800 11111111111111111111111011111111\*

L0832 11111111111010111111111101111111\*

L0864 11101111111111011111111110111111\*

L0896 11111111110111011111111110111111\*

L1792 11111111111111111111111111111111\*

L1824 11111111111111111111111011111111\*

L1856 01111111111111111111011111111111\*

L1888 11101111111111111111111111111111\*

V0001 1X0100011N1H111LHLHN\*

V0002 1X0000111N0H111LLLLN\*

V0003 1X0000110N1H111LHLHN\*

V0004 101000111N1H111LHHHN\*

V0005 110000101N1H111LHHHN\*

V0006 110000001N1L110LHLHN\*

V0007 100010101N1H110HLLN\*

V0008 100000101N1L101HLLN\*

V0009 100000111N1L101LHLHN\*

V0010 110000101N1L101LHLHN\*

V0011 110000101N1L001HLLN\*

V0012 000000101N1L101HLLN\*

V0013 000000111N1L101LLLN\*

V0014 001111101N1L111LLLN\*

C5677\*

43C1

PAL16R6

PAL DESIGN SPECIFICATION

PAL S03 Disk & Tape Emulator @ 2K Revision A0

S03 - Next Address Control @ 2K Rev A0

Steve Blightman 10/15/86

MCLK CSD39 CSD38 /MRES 5KPIN6 6KPIN6 NC1 CSD10 CSD19 GND  
 OE NC2 /ZE PUP /FE S0 S1 /ENFLG NC3 VCC

ZE := MRES

/PUP := MRES + /CSD39 + /CSD38

FE := /MRES\*CSD38

/S0 := MRES + /CSD39 +  
 CSD39\*/CSD38\*CSD10\*/CSD19\*5KPIN6 +  
 CSD39\*/CSD38\*/CSD10\*/CSD19\*/5KPIN6 +  
 CSD39\*/CSD38\*CSD10\*CSD19\*6KPIN6 +  
 CSD39\*/CSD38\*/CSD10\*CSD19\*/6KPIN6

/S1 := MRES + CSD39 + /CSD38

ENFLG := /MRES\*/CSD39\*CSD19

FUNCTION TABLE

MCLK CSD39 CSD38 /MRES 5KPIN6 6KPIN6 CSD10 CSD19

/ZE PUP /FE S0 S1 /ENFLG

```

; M   C C / 5 6 C C   / P / S S /
; C   S S M K K S S   Z U F 0 1 E
; L   D D R P P D D   E P E           N
; K   3 3 E I I 1 1           F
;     9 8 S N N 0 9           L
;     6 6                       G

```

---

C	H	H	L	H	H	H	L	L	H	L	L	H	
C	L	H	L	L	L	L	L	L	H	L	L	H	
C	L	L	H	L	L	L	L	H	L	H	L	L	Continue
C	L	L	H	L	L	L	L	H	L	L	L	L	Continue, flag
C	L	H	H	L	L	L	L	H	L	H	H	H	RTS
C	L	H	H	L	L	L	H	H	L	L	H	L	RTS, flag
C	H	L	H	H	L	H	L	H	L	L	H	H	JCT, select 5K, no jump
C	H	L	H	L	H	H	L	H	L	H	L	H	JCT, select 5K, jump
C	H	L	H	L	H	H	H	H	L	L	H	H	JCT, select 6K, no jump
C	H	L	H	H	L	H	H	H	L	H	L	H	JCT, select 6K, jump
C	H	L	H	H	L	L	L	L	H	H	L	H	JCF, select 5K, jump
C	H	L	H	L	H	L	L	L	H	L	L	H	JCF, select 5K, no jump
C	H	L	H	L	H	L	H	L	H	H	L	H	JCF, select 6K, jump
C	H	L	H	H	L	L	H	L	H	L	L	H	JCF, select 6K, no jump
C	H	H	H	H	L	H	H	H	L	H	L	H	JSR

---

# DESCRIPTION

This PAL controls the 2911 next address generation.

CSD39	CSD38	CSD19	CSD10	Instruction
L	L	L	X	Continue
L	L	H	X	Continue, Enable flag
L	H	L	X	Return from subroutine
L	H	H	X	Return from subroutine, Enable flag
H	L	L	L	Jump if condition at mux 5K false
H	L	L	H	Jump if condition at mux 5K true
H	L	H	L	Jump if condition at mux 6K false
H	L	H	H	Jump if condition at mux 6K true
H	H	X	X	Jump to subroutine

S03 - Next Address Control @ 2K Rev A0

```

*****
*
MCLK      * 1*      P A L      *20*      VCC
*****
*
CSD39     * 2*      1 6 R6     *19*      NC3
*****
*
CSD38     * 3*      *          *18*      /ENFLG
*****
*
/MRES     * 4*      *          *17*      S1
*****
*
5KPIN6    * 5*      *          *16*      S0
*****
*
6KPIN6    * 6*      *          *15*      /FE
*****
*
NC1       * 7*      *          *14*      PUP
*****
*
CSD10     * 8*      *          *13*      /ZE
*****
*
CSD19     * 9*      *          *12*      NC2
*****
*
GND       *10*     *          *11*      OE
*****
*
*****

```

PAL16R6

PAL DESIGN SPECIFICATION

PAL S03 Disk & Tape Emulator @ 2K Revision A0

S03 - Next Address Control @ 2K Rev A0

Steve Blightman 10/15/86

\*

G0\*F0\*

L0256 101111101111111111111111111111110111\*

L0512 11111111011111111111111111111111111\*

L0544 01111111111111111111111111111111111\*

L0576 11111011111111111111111111111111111\*

L0768 11111111011111111111111111111111111\*

L0800 10111111111111111111111111111111111\*

L0832 0111101111110111111111111011110111\*

L0864 011110111111101111111111101110111\*

L0896 011110111111111101111111011101111\*

L0928 011110111111111110111111011101111\*

L1024 11110111011111111111111111111111111\*

L1280 11111111011111111111111111111111111\*

L1312 10111111111111111111111111111111111\*

L1344 11111011111111111111111111111111111\*

L1536 11111111011111111111111111111111111\*

V0001 C11011X11NXXLLHLLHXXN\*

V0002 C01000X00NXXLLHLLHXXN\*

V0003 C00100X00NXXHLHLLHXXN\*

V0004 C00100X01NXXHLHLLLXXN\*

V0005 C01100X00NXXHLLLHXXN\*

V0006 C01100X01NXXHLLLHLXXN\*

V0007 C10110X10NXXHLHLLHXXN\*

V0008 C10101X10NXXHLHHLHXXN\*

V0009 C10101X11NXXHLHLLHXXN\*

V0010 C10110X11NXXHLHHLHXXN\*

V0011 C10110X00NXXHLHHLHXXN\*

V0012 C10101X00NXXHLHLLHXXN\*

V0013 C10101X01NXXHLHHLHXXN\*

V0014 C10110X01NXXHLHLLHXXN\*

V0015 C11110X11NXXHLLHHLHXXN\*

C3A36\*

1A7E

PAL16L8

PAL DESIGN SPECIFICATION

PAL S03 Disk & Tape Emulator @ 2V Revision A0

S03 - Disk Cylinder & Head Address @ 2V Rev A0

Steve Blightman 10/15/86

/BUS6 /BUS3 /BUS2 /BUS1 /BUS0 /BUS4 /FLT /TAG1 /TAG2 GND  
/TAG3 /SSEEK /RESET /FAULT HADD0 CADD0 CADD1 CADD2 RORW VCC

/RORW = /TAG3 + /BUS0\*/BUS1 + FAULT + RESET

/CADD2 = TAG1\*/BUS2 + /CADD2\*/TAG1 + /CADD2\*/BUS2 + TAG3\*BUS6 + RESET

/CADD1 = TAG1\*/BUS1 + /CADD1\*/TAG1 + /CADD1\*/BUS1 + TAG3\*BUS6 + RESET

/CADD0 = TAG1\*/BUS0 + /CADD0\*/TAG1 + /CADD0\*/BUS0 + TAG3\*BUS6 + RESET

/HADD0 = TAG2\*/BUS0 + /HADD0\*/TAG2 + /HADD0\*/BUS0 + TAG3\*BUS6 + RESET

FAULT = FLT\*/RESET + FAULT\*/BUS4\*/RESET + FAULT\*/TAG3\*/RESET  
+ TAG3\*BUS0\*BUS1 + TAG3\*TAG1 + TAG3\*TAG2 + TAG1\*TAG2

SSEEK = TAG1 + TAG3\*BUS6 + TAG3\*BUS2 + TAG3\*BUS3

FUNCTION TABLE

;



S03 - Disk Cylinder & Head Address @ 2V Rev A0

```

*****
*
* *
*
/BUS6 * 1*          P A L          *20*   VCC
      * 1*          *
      * 1*          *
      *
      * 2*          1 6 L8        *
      * 2*          *
      * 2*          *
      *
/BUS3 * 2*          *
      * 2*          *
      * 2*          *
      *
      * 3*          *
      * 3*          *
      * 3*          *
      *
/BUS2 * 3*          *
      * 3*          *
      * 3*          *
      *
      * 4*          *
      * 4*          *
      * 4*          *
      *
/BUS1 * 4*          *
      * 4*          *
      * 4*          *
      *
      * 5*          *
      * 5*          *
      * 5*          *
      *
/BUS0 * 5*          *
      * 5*          *
      * 5*          *
      *
      * 6*          *
      * 6*          *
      * 6*          *
      *
/BUS4 * 6*          *
      * 6*          *
      * 6*          *
      *
      * 7*          *
      * 7*          *
      * 7*          *
      *
/FLT  * 7*          *
      * 7*          *
      * 7*          *
      *
      * 8*          *
      * 8*          *
      * 8*          *
      *
/TAG1 * 8*          *
      * 8*          *
      * 8*          *
      *
      * 9*          *
      * 9*          *
      * 9*          *
      *
/TAG2 * 9*          *
      * 9*          *
      * 9*          *
      *
      *
      * 10*         *
      * 10*         *
      * 10*         *
      *
GND   * 10*         *
      * 10*         *
      * 10*         *
      *
      *
      *
*****

```



V0013 111111111N1H1HLHHHLN\*  
V0014 111101111N1H1HLHHHLN\*  
V0015 111101110N1H1HHHHHLN\*  
V0016 111101111N1H1HHHHHLN\*  
V0017 011111111N0L1HLLLLLN\*  
V0018 110001100N1L1LHHHHHLN\*  
V0019 110001111N1H1LHHHHHLN\*  
V0020 111111111N1H0HLLLLLN\*  
V0021 010001100N0L1LLLLLLN\*  
V0022 110001100N1L0LLLLLLN\*  
V0023 111001111N0H0LLLLLLN\*  
V0024 111111101N0L0LLLLLLN\*  
V0025 11111110N0H0LLLLLLN\*  
V0026 111110011N0H1LLLLLLN\*  
V0027 111110011N1H1LLLLLLN\*  
V0028 111110111N1H1LLLLLLN\*  
V0029 111111111N1H1LLLLLLN\*  
V0030 111111111N0H1LLLLLLN\*  
V0031 111110111N0H1HLLLLLN\*  
C9C85\*  
4434

PLE11P4

PLE DESIGN SPECIFICATION

S03\_2XA0 Disk & Tape Emulator @ 2X Revision A0

SMD BUS DECODE PROM

Steve Blightman 10/15/86

.ADD /256K /B1 /B2 /B3 /B4 /B5 /B6 /B7 /B8 /B9 /B10  
.DAT ILLCYL ILLHD MULTIBIT

ILLCYL = B10 + B9 + B8 + B7 + B6 + B5 + B4 + B3 +  
/256K\*B2 + /256K\*B1

ILLHD = B10 + B9 + B8 + B7 + B6 + B5 + B4 + B3 + B2 + B1

MULTIBIT = B1\*B3 + B1\*B4 + B1\*B5 + B1\*B6 + B1\*B7 + B1\*B8 +  
B1\*B9 + B1\*B10 +  
B2\*B3 + B2\*B4 + B2\*B5 + B2\*B6 + B2\*B7 + B2\*B8 +  
B2\*B9 + B2\*B10 +  
B3\*B4 + B3\*B5 + B3\*B6 + B3\*B7 + B3\*B8 + B3\*B9 +  
B3\*B10 +  
B4\*B5 + B4\*B6 + B4\*B7 + B4\*B8 + B4\*B9 + B4\*B10 +  
B5\*B6 + B5\*B7 + B5\*B8 + B5\*B9 + B5\*B10 +  
B6\*B7 + B6\*B8 + B6\*B9 + B6\*B10 +  
B7\*B8 + B7\*B9 + B7\*B10 +  
B8\*B9 + B8\*B10 + B9\*B10

FUNCTION TABLE

256K	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	ILLCYL	ILLHD	MULTIBIT
H	L	L	L	L	L	L	L	L	L	L	L	L	L
X	L	L	L	L	L	L	L	L	L	H	H	H	L
X	L	L	L	L	L	L	L	L	H	L	H	H	L
X	L	L	L	L	L	L	H	L	L	L	H	H	L
X	L	L	L	L	L	H	L	L	L	L	H	H	L
X	L	L	L	L	H	L	L	L	L	L	H	H	L
X	L	L	L	H	L	L	L	L	L	L	H	H	L
X	L	L	H	L	L	L	L	L	L	L	H	H	L
H	L	H	L	L	L	L	L	L	L	L	L	H	L
H	H	L	L	L	L	L	L	L	L	L	L	H	L
L	H	L	L	L	L	L	L	L	L	L	H	H	L
L	L	H	L	L	L	L	L	L	L	L	H	H	L
H	H	H	L	L	L	L	L	L	L	L	L	H	L
H	L	H	H	L	L	L	L	L	L	L	H	H	H
H	L	L	H	H	L	L	L	L	L	L	H	H	H
H	L	L	L	H	H	L	L	L	L	L	H	H	H
H	L	L	L	L	H	H	L	L	L	L	H	H	H
H	L	L	L	L	L	L	H	H	L	L	H	H	H
H	L	L	L	L	L	L	L	H	H	L	H	H	H
H	L	L	L	L	L	L	L	L	H	H	H	H	H
H	L	L	L	L	L	L	L	L	H	H	H	H	H

## DESCRIPTION

This PROM decodes the SMD Bus bits for error conditions.

The valid head addresses are 0 or 1. Any other value will cause the ILLHD signal to be raised, which in turn will cause a FAULT condition to be flagged if TAG2 is set.

The valid cylinder addresses are 0 or 1 for 64K DRAMs, or 0-7 for 256K DRAMs. Any other value will cause the ILLCYL to be raised, which in turn will cause a SEEK ERROR condition to be flagged if TAG1 is set.

If any illegal cylinder address is sent with more than one bit set on the SMD bus, the MULTIBIT signal is raised. This will cause a FAULT condition, as well as the SEEK ERROR condition, to be flagged when TAG1 is set. This feature is meant to enable software to test each bus line in turn, issuing seeks to illegal cylinders and checking for SEEK ERROR. If FAULT is also returned this may indicate that two bus lines are shorted together.





07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 .  
07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 .  
07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 .  
07 07 07 07 07 07 07 07 07 07 07 07 07 03 03 .  
07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 .  
07 07 07 07 07 07 07 07 07 07 07 07 07 03 03 .  
07 07 07 07 07 07 07 07 07 07 07 07 07 03 03 .  
07 07 07 07 07 07 03 03 02 03 02 03 02 03 00 00 .

03797



ECCERR =

WC5\*WC6\*WC7\*/WC8\*WC9\*/WC10\*/WC11\*/WC12\*/WC13\*/SECSW2\*/SECSW1 +  
/WC5\*/WC6\*WC7\*WC8\*/WC9\*/WC10\*/WC11\*/WC12\*/WC13\*SECSW2\*/SECSW1 +  
WC5\*/WC6\*WC7\*/WC8\*WC9\*/WC10\*/WC11\*/WC12\*/WC13\*/SECSW2\*SECSW1

FUNCTION TABLE

SECSW1	SECSW2	WC13	WC12	WC11	WC10	WC9	WC8	WC7	WC6	WC5	ECCERR	SECTOR	INDEX
L	L	L	L	L	L	L	L	L	L	L	L	L	H
L	L	L	L	L	L	H	L	L	L	L	L	H	L
H	L	L	L	L	L	L	H	H	L	L	L	H	L
H	L	L	L	L	L	H	L	H	L	H	H	L	L

DESCRIPTION

This PROM controls the index & sector generation. There are 2 switches that control the number of sectors per track as follows

SECSW1	SECSW2	
L	L	32 sectors
L	H	64 sectors
H	L	35 sectors (last sector long)
H	H	Not used

These sector switches together with the Word Count Address, which represents where the heads are located with reference to Index, are used to generate the Index and Sector pulses. They are also used to generate the ECCERR signal, which will force a possible ECC error if the appropriate switches are on.





00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .

00110

```
1 00000 TITLE TAPE EMULATOR FIRMWARE VERSION S03 - A0
2 00000 ;
3 00000 ;S03TA0.MAC
4 00000 ;
5 00000 ;*****
6 00000 ;THESE WERE VERSIONS OF THE INITIAL TAPENORM FIRMWARE
7 00000 ;JULY 16,1984 VERSION IA1 INITIAL RELEASE
8 00000 ;OCT 17,1985 VERSION IA2 CHANGES FOR S34
9 00000 ;*****
10 00000 ;
11 00000 ;NOV 26,1986 VERSION A0 INITIAL RELEASE
12 00000 ;
13 00000 ;
14 00000 ;ALU REGISTER ASSIGNMENTS
15 00000 ;
16 00000 ;R0 LEAST SIGNIFICANT 12 BITS OF RAM ADDRESS
17 00000 ;R1 MOST SIGNIFICANT 4 BITS OF RAM ADDRESS
18 00000 ;R2 5 BIT COMMAND WORD - BITS 7-11
19 00000 ;R3 STROBE FREQUENCY
20 00000 ;R4 WAIT COUNTER
21 00000 ;R5 FIXED ERASE END LSB
22 00000 ;R6 FIXED ERASE END MSB
23 00000 ;R7 VARIABLE ERASE COUNTER
24 00000 ;R8 CONSTANT - H#002
25 00000 ;R9 CONSTANT - H#100
```

```

27 00000
28 00000 ;DIAGNOSTIC TEST 1 - PARTIAL DEST BUS AND QREG TEST
29 00000 ;
30 00000 300E080000 DIAG1: ALU CONST,,DTOR,,H#000 ;LOAD 0 INTO QREG
31 00001 9084020401 JCT ,,MOVQ,,DEST7,$ ;TEST FOR 0'S
32 00002 9084080402 JCT ,,MOVQ,,DEST8,$ ;
33 00003 9084090403 JCT ,,MOVQ,,DEST9,$ ;
34 00004 90840A0404 JCT ,,MOVQ,,DEST10,$ ;
35 00005 90840B0405 JCT ,,MOVQ,,DEST11,$ ;
36 00006 300E080FFF ALU CONST,,DTOR,,H#FFF ;LOAD F INTO QREG
37 00007 9084020007 JCF ,,MOVQ,,DEST7,$ ;TEST FOR 1'S
38 00008 9084080008 JCF ,,MOVQ,,DEST8,$ ;
39 00009 9084090009 JCF ,,MOVQ,,DEST9,$ ;
40 0000A 90840A000A JCF ,,MOVQ,,DEST10,$ ;
41 0000B 90840B000B JCF ,,MOVQ,,DEST11,$ ;
42 0000C ;
43 0000C ;DIAGNOSTIC TEST 2 - ALU FUNCTION TEST
44 0000C ;
45 0000C 31BE180555 DIAG2: ALU CONST,,DTOR,R1,H#555 ;
46 0000D 300A180AAA ALU CONST,,DPRQ,R1,H#AAA ;TEST PLUS
47 0000E 908907040E JCT ,,,,COUT,$ ;CARRY SHOULD BE LOW
48 0000F 1005080000 ALU ,,INCB ;INC B TO 000
49 00010 9084070010 JCF ,,MOVQ,,COUT,$ ;FAILED PLUS
50 00011 9080060011 JCF ,,,,EQUAL,$ ;EQUAL SHOULD BE HI
51 00012 302B180555 ALU CONST,,DMRQ,R1,H#555 ;TEST MINUS
52 00013 9080060013 JCF ,,,,EQUAL,$ ;FAILED MINUS
53 00014 304A180AAA ALU CONST,,DARQ,R1,H#AAA ;TEST AND
54 00015 9080060015 JCF ,,,,EQUAL,$ ;FAILED AND
55 00016 303A180AAA ALU CONST,,DORQ,R1,H#AAA ;TEST OR
56 00017 1005080000 ALU ,,INCB ;
57 00018 9080060018 JCF ,,,,EQUAL,$ ;FAILED OR
58 00019 128E180000 ALU ,,SRR,R1 ;TEST SHIFT RIGHT
59 0001A 302B1802AA ALU CONST,,DMRQ,R1,H#2AA ;
60 0001B 908006001B JCF ,,,,EQUAL,$ ;FAILED SHIFT RIGHT
61 0001C 138E180000 ALU ,,SLR,R1 ;TEST SHIFT LEFT
62 0001D 138E180000 ALU ,,SLR,R1 ;
63 0001E 302B180AAB ALU CONST,,DMRQ,R1,H#AAB ;
64 0001F 908006001F JCF ,,,,EQUAL,$ ;FAILED SHIFT LEFT
65 00020 ;
66 00020 ;DIAGNOSTIC TEST 3 - ALU REGISTER TEST
67 00020 ;
68 00020 31BE180124 DIAG3: ALU CONST,,DTOR,R1,H#124 ;
69 00021 31BE280218 ALU CONST,,DTOR,R2,H#218 ;
70 00022 31BE480481 ALU CONST,,DTOR,R4,H#481 ;
71 00023 31BE880842 ALU CONST,,DTOR,R8,H#842 ;
72 00024 1009180000 ALU ,,RP1Q,R1 ;R1+1+R2+R4+R8=0
73 00025 1000280000 ALU ,,QPRQ,R2 ;
74 00026 1000480000 ALU ,,QPRQ,R4 ;
75 00027 1000880000 ALU ,,QPRQ,R8 ;
76 00028 9089060028 JCF ,,,,EQUAL,$ ;FAIL ALU REG TEST

```

```

78 00029
79 00029 ;DIAGNOSTIC TEST 4 - RAM TEST
80 00029 ; WRITES LOWER 11 BITS OF RAM ADDRESS INTO LOCATION, READS BACK
81 00029 ; AND CHECKS PARITY.
82 00029 ;
83 00029 11C40B0000 DIAS4: ALU ,,ZERR,R0 ;R0=RAM ADDR LSB
84 0002A 1DC41B0000 ALU ,CA,ZERR,R1 ;R1=RAM ADDR MSB
85 0002B 14B800000E WRT: ALUF ,MEMWR,M0VR,R0,,SDE ;LATCH RAM DATA
86 0002C 10B800D00C ALUF ,,M0VR,R0,SALE,SW ;LOAD LSB ADDR
87 0002D 10B00B0000 ALU ;
88 0002E 10B0005004 ALUF ,,,,RALE,RW ;RESET WRITE & ALE
89 0002F 10B000A000 ALUF ,,,,SRFRQ ;REFRESH
90 00030 10B0002000 ALUF ,,,,RRFRQ ;
91 00031 10B0000006 ALUF ,,,,RDE ;RESET DATA ENABLE
92 00032 11B90B0000 ALU ,,INCR,R0 ;INC LSB ADDRESS
93 00033 90B007002B JCF ,,,,COUT,WRT ;LSB=FFF ?
94 00034 30EA1B000F ALU CONST,,DXR,R1,H#00F ;MSB=F ?
95 00035 90B0060437 JCT ,,,,EQUAL,READ ;IF YES, GO READ
96 00036 90B910002B JMP ,CA,INCR,R1,WRT ;INC MSB & CONTINUE
97 00037 11C40B0000 READ: ALU ,,ZERR,R0 ;R0=RAM ADDR LSB
98 00038 1DC41B0000 ALU ,CA,ZERR,R1 ;R1=RAM ADDR MSB
99 00039 31B7B07FF RD1: ALU CONST,,DTOR,R7,H#7FF ;SET MASK CONSTANT
100 0003A 10B00D0000 ALUF ,,RTOE,R0,SALE ;LOAD LSB ADDR
101 0003B 10407B0000 ALU ,,QARQ,R7 ;MASK OUT BIT 11
102 0003C 21C0705000 ALUF MEMRD,,DARR,R7,RALE ;LATCH DATA & MASK BIT11
103 0003D 10E07B0000 ALU ,,QXR,R7 ;COMPARE DATA & ADDRESS
104 0003E 90B0060047 JCF ,,,,EQUAL,RDERR ;MISCOMPARE ON READ
105 0003F 10B0004000 ALUF ,,,,SRFRQ ;REFRESH
106 00040 10B0002000 ALUF ,,,,RRFRQ ;
107 00041 90B0010441 JCT ,,,,PARERR,$ ;CHECK FOR PARITY ERROR
108 00042 11B90B0000 ALU ,,INCR,R0 ;INCREMENT ADDR LSB
109 00043 90B0070039 JCF ,,,,COUT,RD1 ;LSB=FFF ?
110 00044 30EA1B000F ALU CONST,,DXR,R1,H#00F ;MSB=F ?
111 00045 90B0060448 JCT ,,,,EQUAL,DIAS5 ;IF YES, FINISHED
112 00046 90B9100039 JMP ,CA,INCR,R1,RD1 ;INC MSB & CONTINUE
113 00047 90B8000047 RDERR: JMP ,,M0VR,R0,$ ;READ COMPARE ERROR
114 0004E ;
115 0004E ;DIAGNOSTIC TEST 5 - RAM PARITY TEST
116 0004E ; FORCES PARITY ERROR AND VERIFIES ITS OCCURENCE
117 0004E ;
118 0004E 10B00D000A DIAS5: ALUF ,,M0VR,R0,SALE,SPT ;SET PARITY TEST BIT
119 0004F 10B0090000 ALU ;
120 0004A 20B0005000 ALUF MEMRD,RALE ;RESET ALE
121 0004E 90B001004B JCF ,,,,PARERR,$ ;SHOULD GET PARITY ERROR
122 0004C 10B0000002 ALUF ,,,,RPT ;RESET PARITY TEST BIT

```

```

124 0004D ;
125 0004D ;DIAGNOSTICS COMPLETE
126 0004D ;
127 0004D 108000000D DGEND: ALUF ,,,,SRDBAD ;RESET DIAG FAIL LED
128 0004E 11C4000005 ALUF ,,ZERR,RO,,RRDBAD ;ADDR=0
129 0004F 11C410F00F ALUF ,,ZERR,R1,SLDP,SDGOK ;SET BOT & DIAG PASS LED
130 00050 ;
131 00050 ;
132 00050 ;
133 00050 ;IDLE LOOP
134 00050 ;
135 00050 18C400A000 IDLE1: ALUF ,DOUT,ZERO,,SRFRQ ;REFRESH
136 00051 1080002000 ALUF ,,,,RRFRQ ;
137 00052 9080041055 JCFF ,,,,RWD,RRS,IDLE2 ;JUMP IF NOT REWIND
138 00053 11C4006000 ALUF ,,ZERR,RO,REOT ;ADDR=0
139 00054 11C410F000 ALUF ,,ZERR,R1,SLDP ;AT BOT
140 00055 9080050457 IDLE2: JCT ,,,,FBSY,CMD1 ;JUMP IF COMMAND
141 00056 9080000050 JMP ,,,,IDLE1 ;

```

```

143 00057 ;
144 00057 ;COMMAND DECODE
145 00057 ; THE TWELVE VALID TAPE COMMANDS ARE DECODED BELOW:
146 00057 ;
147 00057 ;
148 00057 ;DESTINATION BIT (REV) (WRT) (WFM) (EDT) (ERA)
149 00057 ;-----
150 00057 ;READ FORWARD 0 0 0 0 0
151 00057 ;READ REVERSE 1 0 0 0 0
152 00057 ;READ REV EDIT 1 0 0 1 0
153 00057 ;WRITE 0 1 0 0 0
154 00057 ;WRITE EDIT 0 1 0 1 0
155 00057 ;WRITE FILEMARK 0 1 1 0 0
156 00057 ;ERASE VARIABLE 0 1 0 0 1
157 00057 ;ERASE FIXED 0 1 1 0 1
158 00057 ;SPACE FORWARD 0 0 0 0 1
159 00057 ;SPACE REVERSE 1 0 0 0 1
160 00057 ;FILE SEARCH FWD 0 0 1 0 0(1)
161 00057 ;FILE SEARCH REV 1 0 1 0 0(1)
162 00057 ;
163 00057 118E280000 CMD1: ALU RDCMD, DTOR, R2 ;CMD BITS TO R2
164 00058 90882A0464 JCT ,,MOVR, R2, DEST10, CMD4 ;JUMP IF WRT BIT TRUE
165 00059 908828046C JCT ,,MOVR, R2, DEST8, CMD6 ;JUMP IF EDIT BIT TRUE
166 0005A 9088290461 JCT ,,MOVR, R2, DEST9, CMD3 ;JUMP IF WFMK BIT TRUE
167 0005B 90882B045F JCT ,,MOVR, R2, DEST11, CMD2 ;JUMP IF REV BIT TRUE
168 0005C 908822054E JCT ,,MOVR, R2, DEST7, SPFWD ;SPACE FORWARD
169 0005D 9088220089 JCF ,,MOVR, R2, DEST7, RDFWD ;READ FORWARD
170 0005E 9088200070 JMP ,,MOVR, R2, ILLCMD ;ILLEGAL COMMAND
171 0005F 9088220573 CMD2: JCT ,,MOVR, R2, DEST7, SPREV ;SPACE REVERSE
172 00060 90882000B3 JMP ,,MOVR, R2, RDREV ;READ REVERSE
173 00061 90882B05C5 CMD3: JCT ,,MOVR, R2, DEST11, FSREV ;FILE SEARCH REVERSE
174 00062 90882001A0 JMP ,,MOVR, R2, FSFWD ;FILE SEARCH FORWARD
175 00063 9088200070 JMP ,,MOVR, R2, ILLCMD ;ILLEGAL COMMAND
176 00064 908828046C CMD4: JCT ,,MOVR, R2, DEST8, CMD6 ;JUMP IF EDIT BIT TRUE
177 00065 9088290469 JCT ,,MOVR, R2, DEST9, CMD5 ;JUMP IF WFMK BIT TRUE
178 00066 9088220521 JCT ,,MOVR, R2, DEST7, ERVAR ;VARIABLE LENGTH ERASE
179 00067 90882B0072 JCF ,,MOVR, R2, DEST11, WRITE ;WRITE
180 00068 9088200070 JMP ,,MOVR, R2, ILLCMD ;ILLEGAL COMMAND
181 00069 9088220504 CMD5: JCT ,,MOVR, R2, DEST7, ERFIX ;FIXED LENGTH ERASE
182 0006A 90882B00F4 JCF ,,MOVR, R2, DEST11, WFMK ;WRITE FILE MARK
183 0006B 9088200070 JMP ,,MOVR, R2, ILLCMD ;ILLEGAL COMMAND
184 0006C 9088220470 CMD6: JCT ,,MOVR, R2, DEST7, ILLCMD ;JUMP IF ERASE BIT TRUE
185 0006D 9088290470 JCT ,,MOVR, R2, DEST9, ILLCMD ;JUMP IF WFMK BIT TRUE
186 0006E 90882B04B3 JCT ,,MOVR, R2, DEST11, RDREV ;READ REVERSE (EDIT)
187 0006F 90882A0472 JCT ,,MOVR, R2, DEST10, WRITE ;WRITE EDIT (SAME AS WRITE)
188 00070 1088200009 ILLCMD: ALUF ,,MOVR, R2, SHER ;SET HARD ERROR
189 00071 9088200050 JMP ,,MOVR, R2, IDLE1 ;DUE TO ILLEGAL COMMAND

```

```

191 00072 ;
192 00072 ;WRITE COMMAND
193 00072 ; WRITES LWD BEFORE RECORD AND COINCIDENT WITH LAST BYTE OF RECORD
194 00072 ;
195 00072 D1C4400202 WRITE: JSR ,,ZERR,R4,WAIT1 ;WAIT 800 US
196 00073 D0800001F2 JSR ,,,,SFREQ ;GET STROBE FREQUENCY
197 00074 31BE980100 ALU CONST,,DTOR,R9,H#100 ;LWD CONSTANT
198 00075 1C88107000 ALUF ,CA,MOVR,R1,RLDP ;LOAD MSB ADDR
199 00076 148890000E ALUF ,MEMWR,MOVR,R9,,SDE ;LATCH & ENABLE DATA
200 00077 108800D00C ALUF ,,MOVR,R0,SALE,SW ;SET ALE & WRITE
201 00078 90800105DC JCT ,,,,PARERR,PERR ;PARITY ERROR?
202 00079 1080005004 ALUF ,,,,RALE,RW ;END WRITE CYCLE
203 0007A 118900A000 ALUF ,,INCR,R0,SFRQ ;INCREMENT LSB
204 0007B 908007207E JCFF ,,,,COUT,RRFRQ,WR1 ;WAS LSB=FFF ?
205 0007C D0800001FB JSR ,,,,ADC ;ADDRESS CHECK
206 0007D 11891B0000 ALU ,,INCR,R1 ;INC MSB
207 0007E 1C8810C000 WR1: ALUF ,CA,MOVR,R1,SWS ;SET WRITE STROBE
208 0007F 1008300006 ALUF ,,RTOB,R3,,RDE ;LOAD MSB ADDR
209 00080 9014064080 JCFF ,,DECB,,EQUAL,RWS,$ ;DELAY LOOP
210 00081 040E00000E ALUF DATIN,MEMWR,DTOB,,SDE ;LATCH & ENABLE DATA
211 00082 108800D00C ALUF ,,MOVR,R0,SALE,SW ;SET ALE & WRITE
212 00083 90800105DC JCT ,,,,PARERR,PERR ;PARITY ERROR?
213 00084 1084005004 ALUF ,,MOVE,,RALE,RW ;END WRITE CYCLE
214 00085 918908A5EC JCTF ,,INCR,R0,DESTB,SFRQ,WREND ;CHECK FOR LWD
215 00086 908007207E JCFF ,,,,COUT,RRFRQ,WR1 ;WAS LSB=FFF ?
216 00087 D0800001FB JSR ,,,,ADC ;ADDRESS CHECK
217 00088 918910007E JMP ,,INCR,R1,WR1 ;INC MSB

```

```

219 000B9 ;
220 000B9 ;READ COMMAND
221 000B9 ;
222 000B9 ; IF NO LWD IN FIRST BYTE OF RECORD, INCREMENTS POINTER UNTIL
223 000B9 ; NEXT LWD AND READS THAT RECORD.
224 000B9 DCBB1071F2 RFDND: JSRF ,CA,MOVR,R1,RLDP,SFREQ ;GET STROBE FREQUENCY
225 000BA 108B00D000 ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
226 000BB 10800B0000 ALU ;
227 000BC 280E005000 ALUF MEMRD,DOUT,DTQ,,RALE ;READ DATA OUT
228 000BD 90800B00A1 JCF ,,,DESB,RDF5 ;JUMP IF NO LWD
229 000BE 108000A000 ALUF ,,,SRFRQ ;REFRESH
230 000BF 90800125DC JCTF ,,,PARERR,RRFRQ,PERR ;PARITY ERROR?
231 00090 1189080000 ALU ,,INCR,RO ;INCREMENT LSB
232 00091 9008370095 JCF ,,RTOQ,R3,COUT,RDF2 ;WAS LSB=FFF ?
233 00092 D0800001FB JSR ,,,ADC ;ADDRESS CHECK
234 00093 1D89180000 ALU ,CA,INCR,R1 ;INC MSB
235 00094 1008380000 ALU ,,RTOQ,R3 ;DELAY VALUE
236 00095 9014060095 RDF2: JCF ,,DECR,EQUAL,$ ;DELAY LOOP
237 00096 108B00D000 ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
238 00097 10800B0000 ALU ;
239 00098 280E005000 ALUF MEMRD,DOUT,DTQ,,RALE ;READ DATA OUT
240 00099 908401A5DC JCTF ,,MOVQ,,PARERR,SRFRQ,PERR ;PARITY ERROR?
241 0009A 90840A24AB JCTF ,,MOVQ,,DESB,RRFRQ,RDFE4 ;CHECK FMK
242 0009B 90800B94AF JCTF ,,,DESB,SRS,RDFE5 ;CHECK LWD
243 0009C 1189080000 ALU ,,INCR,RO ;INCREMENT LSB
244 0009D 9008371095 JCF ,,,RTOQ,R3,COUT,RRS,RDF2 ;LSB=FFF ?
245 0009E D0800001FB JSR ,,,ADC ;ADDRESS CHECK
246 0009F 1D89180000 ALU ,CA,INCR,R1 ;INC MSB
247 000A0 9008300095 JMP ,,RTOQ,R3,RDF2 ;CONTINUE
248 000A1 108B00D000 RDF5: ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
249 000A2 10800B0000 ALU ;
250 000A3 280E005000 ALUF MEMRD,DOUT,DTQ,,RALE ;READ DATA OUT
251 000A4 108000A000 ALUF ,,,SRFRQ ;REFRESH
252 000A5 1080002000 ALUF ,,,RRFRQ ;
253 000A6 90840105DC JCT ,,,MOVQ,,PARERR,PERR ;PARITY ERROR?
254 000A7 91890B0495 JCT ,,INCR,RO,DESB,RDF2 ;CHECK LWD
255 000A8 90800700A1 JCF ,,,COUT,RDF5 ;WAS LSB=FFF ?
256 000A9 D0800001FB JSR ,,,ADC ;ADDRESS CHECK
257 000AA 9D891000A1 JMP ,CA,INCR,R1,RDF5 ;INC MSB
258 000AB 208E080000 RDFE4: ALU MEMRD,,MOVD ;
259 000AC 90800B00AF JCF ,,,DESB,RDFE5 ;JUMP IF PE
260 000AD 388E080013 ALU CONST,DOUT,MOVD,,H#013 ;EOF CODE (NRZ)
261 000AE 10800B0009 ALU ,,,SRS ;READ STROBE
262 000AF 1189080000 RDFE5: ALU ,,INCR,RO ;INCREMENT LSB
263 000B0 90800711E3 JCF ,,,COUT,RRS,RDEND ;WAS LSB=FFF ?
264 000B1 D0800001FB JSR ,,,ADC ;ADDRESS CHECK
265 000B2 9D891001E3 JMP ,CA,INCR,R1,RDEND ;INC MSB

```

```

267 000B3 ;
268 000B3 ;READ REVERSE
269 000B3 ;
270 000B3 ; CHECKS TO MAKE SURE POINTER NOT AT LDP BEFORE DECREMENTING POINTER.
271 000B3 ; IF NO LWD IN FIRST BYTE READ, DECREMENTS POINTER UNTIL NEXT LWD
272 000B3 ; AND REVERSE READS THE RECORD.
273 000B3 108B080000 RDREV: ALU , ,MOVR,RO ;LSB=0 ?
274 000B4 9C881600B7 JCF ,CA,MOVR,R1,EQUAL,RDR0 ;IF NOT, START RD REV
275 000B5 90800605DB JCT , , ,EQUAL,BERR ;IF MSB=0 GOTO BOT ERROR
276 000B6 1D98180000 ALU ,CA,DECR,R1 ;DECREMENT MSB
277 000B7 1198080000 RDR0: ALU , ,DECR,RO ;DECREMENT LSB
278 000B8 108800D000 RDR1: ALUF , ,MOVR,RO,SALE ;LOAD LSB ADDRESS
279 000B9 1080080000 ALU ;
280 000BA 280E005000 ALUF MEMRD,DOUT,DT0Q, ,RALE ;READ DATA OUT
281 000BB 90840800DB JCF , ,MOVE, ,DEST8,RDR5 ;JUMP IF NO LWD
282 000BC 90800AA4E9 RDR8: JCTF , , ,DEST10,SFRQ,RR4END ;CHECK FILEMARK
283 000BD 90880195DC JCTF , ,MOVR,RO,PARERR,SRS,PERR ;PARITY ERROR?
284 000BE 90881620C3 JCF , ,MOVR,R1,EQUAL,RRFRQ,RDR3 ;IF LSB<>0, KEEP READING
285 000BF 90800610C2 JCF , , ,EQUAL,RRS,RDR7 ;IF MSB<>0, KEEP READING
286 000C0 90800C00C2 JCF , , ,ENB0T,RDR7 ;CHECK ENB0T SW
287 000C1 90800001E3 JMP , , ,RDEND ;END READ
288 000C2 1D98180000 RDR7: ALU ,CA,DECR,R1 ;DEC MSB
289 000C3 D1980001F2 RDR3: JSR , ,DECR,RO,SFRQ ;GET STROBE FREQUENCY
290 000C4 90140600C4 RDR2: JCF , ,DECR, ,EQUAL, $ ;DELAY LOOP
291 000C5 108800D000 ALUF , ,MOVR,RO,SALE ;LOAD LSB ADDR
292 000C6 1080001000 ALUF , , ,RRS ;RESET READ STROBE
293 000C7 280E005000 ALUF MEMRD,DOUT,DT0Q, ,RALE ;READ DATA OUT
294 000C8 908408A5E3 JCTF , ,MOVE, ,DEST8,SFRQ,RDEND ;CHECK LWD
295 000C9 90800A24E9 JCTF , , ,DEST10,RRFRQ,RR4END ;CHECK FILEMARK
296 000CA 90800195DC JCTF , , ,PARERR,SRS,PERR ;PARITY ERROR?
297 000CB 1198080000 ALU , ,DECR,RO ;DECREMENT LSB
298 000CC 90083714C4 JCTF , ,RTOQ,R3,COU,RRS,RDR2 ;WAS LSB=0 ?
299 000CD 108B180000 ALU , ,MOVR,R1 ;MSB=0?
300 000CE 90800604D4 JCT , , ,EQUAL,RREND ;IF YES, GOTO RREND
301 000CF 30EA18000C ALU CONST, ,DXR,R1,H#00C ;MSB=C ?
302 000D0 90800600D2 JCF , , ,EQUAL,RDR4 ;IF NOT, JUMP
303 000D1 1080006000 ALUF , , ,REDT ;ELSE, RESET EDT
304 000D2 1D98180000 RDR4: ALU ,CA,DECR,R1 ;DEC MSB
305 000D3 90083000C4 JMP , ,RTOQ,R3,RDR2 ;CONTINUE
306 000D4 90800C04D7 RREND: JCT , , ,ENB0T,RR1END ;CHECK EDT DISABLE SW
307 000D5 1D98180000 ALU ,CA,DECR,R1 ;DECREMENT MSB
308 000D6 90083000C4 JMP , ,RTOQ,R3,RDR2 ;CONTINUE
309 000D7 91890001E3 RR1END: JMP , ,INCR,RO,RDEND ;INCREMENT LSB
310 000D8 108800D000 RDR5: ALUF , ,MOVR,RO,SALE ;LOAD LSB ADDR
311 000D9 1080060000 ALU ;
312 000DA 280E005000 ALUF MEMRD,DOUT,DT0Q, ,RALE ;READ DATA OUT
313 000DB 108000A000 ALUF , , ,SFRQ ;REFRESH
314 000DC 1084002000 ALUF , ,MOVE, ,RRFRQ ;
315 000DD 90840A04E9 JCT , ,MOVE, ,DEST10,RR4END ;CHECK FILEMARK
316 000DE 90840804BC JCT , ,MOVE, ,DEST8,RDR8 ;CHECK LWD
317 000DF 90800105DC JCT , , ,PARERR,PERR ;PARITY ERROR?
318 000E0 1198080000 ALU , ,DECR,RO ;DECREMENT LSB
319 000E1 90881704DB JCT , ,MOVR,R1,COU,RDR5 ;WAS LSB=0 ?

```

320	000E2	90800604E7	JCT	,,,EQUAL,RR5END	;MSB=0 ?
321	000E3	30EA18000C	ALU	CONST,,DXR,R1,H#00C	;MSB=C ?
322	000E4	90800600E6	JCF	,,,EQUAL,RDR6	;IF NOT, JUMP
323	000E5	1080060000	ALUF	,,,REOT	;ELSE, RESET EOT
324	000E6	90981000DB	RDR6: JMP	,CA,DECR,R1,RDR5	;DEC MSB
325	000E7	90800C00F3	RR5END: JCF	,,,ENBEDT,RR7END	;CHECK EOT DISABLE SW
326	000E8	918900F1E3	JMPF	,,INCR,RO,SLDP,RDEND	;DECREMENT LSB
327	000E9	208E002000	RR4END: ALUF	MEMRD,,MOVD,,RRFRQ	;
328	000EA	90800B00ED	JCF	,,,DEST11,RR6END	;JUMP IF PE
329	000EB	388E080013	ALU	CONST,DOUT,MOVD,,H#013	;EOF CODE (NRZ)
330	000EC	1080080009	ALU	,,,SRS	;READ STROBE
331	000ED	1198002000	RR6END: ALUF	,,DECR,RO,RRFRQ	;DECREMENT LSB
332	000EE	90681715E3	JCTF	,,MOVR,R1,CDUT,RRS,RDEND	;WAS LSB=0 ?
333	000EF	90800604E7	JCT	,,,EQUAL,RR5END	;MSB=0 ?
334	000F0	30EA18000C	ALU	CONST,,DXR,R1,H#00C	;MSB=C ?
335	000F1	90800600F3	JCF	,,,EQUAL,RR7END	;IF NOT, JUMP
336	000F2	1080060000	ALUF	,,,REOT	;ELSE, RESET EOT
337	000F3	91981000BB	RR7END: JMP	,,DECR,R1,RDR1	;DECREMENT MSB & CONTINUE

```

339 000F4 ;
340 000F4 ;WRITE FILEMARK
341 000F4 ;
342 000F4 1C88107000 WFMK: ALUF ,CA,MOVR,R1,RLDP ;LOAD MSB ADDR
343 000F5 340E080100 ALU CONST,MEMWR,DT00,,H#100 ;SET LWD
344 000F6 188400000E ALUF ,DOUT,MOVQ,,,SDE ;FILEMARK TO CONTROLLER
345 000F7 108800D00C ALUF ,,MOVR,R0,SALE,SW ;LOAD LSB ADDR
346 000FB 1080080000 ALU ;
347 000F9 1080005004 ALUF ,,,,RALE,RW ;RESET WRITE & ALE
348 000FA 1189080000 ALU ,,INCR,R0 ;INCREMENT LSB
349 000FB 90800700FE JCF ,,,,COUT,WFM1 ;WAS LSB=0?
350 000FC D0800001FB JSR ,,,,ADC ;ADDRESS CHECK
351 000FD 1D89180000 ALU ,CA,INCR,R1 ;LOAD MSB ADDR
352 000FE 340E080500 WFM1: ALU CONST,MEMWR,DT00,,H#500 ;SET FILEMARK & LWD
353 000FF 188400000E ALUF ,DOUT,MOVQ,,,SDE ;FILEMARK TO CONTROLLER
354 00100 108800D00C ALUF ,,MOVR,R0,SALE,SW ;LOAD LSB ADDR
355 00101 1080080000 ALU ;
356 00102 1080005004 ALUF ,,,,RALE,RW ;RESET WRITE & ALE
357 00103 91890001EC JMPF ,,INCR,R0,,WREND ;INCREMENT LSB
358 00104 ;
359 00104 ;FIXED LENGTH ERASE (3.75 INCHES)
360 00104 ;
361 00104 1C88107000 ERFIX: ALUF ,CA,MOVR,R1,RLDP ;LOAD MSB ADDR
362 00105 208E080000 ALU MEMRD,,MOVQ ;NRZ OR PE ?
363 00106 91C46B010D JCF ,,ZERR,R6,DEST11,ERF1 ;JUMP IF PE
364 00107 300A080BBB ALU CONST,,DPRQ,R0,H#BBB ;IF NRZ, ERA LSB=BBBH
365 00108 918457010A JCF ,,BTOR,R5,COUT,ERF ;NO MSB INC IF NO CARRY
366 00109 1189680000 ALU ,,INCR,R6 ;INC ERA MSB
367 0010A 10086B0000 ERF: ALU ,,RTOR,R6 ;ERA MSB
368 0010B 1000180000 ALU ,,BPRQ,R1 ;ERA MSB + MSB ADR
369 0010C 9184600114 JMP ,,BTOR,R6,ERF2 ;ERA MSB INTO R6
370 0010D 1189680000 ERF1: ALU ,,INCR,R6 ;PUT 1 INTO ERA MSB
371 0010E 300A080770 ALU CONST,,DPRQ,R0,H#770 ;IF PE, ERA LSB=770H
372 0010F 9184570111 JCF ,,BTOR,R5,COUT,ERF3 ;NO MSB INC IF NO CARRY
373 00110 1189680000 ALU ,,INCR,R6 ;INC ERA MSB
374 00111 10086B0000 ERF3: ALU ,,RTOR,R6 ;ERA MSB
375 00112 1000180000 ALU ,,BPRQ,R1 ;ERA MSB + MSB ADR
376 00113 1184680000 ALU ,,BTOR,R6 ;ERA MSB INTO R6
377 00114 14C400000E ERF2: ALUF ,MEMWR,ZERO,,,SDE ;ZER0ES INTO RAM
378 00115 108800D00C ALUF ,,MOVR,R0,SALE,SW ;LOAD LSB ADDR
379 00116 1080080000 ALU ;
380 00117 1080005004 ALUF ,,,,RALE,RW ;RESET WRITE & ALE
381 00118 118900A006 ALUF ,,INCR,R0,SRFRQ,RDE ;INC LSB
382 00119 900857211C JCFF ,,RTOR,R5,COUT,RRFRQ,ERF5 ;WAS LSB=FFF ?
383 0011A D0800001FB JSR ,,,,ADC ;ADDRESS CHECK
384 0011B 1D89180000 ALU ,CA,INCR,R1 ;INC MSB ADR
385 0011C 1091080000 ERF5: ALU ,,QMR,R0 ;ADR LSB=ERA LSB?
386 0011D 9008660114 JCF ,,RTOR,R6,EQUAL,ERF2 ;IF NOT, KEEP ERASING
387 0011E 1091180000 ALU ,,QMR,R1 ;ADR MSB=ERA MSB?
388 0011F 9080060114 JCF ,,,,EQUAL,ERF2 ;IF NOT, KEEP ERASING
389 00120 90800001EF JMPF ,,,,WTEWD ;END ERASE

```

```

391 00121 ;
392 00121 ;VARIABLE LENGTH ERASE
393 00121 ; ZER0 LENGTH ERASE ERASES 190H BYTES.
394 00121 ; ERASE LENGTH > 0 ERASES MAXIMUM OF 2000H BYTES.
395 00121 ;
396 00121 D1C4407202 ERVAR: JSRF ,,ZERR,R4,RLDP,WAIT1 ;WAIT 800 US
397 00122 D1C47001F2 JSR ,,ZERR,R7,SFREQ ;GET STROBE FREQUENCY
398 00123 1C8B10C000 ALUF ,CA,MOVR,R1,SWS ;SET WRITE STROBE
399 00124 11C4A80000 ALU ,,ZERR,R10 ;
400 00125 10B0004000 ALUF ,,,,RWS ;
401 00126 14C400000E ALUF ,MEMWR,ZERO,,SDE ;ZER0ES INTO WR LATCH
402 00127 10B800D00C ALUF ,,MOVR,RO,SALE,SW ;ZER0ES INTO RAM
403 00128 10B0080000 ALU ;
404 00129 008E005004 ALUF DATIN,,MOVD,,RALE,RW ;RESET WRITE & ALE
405 0012A 918908A53D JCTF ,,INCR,RO,DESTB,SRFRQ,ERV5 ;CHECK FOR LWD
406 0012B 90B007212E JCFE ,,,,COUT,RRFRQ,ERV1 ;WAS LSB=FFF ?
407 0012C D0800001FB JSR ,,,,ADC ;ADDRESS CHECK
408 0012D 1189180000 ALU ,,INCR,R1 ;INCREMENT MSB
409 0012E 1C8B10C000 ERV1: ALUF ,CA,MOVR,R1,SWS ;SET WRITE STROBE
410 0012F 100B307006 ALUF ,,RTOQ,R3,RLDP,RDE ;LOAD MSB ADDR
411 00130 9014064130 JCFE ,,DECC,,EQUAL,RWS,$ ;DELAY LOOP
412 00131 14C400000E ALUF ,MEMWR,ZERO,,SDE ;ZER0ES INTO WR LATCH
413 00132 10B800D00C ALUF ,,MOVR,RO,SALE,SW ;ZER0ES INTO RAM
414 00133 10B0080000 ALU ;
415 00134 008E005004 ALUF DATIN,,MOVD,,RALE,RW ;RESET WRITE & ALE
416 00135 918908A5EC JCTF ,,INCR,RO,DESTB,SRFRQ,WREND ;CHECK FOR LWD
417 00136 90B007212E JCFE ,,,,COUT,RRFRQ,ERV1 ;WAS LSB=FFF ?
418 00137 D0B00001FB JSR ,,,,ADC ;ADDRESS CHECK
419 00138 30EA780001 ALU CONST,,DYE,R7,H#001 ;SECOND TIME THRU LOOP?
420 00139 90B006053C JCT ,,,,EQUAL,ERV2 ;IF YES, STOP ERASING
421 0013A 1189780000 ALU ,,INCR,R7 ;INCREMENT COUNT
422 0013B 918910012E JMP ,,INCR,R1,ERV1 ;INC MSB
423 0013C 91891001EF ERV2: JMP ,,INCR,R1,WTEND ;INCREMENT MSB & JMP
424 0013D 30EA780190 ERV5: ALU CONST,,DYE,R7,H#190 ;400 TIMES?
425 0013E 90B006054C JCT ,,,,EQUAL,ERV3 ;IF YES, STOP ERASING
426 0013F 1189780000 ALU ,,INCR,R7 ;INCREMENT COUNT
427 00140 1C8B10C000 ERV6: ALUF ,CA,MOVR,R1,SWS ;SET WRITE STROBE
428 00141 100B300006 ALUF ,,RTOQ,R3,,RDE ;LOAD MSB ADDR
429 00142 9014064142 JCFE ,,DECC,,EQUAL,RWS,$ ;DELAY LOOP
430 00143 148BA0000E ALUF ,MEMWR,MOVR,R10,,SDE ;R10 INTO WR LATCH
431 00144 10B800D00C ALUF ,,MOVR,RO,SALE,SW ;R10 INTO RAM
432 00145 10B0080000 ALU ;
433 00146 108BA05004 ALUF ,,MOVR,R10,RALE,RW ;RESET WRITE & ALE
434 00147 90B008A5EF JCTF ,,,,DESTB,SRFRQ,WTEND ;CHECK FOR LWD
435 00148 1189080000 ALU ,,INCR,RO ;INCREMENT LSB
436 00149 90B007213D JCFE ,,,,COUT,RRFRQ,ERV5 ;WAS LSB=FFF ?
437 0014A D0800001FB JSR ,,,,ADC ;ADDRESS CHECK
438 0014B 918910013D JMP ,,INCR,R1,ERV5 ;INC MSB
439 0014C 318EA80100 ERV3: ALU CONST,,DYE,R10,H#100 ;SET LWD
440 0014D 90B0000140 JMP ,,,,ERV6 ;ONE LAST TIME

```

```

442 0014E
443 0014E ;SPACE FORWARD
444 0014E ; IF NO LWD WITH FIRST BYTE, SKIP TO NEXT LWD AND SPACE TO THE
445 0014E ; END OF THAT RECORD.
446 0014E ;
447 0014E 1C8B107000 SPFFND: ALUF ,CA,MOVR,R1,RLDP ;LOAD MSB ADDR
448 0014F 108B00D000 ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
449 00150 1080080000 ALU ;
450 00151 2B0E005000 ALUF MEMRD,DOUT,DTOR,,RALE ;READ DATA
451 00152 108400A000 ALUF ,,MOVQ,,SRFRQ ;REFRESH
452 00153 90800125DC JCTF ,,,PARERR,RRFRQ,PERR ;PARITY ERROR?
453 00154 1084080000 ALU ,,MOVE ;
454 00155 9189080163 JCF ,,INCR,RO,DEST8,SPF5 ;JUMP IF NO LWD
455 00156 9080070159 JCF ,,,COUT,SPF2 ;LSB=FFF ?
456 00157 D0800001FB JSR ,,,ADC ;ADDRESS CHECK
457 00158 1D89180000 ALU ,CA,INCR,R1 ;INC MSB
458 00159 108B00D000 SPFF2: ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
459 0015A 1080080000 ALU ;
460 0015B 2B0E005000 ALUF MEMRD,DOUT,DTOR,,RALE ;READ DATA
461 0015C 108400A000 ALUF ,,MOVQ,,SRFRQ ;REFRESH
462 0015D 90800125DC JCTF ,,,PARERR,RRFRQ,PERR ;PARITY ERROR?
463 0015E 1084080000 ALU ,,MOVE ;
464 0015F 91890805EC JCT ,,INCR,RO,DEST8,WREND ;CHECK LWD
465 00160 9080070159 JCF ,,,COUT,SPF2 ;LSB=FFF ?
466 00161 D0800001FB JSR ,,,ADC ;ADDRESS CHECK
467 00162 9DB9100159 JMP ,CA,INCR,R1,SPF2 ;INC MSB & CONTINUE
468 00163 108B00D000 SPFF5: ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
469 00164 1080080000 ALU ;
470 00165 2B0E005000 ALUF MEMRD,DOUT,DTOR,,RALE ;READ DATA
471 00166 108000A000 ALUF ,,,SRFRQ ;REFRESH
472 00167 1084002000 ALUF ,,MOVQ,,RRFRQ ;
473 00168 90800105DC JCT ,,,PARERR,PERR ;PARITY ERROR?
474 00169 1084080000 ALU ,,MOVE ;
475 0016A 9189080559 JCT ,,INCR,RO,DEST8,SPF2 ;CHECK LWD
476 0016B 9080070163 JCF ,,,COUT,SPF5 ;LSB=FFF ?
477 0016C D0800001FB JSR ,,,ADC ;ADDRESS CHECK
478 0016D 9DB9100163 SPFF4: JMP ,CA,INCR,R1,SPF5 ;INC MSB
479 0016E 9080070171 SPFFND: JCF ,,,COUT,SPF8 ;LSB=FFF ?
480 0016F D0800001FB JSR ,,,ADC ;ADDRESS CHECK
481 00170 1D89180000 ALU ,CA,INCR,R1 ;INC MSB
482 00171 108000B000 SPFF8: ALUF ,,,RBSY ;RESET FBSY
483 00172 9080003050 JMFF ,,,RRBSY,IDLE1 ;END SPACE FWD

```

```

485 00173 ;
486 00173 ;SPACE REVERSE
487 00173 ;
488 00173 ; CHECKS TO MAKE SURE POINTER NOT AT LDP BEFORE DECREMENTING POINTER.
489 00173 ; IF NO LWD IN FIRST BYTE READ, DECREMENTS POINTER UNTIL NEXT LWD
490 00173 ; AND REVERSE SPACES THROUGH THAT RECORD.
491 00173 108B080000 SPREV: ALU ,,MOVR,RO ;LSB=0 ?
492 00174 9C8B160177 JCF ,,CA,MOVR,R1,EQUAL,SPRO ;IF NOT, START SP REV
493 00175 90800605DB JCT ,,,EQUAL,BERR ;IF MSB=0 GOTO BOT ERROR
494 00176 1D98180000 ALU ,,CA,DECR,R1 ;DECREMENT MSB
495 00177 1198080000 SPRO: ALU ,,DECR,RO ;DECREMENT LSB
496 00178 108800D000 SPR1: ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDRESS
497 00179 1080080000 ALU ;
498 0017A 280E005000 ALUF MEMRD,DOUT,DTQB,,RALE ;READ DATA OUT
499 0017B 908401A5DC JCTF ,,MOVQ,,PARERR,SFRQ,PERR ;PARITY ERROR?
500 0017C 1084002000 ALUF ,,MOVQ,,RRFRQ ;
501 0017D 919808018F JCF ,,DECR,RO,DESTB,SPR5 ;JUMP IF NO LWD
502 0017E 9088160183 JCF ,,MOVR,R1,EQUAL,SPR3 ;IF LSB NOT 0, KEEP READING
503 0017F 9080060182 JCF ,,,EQUAL,SPR10 ;MSB=0 ?
504 00180 90800C0182 JCF ,,,ENBEOI,SPR10 ;CHECK ENBEOI SW
505 00181 918900F1EF JMPF ,,INCR,RO,SLDP,WTEND ;SET BOT & END
506 00182 1D98180000 SPR10: ALU ,,CA,DECR,R1 ;DEC MSB
507 00183 1198080000 SPR3: ALU ,,DECR,RO ;DECREMENT LSB
508 00184 108800D000 SPR2: ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
509 00185 1080080000 ALU ;
510 00186 280E005000 ALUF MEMRD,DOUT,DTQB,,RALE ;READ DATA OUT
511 00187 908008A59E JCTF ,,,DESTB,SFRQ,SR6END ;CHECK LWD
512 00188 1080002000 ALUF ,,,RRFRQ ;
513 00189 90800105DC JCT ,,,PARERR,PERR ;PARITY ERROR?
514 0018A 1198080000 ALU ,,DECR,RO ;DECREMENT LSB
515 0018B 9088170564 JCT ,,MOVR,R1,COU,SPR2 ;WAS LSB=0 ?
516 0018C 908006018E JCF ,,,EQUAL,SPR9 ;MSB=0 ?
517 0018D 90800C059F JCT ,,,ENBEOI,SPR11 ;CHECK ENBEOI SW
518 0018E 9D98100184 SPF9: JMP ,,CA,DECR,R1,SPR2 ;DEC MSB
519 0018F 108800D000 SPR5: ALUF ,,MOVR,RO,SALE ;LOAD LSB ADDR
520 00190 1080080000 ALU ;
521 00191 280E005000 ALUF MEMRD,DOUT,DTQB,,RALE ;READ DATA OUT
522 00192 108000A000 ALUF ,,,SFRQ ;REFRESH
523 00193 1080002000 ALUF ,,,RRFRQ ;
524 00194 90800105DC JCT ,,,PARERR,PERR ;PARITY ERROR?
525 00195 1084080000 ALU ,,MOVQ ;
526 00196 91980805B4 JCT ,,DECR,RO,DESTB,SPR2 ;CHECK LWD
527 00197 908817056F JCT ,,MOVR,R1,COU,SPR5 ;WAS LSB=0 ?
528 00198 908006019A JCF ,,,EQUAL,SPR4 ;MSB=0 ?
529 00199 90800C059F JCT ,,,ENBEOI,SPR11 ;CHECK ENBEOI SW
530 0019A 30EA18000C SPR4: ALU CONST,,DXR,R1,H#00C ;MSB=C ?
531 0019B 908006019D JCF ,,,EQUAL,SPR6 ;IF NOT, JUMP
532 0019C 1080006000 ALUF ,,REOT ;ELSE, RESET EOI
533 0019D 9D9810018F SPR6: JMP ,,CA,DECR,R1,SPR5 ;DEC MSB
534 0019E 90800021EF SR6END: JMPF ,,,RRFRQ,WTEND ;RESET REFRESH REQ
535 0019F 918900F1EF SPR11: JMPF ,,INCR,RO,SLDP,WTEND ;SET BOT & END

```

