GenRad

GR 2515
COMPUTER-AIDED TEST SYSTEM
SERVICE MANUAL

Publication 2515-0101

Revision 1, Original
December 1984

GENRAD, INC.
Santa Clara Facility
2855 Bowers Avenue
Santa Clara, CA 95051

Telephone (408) 727-4400

REVISION PAGE

September 1984	PRELIMINARY issued.
December 1984	Revision 1, Original issued.

FOREWORD

The material in this manual is for informational purposes only and is subject to change without notice. GenRad assumes no responsibility for any errors or for consequential damages that may result from the use or misinterpretations of any procedure in this publication.

GenRad

We warrant that this product is free from defects in material and workmanship and, when properly used, will perform in accordance with applicable GenRad specifications. If within 90 days after original shipment it is found not to meet this standard, it will be repaired or, at the option of GenRad, replaced at no charge when returned to a GenRad service facility. Changes in the product not approved by GenRad shall void this warranty. GenRad shall not be liable for any indirect, special, or consequential damages, even if notice has been given of the possibility of such damages.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

GenRad policy is to maintain product repair capability for a period of five years after original shipment and to make this capability available at the then prevailing schedule of charges.

SERVICE POLICY

After ninety (90) days following original installation, the product will be repaired at our then prevailing schedule of charges.

Your local GenRad office or representative will assist you in all matters relating to product maintenance, such as calibration, repair, replacement parts and service contracts. Field servicing of GenRad system products can be accomplished by any of the following methods:

- By GenRad on a contract specifying a fixed price per period.
- By GenRad on a per call basis with no contract, or
- By the customer, after purchase of spares and service training from GenRad.

July 1980

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SAFETY PRECAUTIONS

WARNING

Beware of possible internal lethal voltages present even though AC power has been disconnected. Death or serious bodily injury can occur. Only qualified personnel should perform any service or repair of the system components.

CAUTION

Before attempting any service, repair, or power switch setting changes, turn power OFF and disconnect unit from power source.

CAUTION

Before turning power ON remove the cardboard card from the floppy disk drive slot. Save this card. Reinsert it before transporting the unit to another location.

CAUTION

Do NOT operate the system with top cover removed. Serious damage to internal components may result due to overheating.

CAUTION

Before disconnecting the power cable, main unit, or any peripheral, turn power OFF. Do NOT perform any disconnection while system is operating.

SECTION I

INTRODUCTION

1.1 SCOPE AND ORGANIZATION

GenRad operating and service information for the 2515 Computer-Aided Test System is supplied in the following publications:

2515 Computer-Aided Test System Operating Manual	2515-0100
2515 Computer-Aided Test System Service Manual	2515-0101
2515 Computer-Aided Test System Diagrams and Parts Lists	2515-0102

This manual is intended for use by qualified service technicians who have attended the GenRad 2515 service training course and prerequisite courses. The user is assumed to be familiar with the Digital Equipment Corporation PDP-11 Unibus operation, LSI-11/23 Processor operation, peripheral interfaces, general digital circuitry, as well as the system components, terminology, general hardware controls, and basic software procedures as described in the system operating manual.

This service manual contains theory of operation for the GenRad manufactured hardware components, alignment procedures and diagnostic programs and instructions for their use. Service of the 2515 system depends primarily on the use of the diagnostic software. The diagnostic programs exercise and test both GenRad and vendor units within the system. The diagnostics section provides troubleshooting aids to isolate a problem to a particular board. Background information on the units and circuits is provided to aid understanding and effective use of the diagnostics. Appendices containing a manual reference list, mnemonics and abbreviations, and specific data are included in the back of this manual. Hardware manuals for the vendor components are supplied with the system.

The Diagrams and Parts Lists volume 2515-0102 contains the assembly drawings, parts lists, and detailed schematics that are referenced herein.

1.2 GENRAD FIELD SERVICE

Our warranty attests the quality of materials and workmanship in our products. If difficulties do occur, our service engineers will assist you in anyway possible. If the difficulty cannot be eliminated by use of the service material provided, contact the nearest GenRad service facility and give full information about the trouble and the actions taken to remedy it. Describe items by type, serial number, and ID numbers.

1.3 EQUIPMENT RETURN

Before returning any equipment to GenRad for service, obtain a "Returned Material" (or RM) number from the nearest GenRad office. Use of this number in correspondence and on a tag tied to the returned item will ensure proper handling and identification. For equipment not covered by warranty, a purchase order should be forwarded to avoid unnecessary delay.

For return shipment, use the original packing, shipping cases, or packaging that will protect the unit from damage during transit. Advice may be obtained from the nearest GenRad office.

1.4 SPECIFICATIONS

Table 1-1 lists the specifications for the 2515 systems.

Table 1-1. List of Specifications

DIMENSIONS AND WEIGHT	
Main Chassis:	
Height	10.5 inches (18 cm)
Width	17.0 inches (43 cm)
Depth	24.0 inches (41 cm)
Weight	70.0 pounds (31.5 kg)
Storage Chassis:	
Height	4.0 inches (10 cm)
Width	6.0 inches (15 cm)
Depth	9.0 inches (23 cm)
Weight	3.2 pounds (1.45 kg)
Keyboard Chassis:	
Height	1.5 inches (4 cm)
Width	19.0 inches (48 cm)
Depth	8.5 inches (22 cm)
Weight	5.6 pounds (2.52 kg)
Printer Chassis (Opt):	
Height	4.0 inches (10 cm)
Width	17.0 inches (43 cm)
Depth	17.0 inches (4.3 cm)
Weight	12 pounds (5.4.kg)
POWER REQUIREMENTS	
Voltage	120, 220, or 240 nominal
Frequency	50/60 Hertz power 500 watts 650 watts, standard system 800 watts, fully loaded system
ENVIRONMENT	
Temperature	10°C to 45°C, operating
Humidity	20% to 80% RH @ 40°C
Safety Standard	UL 1244 Specifications
Test Standard	GenRad Class 2

Table 1-1. List of Specifications (continued)

INPUT

Number of Baseband Mode Channels:	Four analog channels standard (optionally expandable to 16).
Number of Zoom Mode Channels:	Two channels standard (optionally expandable to 8).
Frequency Range:	DC to 25.6 kHz with alias protection on all channels.
Anti-Alias Filters:	15 selectable digital filter ranges below 25.6 kHz. One analog filter at 25.6 kHz.
Dynamic Range:	>70 dB with 12 bit A/D conversion.
Amplitude Flatness:	±0.25 dB over the entire frequency range.
Input Sampling:	Simultaneous sample and hold (all channels).
Channel-to-Channel Match:	< 0.4 dB amplitude difference, < 2.0 degrees phase difference to 10 kHz < 4.0 degrees phase difference to 20 kHz < 8.0 degrees, phase difference to 25 kHz
Input Impedance:	100K ohm shunted by <150 pF.
Input Coupling:	AC (-3 dB at 0.8 Hz), DC, PCB transducer bias source.
Sensitivity:	8 ranges (62.5 mV, 0.125 V, 0.25 V, 0.5 V, 1.0 V, 2.0 V, 4.0 V, 8.0 V peak input) and auto ranging.
Maximum Operating Level:	10 V (full-scale sensitivity typically tolerates a 25% overrange).
Maximum Voltage:	Protected to 25 V.

BASEBAND RESOLUTION

Bandwidths:	15 selectable ranges from 1.28 Hz to 20.5 kHz, plus 25.6 kHz.
Frame Size:	6 selectable ranges from 256 to 8,192 points/channel.
Frequency Lines:	320 alias-free lines for every 1,024 input points in the frame.

Table 1-1. List of Specifications (continued)

Maximum Resolution (lines):

Maximum Number of Frequency Lines (Cross-Channel Acquisition Mode)	Number of Channels	
	Standard Memory	Optional Memory
320	16	16
512	12	16
640	8	16
1280	4	16
2560	2	12

ZOOM

Operation: User specified window in the range DC to 25.6 kHz.

Control: Center frequency selectable to 1.0 Hz resolution anywhere in DC to 25.6 KHz range.

Bandwidth: Selectable in powers of 2 from 2.5 Hz to 20.5 kHz.

Resolution: Same as baseband mode for one-half the number of channels.

PROCESSING

Measurements Performed:

Time domain: Input time, averaged time, auto correlation, cross correlation, impulse response.

Frequency domain: Input spectrum, frequency response function, coherence, auto power spectrum, linear spectrum, power spectral density, cross power spectrum, transmissibility, coherent output power.

Amplitude domain: Probability density function, cumulative distribution function.

Block Arithmetic Operations: +, -, x, ÷, integration (single or double), differentiation (single or double).

Table 1-1. List of Specifications (continued)

AVERAGING	
Types:	Additive, subtractive, exponential, peak hold.
Control:	Run, stop, resume, automatic reject, manual reject.
Weighting Windows:	Uniform, Hanning, flat top, special (Blackman-Harris), force (impact), response (impact), correlation (zero padding).
TRIGGERING	
Acquisition Modes:	Free run, auto arm trigger, manual arm trigger.
Free Run:	New measurement initiated at completion of previous measurement.
Automatic or Manual:	New measurement initiated when signal into armed trigger channel meets specified threshold. Any channel may be used for trigger.
Trigger Level:	Adjustable percentage of full scale.
Trigger Slope:	+ or -
Trigger Delay:	Pre-trigger and post-trigger delays
Pre-trigger:	$\leq$T (T is the time of one frame)
Post-trigger:	$\leq$32,000 t (t is sample interval)
Resolution:	$\pm \frac{1}{2} \Delta t$
External Trigger:	TTL level.
Overlapped Processing:	0%, 50%, Max.

Table 1-1. List of Specifications (continued)

STORAGE	
Capacity:	10 Mbytes on Winchester drive and 0.5 Mbytes on mini-floppy drive.
Spectrum Recall:	Data can be stored and retrieved with calibration and annotation information.
Setup/Recall:	Multiple setup states can be stored and recalled.
OUTPUT SIGNALS	
Sine Wave:	1 Hz to 25.6 kHz in 1 Hz increments. 4.5 Vrms maximum level.
Random Noise:	25.6 kHz fixed bandwidth. 1.0 Vrms maximum level.
Band Translated Random Noise:	Noise bandwidth concentrated in analysis band (including zoom bands). 1.0 Vrms maximum level.
Amplitude Adjustment (all signals):	Adjustable attenuation of output level.
DISPLAY	
Size:	7-inch (diagonal measure).
Resolution:	640 lines horizontal. 480 lines vertical.
Graphics:	Real-time displays of measurement data with interactive format and unit selections.
Formats:	Full screen, split screen (top/bottom), superimposed displays of measured functions.
Cursors:	
Single:	Main cursor and reference cursor for relative readouts.
Sideband:	Up to 7 on both sides of main cursor.
Harmonic:	Up to 15 at harmonics of main cursor.

SECTION II

INSTALLATION

This section describes the GenRad installation policy for the 2515 System, installation, not normally performed by the customer, the procedure for power switching, and the factory installed jumper configurations.

2.1 INSTALLATION POLICY

Initial installation is to be performed by the customer. Normal installation procedures are described in the 2515 System Operating Manual 2515-0100.

2.2 SWITCHING POWER

The system is designed to operate at 110 VAC or 220 VAC. Power setting will normally be performed prior to shipment. If after installation, should on site power requirements change the following steps are to be followed:

a. Switch power ON/OFF switch to OFF.

b. Disconnect power cable from AC power source.

c. Remove 6 pan head screws from sides of top cover and carefully lift cover off unit.

d. On right side of power supply, locate small rectangular opening. Visually inspect interior switch.

e. Using a $\frac{1}{4}$" flat screwdriver, push switch up for 220-240V; push switch down for 120V.

f. On rear panel locate 120V versus 220-240V label and remove the two retaining screws.

g. With new power setting facing outward, re-attach label.

h. Replace cover and 6 screws.

i. Before unit is connected to ac power source, be certain power cable plug and fuse are changed to accommodate the new voltage setting. (120V requires a 8A slow blow fuse, 220-240V requires a 4A slow blow fuse.)

2.3 CRT INSTALLATION

To install a new CRT assembly, perform the following in order:

a. Switch power ON/OFF switch to OFF.

b. Disconnect power cable from AC power source.

c. Perform steps b. through k. of paragraph 3.3.5 to remove faulty CRT assembly.

d. Position CRT into system through top side and slide forward to the front panel assembly.

e. Replace 4 screws to secure CRT and bezel to front panel assembly.

f. Lift hinged mounting plate to expose underside. Position CRT control board onto underside of mounting plate and secure with 4 allen head screws.

g. Connect 10-pin connector (from display board) to right side of CRT control board.

h. Replace WINC05 right side slide bracket and secure with 2 screws.

i. Replace WINC05; slide unit into system through front panel.

j. Connect DC power connector P3 to WINC05.

k. Connect P/J connector cables to WINC05.

l. Be certain WINC05 assembly is seated properly; secure by tightening the 4 mounting screws to the frame to hold the drive.

m. Replace bottom cover and tighten the 9 retaining screws.

n. Rotate unit so that top side is on top.

o. Connect system power cable to AC power source and switch power ON/OFF switch to ON.

p. Perform standard CRT alignment procedures.

q. Replace top cover and secure by tightening 6 pan head screws.

2.4 POWER SUPPLY INSTALLATION

To install a new power supply, perform the following in order:

a. Switch power ON/OFF switch to OFF.

b. Disconnect power cable from AC power source.

c. Perform steps b. through m. of paragraph 3.3.4 to remove faulty power supply assembly.

d. Place new power supply into system through the top side.

e. Connect AC and 5V rear connectors.

f. Secure top cover panel (8 screws) to power supply.

g. Connect base panel (4 screws) holding line filter and splitter board.

h. Connect remainder of rear connections. These wires should be marked to ensure proper connection to new power supply. If not, refer to the interconnect diagram, figure 3-2, and to the vendor manual listed in Appendix A. Once connections are completed properly, mark connecting wires to facilitate future servicing.

i. Install rear access panel with 4 retaining screws.

j. Put on bottom cover and secure 9 retaining screws.

k. Rotate system so that unit is upright for use (top is at top).

l. Replace top cover and secure with 6 pan head screws. If power fail adjustment is required, see following sub-paragraph 2.4.1.

2.4.1 Power Fail Option

Access to the power fail (PF) adjustment potentiometer (pot R17) is through the PF hole in the power supply top cover. The power fail threshold and reset signal is factory set at 86-90/172-180(VAC) with all outputs loaded to nominal output current. To check and/or adjust, perform the following:

a. Connect a scope probe to J1-5 with respect to J1-4. Also connect a 5.1k resistor from J1-5 to an external +12V source.

b. The power fail signal shall be a logic 1 when the AC input voltage is above the threshold or reset levels.

c. To set, adjust the AC line to the lower end of the threshold window and adjust the power fail potentiometer so that the signal just goes low. Slowly raise the AC input voltage and check the power fail reset (back to a logic 1) level. The reset level should be back to a logic 1 before the AC input reaches the upper threshold limit. If not, reset the PF pot R17.

2.5 JUMPER CONFIGURATIONS

The following represents factory configurations for the various board assemblies being used in the 2515 Computer Aided Test System.

2.5.1 Unibus Memory Board, Cambex 207-104-600 Rev. A

Jumpers:

J-1 Not jumpered
J-2 Jumpered

Switch Settings:

Switch 1	Switch 2	Switch 3
1-5 = OFF	1-4 = OFF	1 = ON
6-7 = OFF	5-6 = ON	2 = OFF
	7-8 = OFF	3 = ON
		4-5 = OFF
		6-7 = ON

2.5.2 Serial Line Interface Board

2.5.2.1 Serial Line Interface Board, PM-DLV11J.

Channel 1 will be equivalent to 422 setup on wire jumpers: 1G-1F 1C-1B

Channels 2 - 4 will be equivalent to 232 setup as follows:

	From	To
Channel 2	2E	2F
	2B	2D
Channel 3	3E	3F
	3B	3D
Channel 4	4E	4F
	4B	4D

Baud Rate Selection:

Channel 1 = 300	Wire Wrap Pin G
Channel 2 = 9600	Wire Wrap Pin F
Channel 3 = 1200	Wire Wrap Pin F
Channel 4 = 1200	Wire Wrap Pin F

Data Word Format:

1) There are 8 data bits.
2) There is no parity on all channels.
3) Number of stop bits equals 2 (same as before).

Break Response Jumper: E4 - E5 Will give halt when break is desired.

Switch Settings:

Switch 1,	Pin 1 = OFF	Channel 1 starts at address 775640.
	Pin 2 = OFF	
	Pin 3 = ON	
	Pin 4 = OFF	
	Pin 5 = OFF	Channel 2 starts address at 775650.
	Pin 6 = OFF	
	Pin 7 = OFF	
	Pin 8 = ON	
Switch 2,	Pin 1 = OFF	Channel 3 starts at address 775660.
	Pin 2 = ON	
	Pin 3 = OFF	
	Pin 4 = OFF	
	Pin 5 = OFF	Channel 4 starts at address 775670.
	Pin 6 = ON	
	Pin 7 = OFF	
	Pin 8 = OFF	

Vector Base Address 340-370 will put Switch 1 - 4 to OFF.

2.5.2.2 Serial Line Interface Board, DEC DLV11J.

Channel 1 is configured to 422 set up on wire jumpers:

M0	X to 2
N0	X to 2

Channels 2-4 are configured to 232 setup as follows:

Channel 2	M1, N1	X to 3
Channel 3	M2, N2	X to 3
Channel 4	M3, N3	X to 3

Baud Rate Selection:	Channel 1 = 300	Wire Wrap Pin T
	Channel 2 = 9600	Wire Wrap Pin N
	Channel 3 = 1200	Wire Wrap Pin W
	Channel 4 = 1200	Wire Wrap Pin W

Date Word Format:

- There are 8 data bits accomplished by D connections at 1 to X for all four channels.
- There is no parity on all channels:

Jumper	E	X to 0
	P	X to 1

- Stop Bits:

Channel 1 = 2	S	X to 1
Channels 2-4 = 1	S	X to 0

Break Response Jumper: X and H are jumpered to give halt when break is desired.

Address Jumpers:

Pins	Jumpered
A5	X to 1
A6	Removed
A7	Installed
A8	X to 1
A9	X to 1
A10	X to 0
A11	X to 1
A12	X to 1

Vector Base Address 340-370 is configured by the following:

Pin	V5	X to 1
Pin	V6	Installed
Pin	V7	Installed

2.5.3 Winchester Interface Board

Jumpers E2 - E3 Give auto boot to automatically come up to RT-11.

E4 - E5 Sets Winchester at standard address of 774400.

E7 - E8 Sets Floppy to its standard address of 777170.

E10 - E11 Disables 22 bit addressing.

E12 - E13 For internal test purposes.
E14 - E15 For internal test purposes.
E16 - E17 For internal test purposes.

2.5.4 Floppy Interface Board

Jumpered 3 and 10

2.5.5 LSI-11/23 Processor Board

W18 is removed (enables wakeup circuit).
W1 is installed (master clock).
W4 is removed (BEVENT line enable).
W5 is removed.
W6 is installed (power up mode to execute Boot).
W7 is removed (enter ODT on HALT).
W8 - W15 are installed.

2.5.6 Memory Controller Board

Switch Settings:

Switch 1 = OFF
2 = ON
3 = OFF
4 = ON
5 = ON
6 = OFF
7 = ON
8 = OFF
9 = OFF
10 = OFF

2.5.7 CPU and PROM Boards

These boards are the same with present systems.

2.5.8 Display Board

Interrupt Vector Select Jumpers:	V6, V5, and V4 are jumpered. V1, V2, and V3 are not jumpered.
Address Select Jumpers:	SD4 and SD5 are jumpered. SD3, SD6, SD7, and SD8 are not jumpered.

2.5.9 Utility Board

Switch Settings:

E1 shorted to E2	E11 shorted to E12
E3 shorted to E4	E19 shorted to E21
E5 shorted to E6	TP3 shorted to TP4
E7 shorted to E8	

2.5.10 Channel Control Board

Wire tie 1 to wire tie 2, wire tie 3 to wire tie 4.

2.5.11 25 kHz Channel Board

E3 to E34 E2 to E1

2.5.12 Front Panel Control Board 2

E1, W1, and W2 shunted.

SECTION III

MECHANICAL AND DISASSEMBLY

This section contains the physical description, interconnections, and disassembly instructions required for proper service of the 2515 CAT System hardware.

Table 3-1 lists the major components supplied for the standard system. Table 3-2 lists the available factory installed hardware options. Specific parts information for GenRad manufactured components are found in the Diagrams and Parts Lists manual 2515-0102. Refer to the appropriate vendor manual listed in Appendix A for vendor supplied components.

Table 3-1. List of Major Components Supplied

Discription	Part Number
2515 Computer Aided Test System (Bench Top; 128K, 110V/220V)	2500-9700-1, -2
Chassis Sub-Assembly	2515-2041
Front Panel PCB Assembly	2515-2042
CPU Board	2501-4700
PROM Board	2515-4709-01
Utility Board	2515-4705
Memory Controller Board	2515-4702
Medium Resolution Display Board	2515-4706
Channel Control Board	2515-4701
25 kHz Channel Board	2515-4700
Winchester/Floppy Controller	1765-3223
Serial Line Interface	1765-0066
Unibus Memory	1765-1112
LSI-11/23 CPU	1765-0553
Power Supply	6067-0068
Monitor Kit, 7"	1765-5537
CRT Frame	2515-8048
Keyboard	1765-7002
Winchester Disk Drive	1765-3221
Floppy Assembly	2515-2043
Cables	
Power	2515-2029
Jumper, 10" BNC-BNC(1)	1923-2098-1
Jumper, 7' BNC-BNC(2)	1923-2140-01/02
Diskettes, 5¼" Floppy (1 Box)	2515-9411

Table 3-2. List of Optional Equipment

Description	Part Number
Rackmount Kit	2515-9400
Channels 5-8	2515-9401
Four Additional Channels	2515-9402
Large Screen Monitor	2515-9403
Extended Memory	2515-9404
Dual Floppy Disk Drive	2515-9405
Printer/Plotter	2515-9406
Shipping Case (System Unit)	2515-9407
Shipping Case (Monitor and Printer)	2515-9408
Paper, Printer/Plotter, Case	2515-9410
Diskettes, Mini-Floppy Box of Ten (10)	2515-9411
Diskettes, Dual Floppy, Box of Ten (10)	2515-9412
Q-Bus Expansion	2515-0113

3.1 PHYSICAL DESCRIPTION

Dimensions and weights are listed in Table 1-1. All standard components, except for the Keyboard and Floppy assemblies are located within the main chassis. The following sub-paragraphs describe the major components and their location.

3.1.1 Chassis Sub-Assembly

The 2515 chassis sub-assembly, part number 2515-2041-1D, consists of the front panel assembly, card cage chassis, rear panel assembly, top and bottom covers, and other miscellaneous hardware required to house the 2515 system components in a compact, portable unit. Two machined handles, left and right, facilitate lifting. Additionally, a tilt stand located on the front bottom of the chassis can be utilized to angle the unit to facilitate viewing. The following sub-paragraphs describe the major components comprising the chassis.

CAUTION

Do not operate system with the top cover off. Serious damage to components may result.

3.1.1.1 Front Panel. Two major assemblies comprise the front panel: Front Panel Sub-assembly P/N 2515-3001 and Front Panel PCB Assembly P/N 2515-2042.

a. Front Panel Sub-Assembly The front panel sub-assembly includes the CRT frame, P/N 2515-8048 and the lower front panel, P/N 2515-8026. When installed, the display monitor CRT is framed by and mounted onto the CRT frame. The lower front panel is machined to fit over the POWER ON/OFF rocker switch (DPST, 16A, 250 VAC), BOOT/HALT toggle switch, and five BNC connectors.

b. Front Panel PCB Assembly The front panel PCB assembly consists of four items: label overlay, upper front panel, and control panel boards 1 and 2. The label overlay adhers to the upper front panel and both have identical cutouts through which the momentary button switches and LEDs on control panel 1 protrude, making them visable and available to a user. Note that of 21 LEDs, only 14 are used in this system; of 101 available switches, 79 are currently used. Unused LEDs and switch capabilities are available for future expansion. Control panel boards 1 and 2 are discussed in Section IV, paragraph 4.3.

3.1.1.2 Card Cage Chassis. The card cage chassis, part number 2515-8047, is primarily a center bracket and a single, three-sided, machined enclosure with card guides sufficient to hold a total of seventeen circuit board assemblies. Of the seventeen slots, eleven are used by 6-wide boards and six are used by 2-wide boards. In the standard system eleven slots are used leaving three 2-wide and three 6-wide slots available for future expansion.

The center bracket separates the space left by the six 2-wide board assemblies. Both pieces are joined to the center bulkhead by socket screws.

3.1.1.3 Rear Panel Assembly. The rear panel assembly, part number 2515-3000-ID, is available in two modifications, Rev. 5 and Rev. 6. Rev. 5 is for systems manufactured prior to September 1984. Rev. 6 is for systems manufactured after September 1984 and has been designed to accept the Q-Bus Option and two additional BNC connectors.

This assembly consists of a footed bezel to which a 4 x 5" fan screen, 4 x 5" 122 CFM fan, power plug connector, 6 amp fuse and holder, floppy and Winchester cable assemblies are connected. Two handle brackets are used to secure the assembly to the left and right side handles on the chassis.

The fuse is slow-blow, 8 ampere, 125-volt, 3AG and should be replaced as required. For 220V operation a 4 ampere, slow-blow fuse is used.

3.1.1.4 Covers, Top and Bottom. The top cover, part number 2515-8015, fits over the top and both sides of the system attaching to the inside edge of the side handles by six screws and measures approximately 16-2/3 W x 23-1/3 L inches.

The bottom cover, part number 2515-2003-1D, is a flat, machined piece with cutouts onto which two filters are mounted. The bottom is attached to the unit by nine screws.

CAUTION

Do not operate system with covers removed.

3.1.2 Power Supply and Line Filter

The power supply used, part number 6067-0068, is Model RSF500 manufactured by ACDC Electronics Corporation. It provides four channel output of +5V, +12V, +15V, and -15V. Input voltage is 115/230 VAC and is switch selectable. See Section II, paragraph 2.2 for switch setting instructions.

Additional AC line filtering is provided by a 10 amp line filter, part number 5280-1354.

Refer to the vendor manual listed in Appendix A for additional power supply details.

3.1.3 Drive Interconnect

This assembly, part number 2515-4710, is simply a flat, machined board with J connectors sufficient to provide interconnections for the floppy disk drive, the Winchester disk drive, the WINC05 interface board, and the power supply.

It is located directly beneath the power supply and is mounted to it by four Allen screws. The line filter is also mounted onto the drive interconnect for convenience. Sufficient connector space (J2B) has been reserved for the future addition of a second floppy connector.

3.1.4 Fan and Cooling

A 122 CFM fan provides air flow for the circuitry allowing a maximum of 45°C. Operating the system without the cover attached or without one or more of the 6-wide circuit card assemblies will cause a rapid heat increase and probable damage to components.

3.1.5 Backplane

The backplane (or motherboard), part number 2515-4704, is a printed circuit board located below the card cage and above the bottom cover. Each of the board assemblies are normally seated onto 78 connector housings which are designed to accommodate eleven 6-wide boards and six 2-wide boards, six of which are spare in the standard system. The backplane provides power, grounding, and signal routing for the board assemblies. The backplane card slot designations are shown in figure 3-1.

Thirteen pan-head screw terminations are provided on the backplane for +5-volts, +12-volts, -15-volts, and +15-volts power and digital and analog grounds. One 16-pin header, J17, receives the power from the power supply to feed the Floppy drive with +5V, +12V and ground. One 40-pin header, J11, connects the backplane to the input connector located on the rear panel which will feed power to a printer, if connected. Sixteen Berg headers (two 2-pin, two 6-pin, and twelve 4-pin) connect the backplane to rear and front panel connectors and to the CRT. Table 3-3 lists the backplane connections.

Table 3-3. Backplane Connections

From Connector	Pin	To Connector	Pin	Function
Channel 1 BNC		P1A	1	Ch. 1 HI
			2	Ch. 1 LO
Channel 2 BNC		P1A	3	Ch. 2 HI
			4	Ch. 2 LO
Channel 3 BNC		P2A	1	Ch. 3 HI
			2	Ch. 3 LO
Channel 4 BNC		P2A	3	Ch. 4 HI
			4	Ch. 4 LO
Test Out BNC		P9	1	Test Signal
			2	Test Ground
P1B	1	Input Connector	A	Ch. 1 HI
	2		A	CH. 1 LO
	3		B	CH. 2 LO
	4		B	CH. 2 HI
P2B	1		C	CH. 3 HI
	2		C	CH. 3 LO
	3		D	CH. 4 LO
	4		D	CH. 4 HI
P3	1		E	CH. 5 HI
	2		E	CH. 5 LO
	3		F	CH. 6 LO
	4		F	CH. 6 HI
P4	1		H	CH. 7 HI
	2		H	CH. 7 LO
	3		J	CH. 8 HI
	4		J	CH. 8 HI
P5	1		K	CH. 9 HI
	2		K	CH. 9 LO
	3		L	CH. 10 LO
	4		L	CH. 10 HI
P6	1		M	CH. 11 HI
	2		M	CH. 11 LO
	3		N	CH. 12 LO
	4		N	CH. 12 HI

Table 3-3. Backplane Connections (Continued)

From Connector	Pin	To Connector	Pin	Function
P7	1	Input Connector	P	CH. 13 HI
	2		P	CH. 13 LO
	3		R	CH. 14 LO
	4		R	CH. 14 HI
P8	1		S	CH. 15 HI
	2		S	CH. 15 LO
	3		T	CH. 16 LO
	4		T	CH. 16 HI
P10	1	Ext. Trig. BNC		EXTRIN
	2			GND
	3	Ext. Sample		GND
	4	Clock BNC		EXTCLK
P13	1	Comp. Video #1 BNC		COMPVID I
	2			GND
P12	1	TTL Video, C2	2	V SYNC (Signal)
	2	C2	1	V SYNC (Shield)
	3	C2	6	H SYNC (Signal)
	4	C2	10	H SYNC (Shield)
	5	C2	8	COMP VIDEO (Signal)
	6	C2	10	COMP VIDEO (Shield)

3.2 INTERCONNECTIONS

A description of the unit interconnections is found in the 2515 System Operating Manual, 2515-0100 (Section II, paragraph 2.9) Refer also to the overall interconnection diagram, figure 3-2.

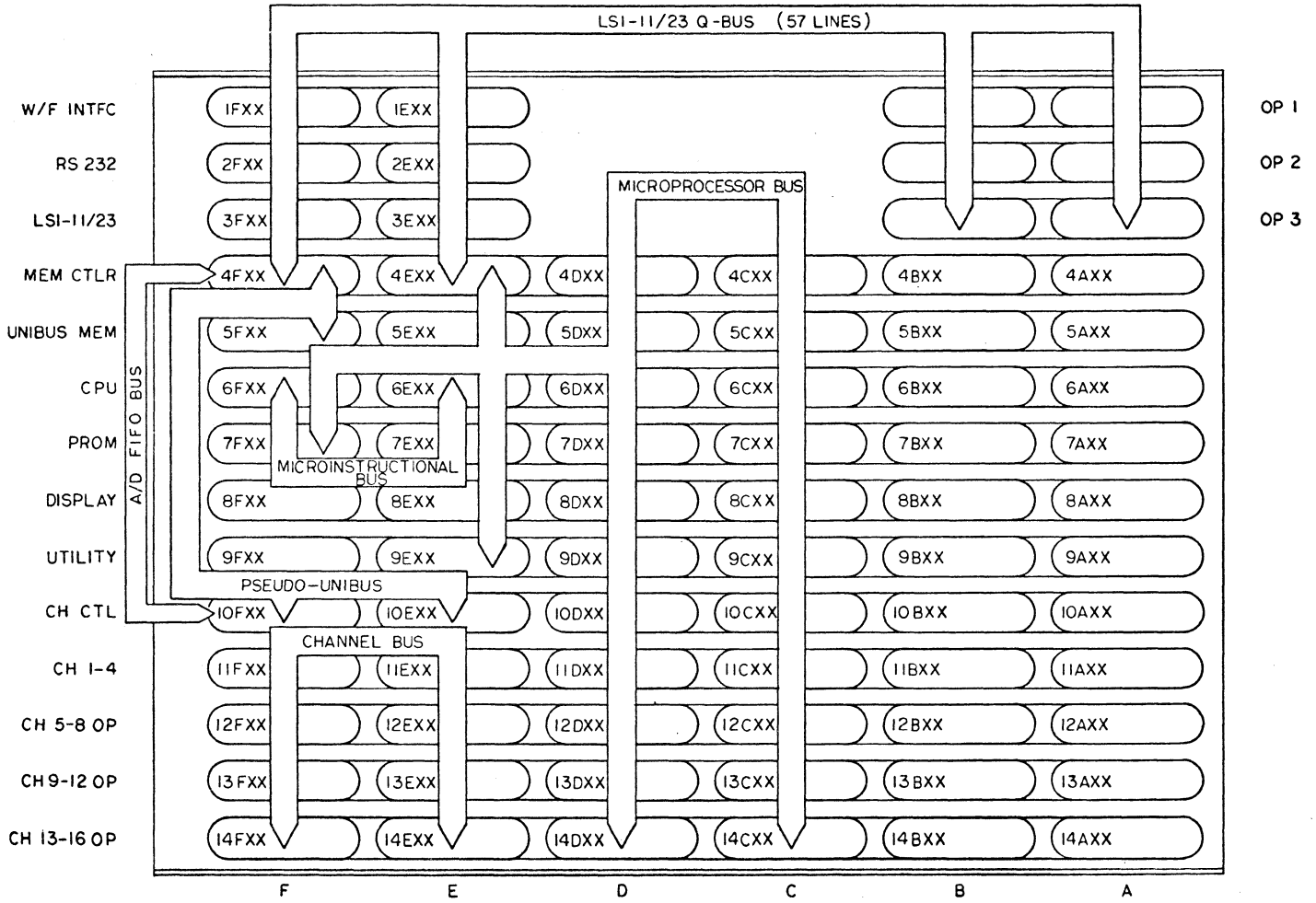


Figure 3-1. Card Slot Designations and Buspaths

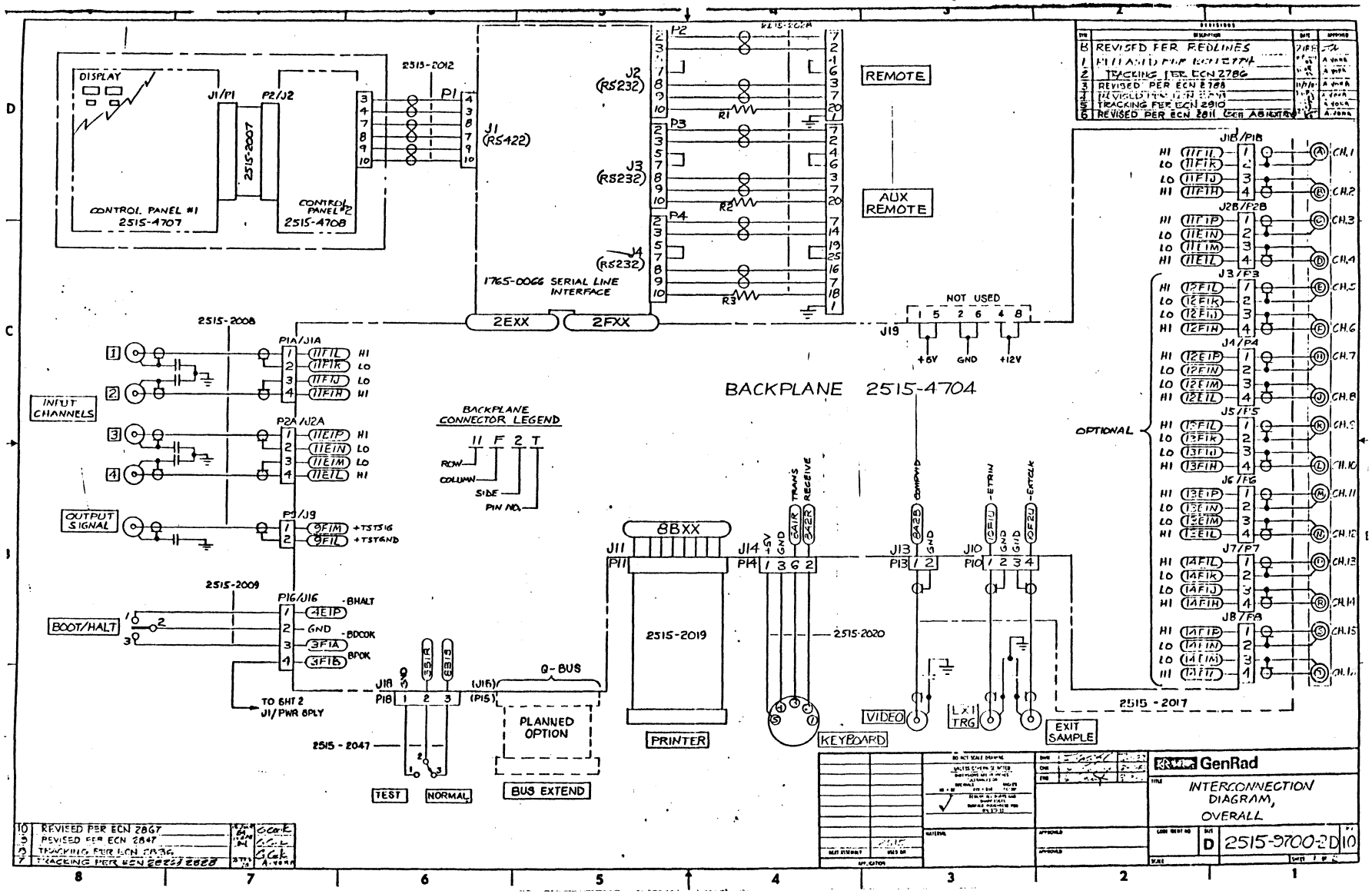


Figure 3-2. Interconnect Diagram

(Sheet 1 of 2)

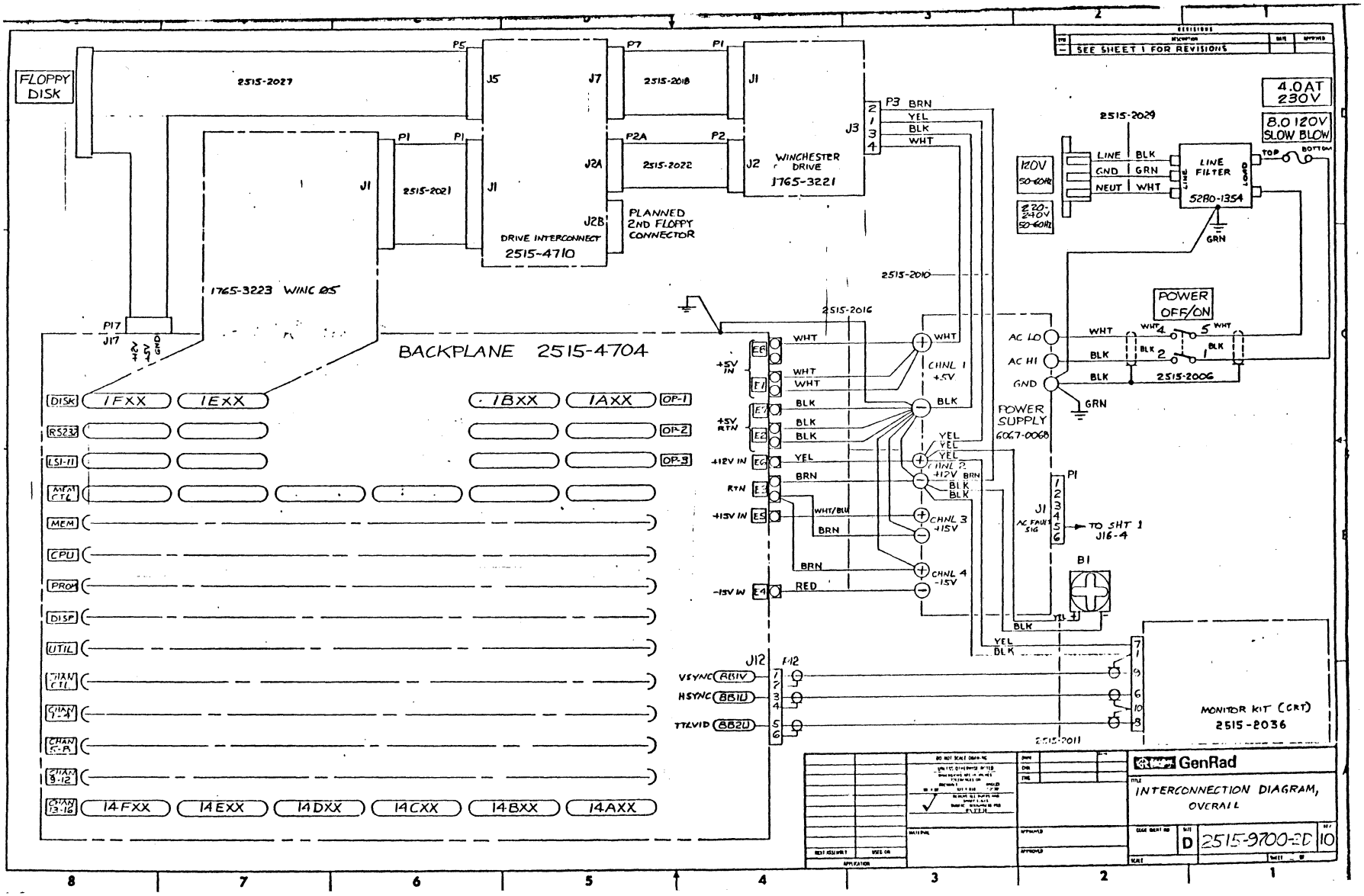


Figure 3-2. Interconnect Diagram
(Sheet 2 of 2)

3.3 DISASSEMBLY PROCEDURES

General disassembly of the chassis sub-assembly is supported by assembly diagram 2515-2014. Specific procedures for removing covers and major internal components are given in the following sub-paragraphs.

WARNING

Lethal voltages are present. Before attempting removal of any assembly or component, switch power to OFF and disconnect from power source.

3.3.1 Top Cover, Removal

a. Remove 6 pan head screws from sides of top cover.

b. Carefully lift cover up and off.

3.3.2 Bottom Cover, Removal

a. Disconnect system from power source/wall outlet.

b. Rotate system to rest on the side.

c. Remove the 9 retaining screws.

d. Lift bottom cover off.

3.3.3 Winchester WINC05, Removal

a. Disconnect system from power source/wall outlet.

b. Rotate system so that it rests on right side.

c. Remove bottom cover of system by performing procedure in paragraph 3.3.2.

d. Remove the 4 mounting screws from frame holding the drive.

e. Slide unit forward slightly.

f. Disconnect P/J connector cables. Using left hand, reach in to disconnect dc power connector P3.

g. Slide unit forward and out of system.

3.3.4 Power Supply, Removal

a. Disconnect system from power source/wall outlet.

b. Remove the top cover of system by performing procedure in paragraph 3.3.1.

c. Rotate system so that it rests on right side.

d. Remove bottom cover of system by performing procedure in paragraph 3.3.2.

e. Remove two screws from CRT mounting plate; slide plate forward.

f. At rear panel, remove access panel by removing the 4 retaining screws.

g. Mark connecting wires before disconnecting to assure proper reconnection.

h. Remove connections after marking.

i. To safely disconnect the AC and 5V connectors it will be necessary to perform i and j.

j. Remove 4 screws from base panel holding the line filter and splitter board.

k. Remove the 4 screws from the top cover of the power supply. The power supply will drop slightly. Cover will be free to remove.

l. Remove 8 screws holding cover panel. Complete the disconnection at rear.

m. Pull power supply out from top of system.

n. To install new power supply use procedure in Section II, paragraph 2.4.

3.3.5 CRT, Removal

WARNING

High voltage of approximately 18 kilovolts is stored on CRT assembly. Proceed with caution.

a. Disconnect system from power source/wall outlet.

b. Remove top cover of system by performing procedure in paragraph 3.3.1.

c. Rotate system so that it rests on right side.

d. Remove bottom cover of system by performing procedure in paragraph 3.3.2.

e. Remove WINC05 assembly by performing procedure in paragraph 3.3.3.

f. Lift hinged mounting plate to expose CRT control board.

g. Remove CRT 10-pin connector on right side of CRT control board.

h. Remove 4 allen head screws holding control board.

i. Remove screws retaining mounting plate and position (or remove) plate and control board as necessary to remove the CRT.

j. Remove 4 screws holding CRT and bezel to the front panel assembly.

k. Slide CRT toward rear and lift out top side to remove.

3.3.6 Control Panel Boards 1 and 2, Removal

a. Disconnect system from power source/wall outlet.

b. Remove top cover of system by performing procedure in paragraph 3.3.1.

c. Disconnect connector P1/J1.

d. To remove both boards together, remove the 4 mounting screws from board 1.

e. To remove the boards separately, first remove the 4 mounting screws holding board 2. Then remove 4 mounting screws holding board 1.

f. Reverse this procedure to replace boards.

SECTION IV

THEORY OF OPERATION

This section contains functional and circuit descriptions for the GenRad Model 2515 Computer-Aided Test System. Descriptions presented herein will be detailed for equipment manufactured by GenRad. Vendor supplied equipment will be described as required and supported by the appropriate vendor manuals listed in Appendix A.

In addition to the diagrams contained in volume 2515-0102, the content of this section is further supported by functional block diagrams for each board assembly and related illustrations and tables. References will be made to supportive documentation not contained herein.

4.1 OVERALL FUNCTIONAL DESCRIPTION

Under control of the DEC LSI-11/23 Processor, the 2515 Computer-Aided Test System performs high speed analysis using multi-channel calculations. Four 25 kHz channels are provided in the standard system and are optionally expandable to sixteen channels. Refer to figure 4-1.

In the standard system, thirteen circuit board assemblies and the display monitor circuitry interface compatibly via two major and four minor buses. These buses are listed below and described in detail in paragraph 4.2.

LSI-11 Q-Bus (major bus)
2501 Microprocessor Bus (major bus)
Pseudo Unibus
Microinstruction Bus
A/D FIFO Control Bus
Channel Bus

The thirteen board assemblies used in standard systems are:

Serial Interface board
LSI-11/23 Processor Board
Winchester/Floppy Interface Board
2501 Microprocessor (CPU) Board
PROM Board
Unibus Memory Board
Memory Controller Board
Medium Resolution Display Board
Utility Board
Channel Control Board
25 kHz Channel Board
Control Panel Boards 1 and 2

The following paragraphs provide detailed descriptions of the 2515 system components and capabilities.

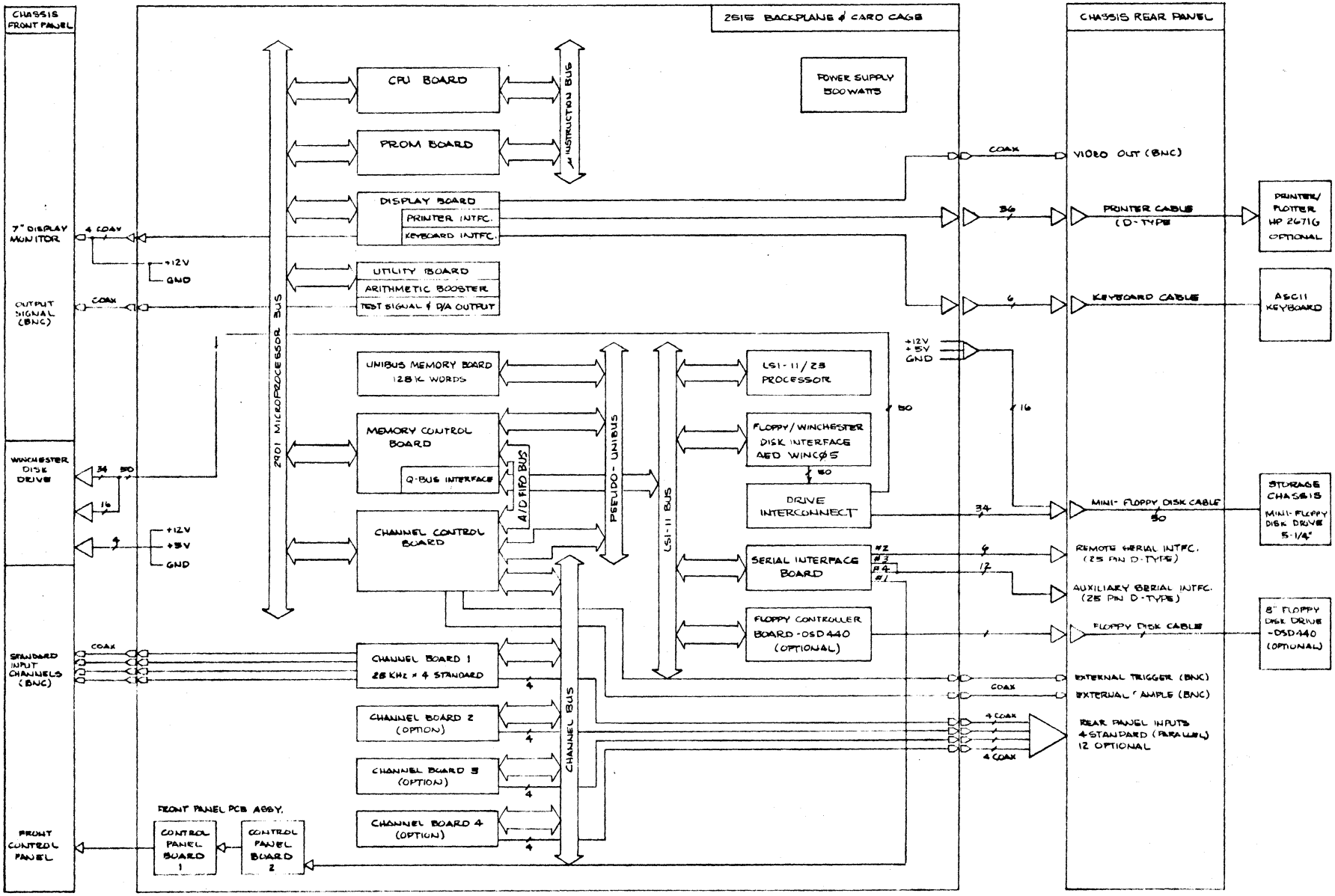


Figure 4-1. Detailed Functional Block Diagram, 2515 Computer-Aided Test System

4.2 BUS STRUCTURE

Six buses are used throughout the system. These are the LSI-11 Q-bus, the pseudo-Unibus, channel bus, analog FIFO control bus, 2501 microprocessor bus, and the microinstruction bus. The following sub-paragraphs describe the buses and how they interact to accomplish required tasks.

4.2.1 2501 Microprocessor Bus

The microprocessor bus is a synchronous bus performing at times relative to the master clock, a clock pulse occurring once each bus cycle (250 ns to 1000 ns). The bus cycles are an integral number of 62.5 ns clock pulses in length. The microprocessor bus performs the following three major functions:

1) Data transfers (16 bit word) are done from one bus device to another as commanded by the microprocessor. A source device is selected by the source field in the microinstruction and a destination device is selected by the destination field in the microinstruction. The source and destination devices decode their selection during the microinstruction prior to the microinstruction in which the data transfer takes place. Readout of microinstructions is always one instruction cycle ahead of the execution of the microinstruction. The source device puts the data on the bus at the beginning of the instruction cycle and holds the data on the bus for the duration of the instruction cycle. At the end of the instruction cycle the destination device receives the data by clocking it into a register with the trailing edge of the master clock pulse.

2) The second major function of the microprocessor bus is to provide a means by which devices on the bus may cause an interrupt in the microprocessor. The device first makes an interrupt request at one of seven levels of priority by enabling one of seven interrupt request lines. The interrupt request lines are open collector ORed. More than one device can request an interrupt on the same line at the same time. When priorities permit, the microprocessor enables one of seven interrupt grant lines. The interrupt grant lines are chained so that only the first requestor on an interrupt grant line sees the grant. On the cycle following the grant the remote device enables eight interrupt vector lines to specify to the microprocessor the address of the interrupt vector in the microprocessor's control store. The microprocessor is then interrupted, executes the instruction stored at the interrupt vector location and saves its return address in the interrupted program.

3) The third major function of the microprocessor bus is to provide a means for a remote device to signal its condition to the microprocessor. The condition is either true or false and the microprocessor jump instructions are controlled by the state of the condition. Thus the microprocessor can do conditional jumps depending upon a condition

signaled by a remote device on its bus. The condition is signaled by a single line, the condition true line. The device which is to send its condition is selected by the source field (also called the condition field) of the microinstruction.

There are four types of bus interconnection lines used: 1) tri-state, 2) open collector, 3) transmission line, and 4) totem pole, or regular TTL output.

Tri-state lines are used for the transmission of data. Generally a tri-state driver enables the line at some time following the clock and the receiving device takes the data from the bus at the next clock edge. The important characteristic for these lines are the enable and disable times of the driver and the bus charge time. The bus is thought of as a lump capacitance which must be charged by the driver. The actual situation is more complicated, a weakly driven transmission line with multiple reflections, but a lumped capacitance model is used as an approximation.

Open collector lines are used where an ORing of several different drives is desired, as in the interrupt request lines, or where the slow rise time of open collector lines can be tolerated, as in the master clock delay control lines. The open collector lines have a 390 ohm pullup resistor and are driven by U39 or by U11, U13, U15 and U64 with the inputs wired to the enable.

Transmission lines are used for only two lines, the master clock (-MCOUT) and the 16 MHz oscillator (OSCOUT). These lines are driven by U99 and U102, 50-ohm line drivers located on the microprocessor board. Unless the total backplane length is very short these lines must have a controlled inpedance and very short stubs leading from the bus to boards. The line could be a 100 ohm line terminated with a 100 ohm resistor on the backplane or the last board on the line, or if the line runs in both directions there could be a 100 ohm resistor at both ends, fully utilizing the 50 ohm drive capability. The master clock is a transmission line to minimize clock skew and to prevent ringing. The OSCOUT line is a transmission line to prevent ringing and preserve pulse shape.

Totem pole, or regular TTL output, is used for lines where there is no common bussing and moderate speed requirements. This includes the interrupt grants, which are chained, and a few other lines.

4.2.1.1 Bus Line Signal Names, Functions and Backplane Connector Pins.

DBO	(C2J)	Data line, 16-bit, least significant.
DB1	(C1K)	
DB2	(C2K)	
DB3	(C1L)	
DB4	(C2L)	
DB5	(C1M)	
DB6	(C2M)	
DB7	(C1N)	
DB8	(C2N)	
DB9	(C1P)	
DB10	(C2P)	
DB11	(C1R)	
DB12	(C2R)	
DB13	(C1S)	
DB14	(C2S)	
DB15	(C2T)	Data line, 16-bit, most significant.

OSCOUT (C1A) 16 MHz oscillator (transmission line). Master clock. It is a negative going pulse of length 62.5 nanoseconds which repeats with a period between 250 nanoseconds and 1000 nanoseconds, depending upon the microinstruction being executed.

CNDTRU (C2D) Condition true line. Used to signal a condition from a remote device selected by the source field to the microprocessor. This is a tri-state line. The microprocessor jump instructions, jump or not jump, depends on the condition signaled. The polarity of the condition is arbitrary, since the software can reverse the sense of the jump. Design capacitance is 200 pF.

INITL (C1D) Initialize line. When held low for a few microseconds it resets the microprocessor and other devices connected to the system. The master clock (MCOUT) does not stop while this line is held low. The initialize signal is driven from the LSI-11/23 processor board via the Q-Bus. The pulse is about 1.0 microseconds long and is derived from the LSI-11/23 initialize pulse (about 10 μsec long). This is an open collector line, with a pullup on the microprocessor board. Thus it may be driven from different locations. When the initialize ends, the microprocessor starts executing the instruction at location zero in the control store.

Six lines are used to control the period of the master clock. These are SCM0-SCM2 and DCM0-DCM2. The master clock has a period which is an integral number of 62.5 nanoseconds. It can never be less than 4 clocks, or 250 nanoseconds. For the purpose of computing the actual period, the period is considered to be 2 clocks long. Additional clocks are added depending upon which of the six lines are pulled low. If the total number of clocks thus computed is 4 or less, the period will be 4 clocks. If it is greater than 4, then the period will be the number of clocks computed. The SCM lines are for the use of a device sourcing to the data bus and they have values of 1, 2 and 4 clocks. Three DCM lines are for the use of the destination device and also have values in binary steps. The most clocks that can be

added are 7 for the source and 7 for the destination, which added to the 2 base clocks, make a maximum period of 16 clocks, or 1 microsecond. These are open collector lines with a capacitance of 200 pF.

SCM0	(C2E)	Add 1 clock for source device
SCM1	(C1F)	Add 2 clocks for source device
SCM2	(C2F)	Add 4 clocks for source device
DCM0	(C1H)	Add 1 clock for destination device
DCM1	(C2H)	Add 2 clocks for destination device
DCM2	(C1J)	Add 4 clocks for destination device

The following 18 lines are the source and destination lines. They are tri-state and come directly from the control store ROM outputs.

S0	(D1A)	Source, least significant
S1	(D1B)	
S2	(D2B)	
S3	(D1C)	
S4	(D1D)	
S5	(D2D)	
S6	(D1E)	
S7	(D2E)	
S8	(D1F)	Source, most significant
D0	(D2F)	Destination, least significant
D1	(D1H)	
D2	(D2H)	
D3	(D1J)	
D4	(D2J)	
D5	(D1K)	
D6	(D2K)	
D7	(D1L)	
D8	(D2L)	Destination, most significant

The following 8 lines are the interrupt vector lines. These are tri-state and are used by a remote device to specify the address of the interrupt vector in the control store.

IV0	(D1M)	Interrupt vector, least significant
IV1	(D2M)	
IV2	(D1N)	
IV3	(D2N)	
IV4	(D2P)	
IV5	(D2P)	
IV6	(D1R)	
IV7	(D2R)	Interrupt vector, most significant

The following 7 lines are the interrupt request lines. They are open collector and are pulled low by a remote device to request an interrupt. 7 is the highest priority, 1 the lowest.

IR1	(D1S)	Interrupt request, lowest priority
IR2	(D2T)	
IR3	(D2V)	
IR4	(D2V)	
IR5	(E1B)	
IR6	(E1C)	
IR7	(E2D)	Interrupt request, highest priority

The following 14 lines are the interrupt grant lines in and out. Each board passes the interrupt grant from the in to the out pin, unless it is using the grant. These are totem pole lines.

IG1in	(F1A)	Lowest priority grant in
IG1out	(F1B)	Lowest priority grant out
IG2in	(F1C)	
IG2out	(F1D)	
IG3in	(F1E)	
IG3out	(F1F)	
IG4in	(F1H)	
IG4out	(F1J)	
IG5in	(F2B)	
IG5out	(F2D)	
IG6in	(F2E)	
IG6out	(F3F)	
IG7in	(F2H)	Highest priority grant in
IG7out	(F2J)	Highest priority grant out

4.2.2 Microinstruction Bus

This bus is actually part of the microprocessor bus but is described separately here for purposes of clarity. The microinstruction bus is used solely for communications between the CPU board and the PROM board. It comprises 32 data line outputs from program memory and 16 address line inputs from the CPU instruction counter. Besides the microinstructions (bits 0-8 and 27-31), the PROM sends 18 lines of source and destination data (bits 9-26) from its control store. Both input and output lines are tri-state. The data and address lines are listed below by bit, signal name, and edge connector pin.

Data Bit	Signal	Pin	Data Bit	Signal	Pin
0	ROMDAT0	E2E	16	D7	D1L
1	ROMDAT1	E1F	17	D8	D2L
2	ROMDAT2	E2F	18	S0	D1A
3	ROMDAT3	E1H	19	S1	D1B
4	ROMDAT4	E2H	20	S2	D2B
5	ROMDAT5	E1S	21	S3	D1C
6	ROMDAT6	E2S	22	S4	D1D
7	ROMDAT7	E1K	23	S5	D2D
8	ROMDAT8	E2K	24	S6	D1E
9	D0	D2F	25	S7	D2E
10	D1	D1H	26	S8	D1F
11	D2	D2H	27	ROMDAT27	E1L
12	D3	D1J	28	ROMDAT28	E2L
13	D4	D2J	29	ROMDAT29	E1M
14	D5	D1K	30	ROMDAT30	E2M
15	D6	D2K	31	ROMDAT31	E1N

Address Bit	Signal	Pin	Address Bit	Signal	Pin
0	IC0	E2N	8	IC8	E1U
1	IC1	E1P	9	IC9	E2U
2	IC2	E2P	10	IC10	E1V
3	IC3	E1R	11	IC11	E2V
4	IC4	E2R	12	IC12	F1A
5	IC5	E1S	13	IC13	F1B
6	IC6	E2S	14	IC14	F2B
7	IC7	E2T	15	IC15	F1C

4.2.3 LSI-11 Q-Bus

The LSI-11 Q-Bus is manufactured by Digital Equipment Corporation. Refer to the appropriate vendor manual listed in appendix A for detailed information on the Q-bus. A brief description is presented here.

The Q-bus consists of 36 bidirectional and 2 unidirectional signal lines along which the LSI-11/23 Processor, memory, and I/O devices communicate. The Q-bus interfaces the Processor with the Serial Line Interface, Winchester/Floppy Interface, and the optional DSD 440 Floppy Interface. The LSI-11/23 Processor is able to communicate via the Q-bus with the Unibus memory and the 2501 microprocessor by utilizing the Q-bus interface on the Memory Controller board.

The Q-bus lines have TTL inputs and outputs and are as follows:

	Signal	Pin
Data and address lines	BDAL0	E2U
	BDAL1	E2V
	BDAL2	F2E
	BDAL3	F2F
	BDAL4	F2H
	BDAL5	F2S
	BDAL6	F2K
	BDAL7	F2L
	BDAL8	F2M
	BDAL9	F2N
	BDAL10	F2P
	BDAL11	F2R
	BDAL12	F2S
	BDAL13	F2T
	BDAL14	F2U
	BDAL15	E1B
	BDAL16	E1A
	BDAL17	E1B
Data transfer control lines	BBS7	E2P
	BDIN	E2H
	BDOUT	E2E
	BRPLY	E2F
	BSYNC	E2J
	BWTBT	E2K
Direct memory access control lines	BDMG	E27
	BDMR	E1N
	BSACK	F1N
Interrupt control lines	BEVNT	F1R
	BIAK	E2N
	BIRQ4	E2L
	BIRQ5	E1A
	BIRQ6	E1B
	BIRQ7	F1P
System control lines	BDCOK	F1A
	BHALT	E1P
	BINIT	E2T
	BPOK	F1B
	BREF	E1R

4.2.4 Pseudo-Unibus

This bus is a bidirectional bus used to interface the Unibus memory with the rest of the system. Memory access is arbitrated by the Memory Controller. The bus consists of control lines -MSYN and -SSYN, C0 and C1, data lines UD0-UD15, and address lines A0-A20. These lines are listed below by signal name and connector pin.

	Signal	Pin
Control Lines	MSYN	B1V
	SSYN	B1U
	C0	B2U
	C1	B2T
Data Lines	UD0	A1C
	UD1	A2D
	UD2	A1D
	UD3	A2E
	UD4	A1E
	UD5	A2F
	UD6	A1F
	UD7	A2H
	UD8	A1H
	UD9	A2J
	UD10	A1J
	UD11	A2K
	UD12	A1K
	UD13	A2L
	UD14	A1L
	UD15	A2M
Address Lines	A0	B2H
	A1	B1H
	A2	B2J
	A3	B1J
	A4	B2K
	A5	B1K
	A6	B2L
	A7	B1L
	A8	B2M
	A9	B1M
	A10	B2N
	A11	B1N
	A12	B2P
	A13	B1P
	A14	B2R
	A15	B1R
	A16	B2S
	A17	B1S
	A18	B2E
	A19	B1E
	A20	A1P

The control signals for the Unibus memory are derived from the Q-bus equivalents of these same control signals. Master sync (-MSYNC) is derived from Q-bus signals -BDIN, -BDOUT, and -BSYNC. The slave sync (-SSYN) from the memory is converted to -BRPLY for the Q-bus master device. Control bits, C0 and C1, determine read/write and word/byte operations as derived from Q-bus signals -BDIN and -BWTBT.

The sixteen lines of data UD0-UD15, are sent to or from the Unibus Memory via the Memory Controller for the device location being serviced.

Address lines A0-A21 are used to select the location at which to read or write data depending on the condition of control lines C0 and C1.

4.2.5 Analog FIFO Control Bus

This bus is used by the Channel Control bus to send eleven lines of status and address signals to the Memory Controller. The Memory Controller monitors the status of the FIFO control bus to determine whether the FIFO device should have access to the main memory factor and stores the overload bits for access by the 2501 microprocessor. The signals carried on the analog FIFO control bus are as follows:

	Signal	Pin
Address Lines	FARA0	D2S
	FARA1	D2T
	FARA2	D2U
	FARA3	D2V
Status Lines	FS0	C2U
	FS1	C2V
	TRIG	C1V
	FIFOCYC	D1U
	CKFIFO	D1S

4.2.6 Channel Bus

The channel bus is a bidirectional bus which interfaces the Channel Control board with up to four 25 kHz channel boards. Through the channel bus, the Channel Control board is able to provide control, timing, and data buffering to the channel boards. Computation results and overload conditions are sent to the control board via the channel bus.

	Signal	Pin
Stage Select Lines	K0	B2N
	K1	B1P
	K2	B2P
	K3	B1R

	Signal	Pin
Serial Data Lines	SERCOS	B2E
	SERSIN	B1E
	SERCON	B2D
Overload Control	CLROVLD	E1V
Board Select	BDSEL1	E2S
	BDSEL2	E2R
	BDSEL3	E2P
	BDSEL4	E2N
Filter Subperiod Selects	FLTX0	B1F
	FLTX0	B2F
Subperiods	SPAD0	B2S
	SPAD1	B1U
	SPAD2	B2T
	SPAD3	B2U
Channel Data Lines	CHDB00	A2M
	CHDB01	A1N
	CHDB02	A2N
	CHDB03	A1P
	CHDB04	A2P
	CHDB05	A1R
	CHDB06	A2R
	CHDB07	A1S
	CHDB08	A2S
	CHDB09	A2T
	CHDB10	A1U
	CHDB11	A2U
	CHDB12	A1V
	CHDB13	A2V
	CHDB14	B1A
Miscellaneous Control	ABC0	B1S
	ABC1	B2K
	AECLR	B2L
	AERND	B1j
	CHINIT	A2B
	CNUT	E1J
	DILOUT	B2K
	LDOUT	B1L
	LDSRA	B1H
	LDSRB	B2H
	RDAD	B1K
	RGB0	E2H
	RGB1	E2F
	RGB2	E1H
	RGDTA	E2J

	Signal	Pin
	SPOUT	B1V
	SRWRT	B2M
	SUBPHE	B1N
	TWSCLR	B1B
	TWDEN	B1C
	TWSCLK	B2B
	MUWRT	E1C
	WRTCLK	E2T
Memory Address Lines	MADR0	E1B
	MADR1	E1F
Board Read	BDRD1	E1S
	BDRD2	E1R
	BDRD3	E1P
	BDRD4	E1N

4.3 FRONT PANEL PCB ASSEMBLY

The circuitry contained on Control Panel Boards 1 and 2 of this assembly is described in the following text and supported by the diagrams listed below:

Functional Block Diagram, Control Panel Boards 1 and 2	Figure 4-2
Schematic Diagram, Control Panel Board 1	2515-4707-2D
Schematic Diagram, Control Panel Board 2	2515-4708-2D

4.3.1 Control Panel Boards 1 and 2

Control Panel Boards 1 and 2, performing together, serially transmit switch locations and receive LED and tone generator data. They interface with the Serial Interface Board using port P1. Transmissions are accomplished on Port 1 by the use of RS-422 compatible line drivers and receivers. Serial data operates at 300 baud and is comprised of one start bit, eight data bits (no parity bit), and two stop bits.

Board 1 decodes and transmits the selected switch locations to Board 2. It receives 7 bits of timing and control from Board 2. Its output, -MUXOUT and +MUXOUT, is transmitted via connector J1 which connects to Board 2 by an interface cable.

Board 2 contains a burst mode, clock circuit, tone generator and UART U17 to perform transmit and receive functions. Its outputs are connected to Board 1 by J2 and to the Serial Interface Board via port P1.

4.3.1.1 Initialize. A power up initialize circuit resets the receiver/transmitter (UART U17), LEDs, and tone generator. An LM323K voltage regulator converts the +12 volts to +5 volts.

4.3.1.2 Clock Circuit. The clock is provided by a 1.2288 MHz crystal oscillator U20 which drives two 4-bit counters, U21 and U18. These counters generate the support clocks and data lines (C4-C6) used in both the transmit and receive circuits.

4.3.1.3 Transmitter Function. Only 79 of the 101 available pushbutton switches are used in the standard system. The logic is designed, however, to accommodate the entire capability should future expansion be desired. The transmitting circuit uses two 3 to 8 line multiplexers (U11, U12) and one 1-of-8 line decoder (U3) to convert 7 bits of parallel pushbutton data, DB1-8, to a serially encoded 10-bit character. The DLV11-J, port 1, register assignments are as follows:

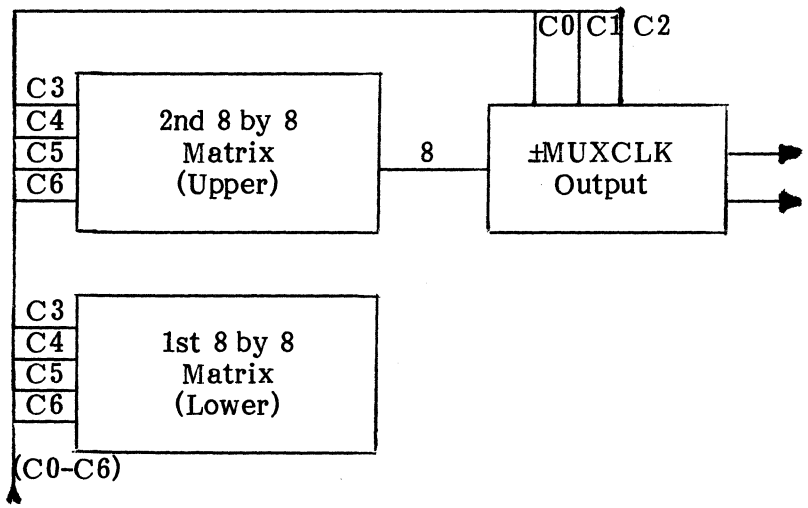
775640	Receive Status Register
775642	Receive Buffer Register
775644	Transmit Status Register
775646	Transmit Buffer Register

The transmitted data byte field is defined as two 8 by 8 switch matrixes, using the section bit (DB7) to determine in which matrix the button has been pressed.

C0-C2 Scan columns 0 thru 7

C3-C5 Scan rows A thru S

C6 Select upper or lower switch matrix



1 scan cycle is completed every 53.3 ms (18.75 Hz).

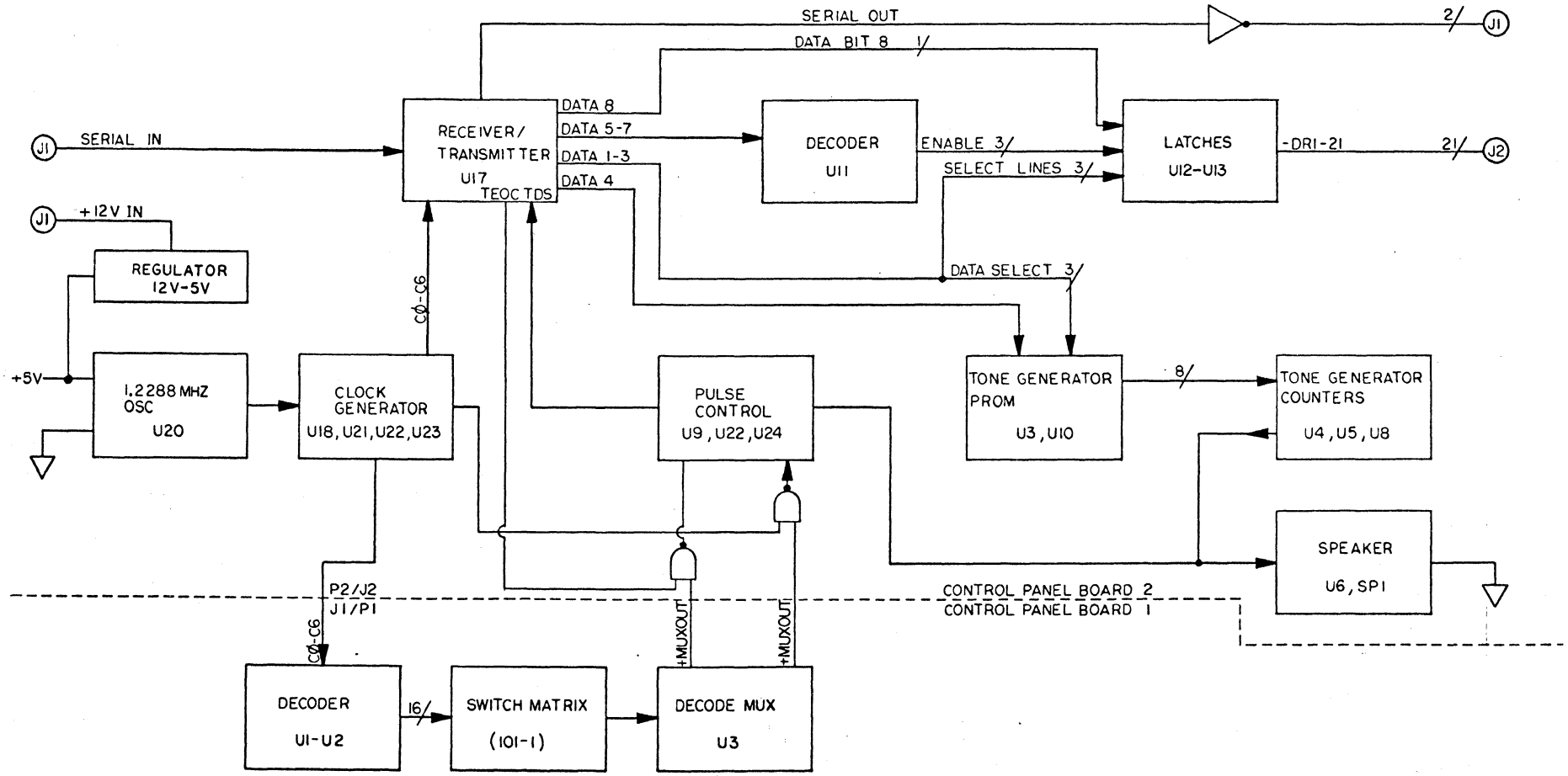


Figure 4-2. Functional Block Diagram,
Control Panel Boards 1 and 2

The transmitted data byte is further divided by vertical columns, horizontal rows, and ground. The transmitter data octal address is 775642. A logic 0 on bit DB7 indicates the section of pushbutton rows A through H. A logic 1 on bit DB7 indicates the section of pushbutton rows J through S.

When a pushbutton is pressed, the switch selection is transmitted via data bits DB1-DB7. Switches are selected by section, row, and column in that order. The bit assignments are shown below. Table 4-1 lists the front panel switches.

DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1
GND	SEC	ROW3	ROW2	ROW1	COL3	COL2	COL1

4.3.1.4 Data Select Output -MUXOUT. In the idling state both -MUXOUT (output Y of U3) and U17 UART's TEOC (transmit end of character) output are high. The two signals are NANDed together producing -CLR, which clears both the enable and delay flip-flops U24. TEOC will go low when a character is transmitted. When a front panel button is pressed, it causes a high to low transition on -MUXOUT and -CLR goes high allowing the enable flip-flop to be clocked by -ECLK. Once the enable flip-flop has been clocked, pushbutton scanning is halted. When the pushbutton is released, -MUXOUT again goes high and at the end of the character transmission TEOC goes high. This state makes -CLR low, thus clearing the enable flip-flop. Pushbutton scanning then resumes.

4.3.1.5 Data Select Output +MUXOUT. When a pushbutton is pressed, a low to high transition occurs on +MUXOUT (an output of U3). +MUXOUT is NANDed with two counter outputs (from U21 and U18) producing a 105 microsecond negative pulse. This pulse is inverted to produce -ECLK. The falling edge clocks the enable flip-flop (at pin 1) disabling pushbutton scanning. -ECLK is also NANDed with TBMT (transmit buffer empty signal of UART U17) and the output of U23 NAND gate to produce -TDS. -TDS loads the data present at DB1-DB8 and TBMT goes low. The falling edge of -TDS triggers a one-shot (U9) with a one-second delay. When the pushbutton is released, scanning resumes.

4.3.1.6 Transmitter Burst Mode. The transmitter section will go into burst mode after an approximately 750 milliseconds to 1 second time delay has elapsed and if a pushbutton is still depressed. The falling edge of -DLY (U9 output) clocks the delay flip-flop of U24. The delay output (pin 7) and the enable output (pin 5) are NANDed together to produce +GATE(HI). The output of (delay flip-flop) U24, pin 7, and also (-ECLK) must be high for TBMT to generate -TDS. Everytime the transmit buffer is empty TBMT is generated and passed on as -TDS. Also, in burst mode -TDS triggers and retriggers, the -DLY length gets longer with every trigger (-TDS). Each time -TDS is gated, it triggers the one-shot. If the pushbutton remains depressed, each retrigger that occurs increases its delay by the period between triggers. The -DLY signal remains high until the button is released. When the pushbutton has been released, retriggering ceases.

Table 4-1. Front Panel Switches

LOCATION (SWITCH NO.)	OCTAL VALUE	SECTION	NAME
SA0 (S101)	000	OUTPUT	Not Labeled
SA1 (S90)	001	OUTPUT	PLOT
SA2 (S79)	002	OUTPUT	Output signal
SB0 (S100)	010	OUTPUT	Spare
SB1 (S89)	011	OUTPUT	Spare
SB2 (S78)	012	OUTPUT	Spare
SC0 (S99)	020	OUTPUT	CRT BRT
SC1 (S88)	021	OUTPUT	Spare
SC4 (S8)	024	VIEW SETUP	Y SCALE
SC5 (S10)	026	ANALYSIS	CONT
SC7 (S9)	027	ANALYSIS	RUN
SD0 (S98)	030	SETUP	STATE
SD1 (S87)	031	SETUP	CHANS
SD2 (S77)	032	SETUP	FREQ
SD4 (S14)	034	VIEW SETUP	Spare
SE0 (S97)	040	SETUP	MODE
SE1 (S86)	041	SETUP	Spare
SE2 (S76)	042	SETUP	TRIG
SE4 (S22)	044	DISPLAY	BLOCK MATH
SE6 (S23)	046	ANALYSIS	ARM
SE7 (S24)	047	ANALYSIS	HOLD
SF0 (S96)	050	MEASUREMENT	WINDOW
SF1 (S85)	051	SETUP	Spare
SF2 (S75)	052	SETUP	LEVEL
SF4 (S33)	054	ENTRY	Spare
SF5 (S34)	055	ENTRY	- and E
SF6 (S35)	056	ENTRY	SPC and F
SF7 (S36)	057	ENTRY	SHIFT
SG0 (S95)	060	SETUP	AVERAGE
SG1 (S84)	061	SETUP	Spare
SG2 (S74)	062	SETUP	CHNL SETUP
SG4 (S44)	064	ENTRY	Spare
SG5 (S45)	065	ENTRY	+ and J
SG6 (S46)	066	ENTRY	↓ and K
SG7 (S47)	067	ENTRY	X and L

Table 4-1. Front Panel Switches (continued)

LOCATION (SWITCH NO.)	OCTAL VALUE	SECTION	NAME
SH0 (S91)	070	STORAGE	RECALL
SH1 (S92)	071	STORAGE	Spare
SH2 (S93)	072	STORAGE	Spare
SH3 (S94)	073	STORAGE	NEXT↓
SH4 (S55)	074	ENTRY	Spare
SH5 (S56)	075	ENTRY	X and P
SH6 (S57)	076	ENTRY	+/- and Q
SH7 (S58)	077	ENTRY	Y and R
SJ0 (S80)	100	STORAGE	STORE
SJ1 (S81)	101	STORAGE	Spare
SJ2 (S82)	102	STORAGE	Spare
SJ3 (S83)	103	STORAGE	NEXT
SJ4 (S66)	104	ENTRY	Spare
SJ5 (S67)	105	ENTRY	÷ and V
SJ6 (S68)	106	ENTRY	= and W
SJ7 (S69)	107	ENTRY	Z and :
SK0 (S70)	110	STORAGE	SELECT
SK1 (S71)	111	STORAGE	DIR
SK2 (S72)	112	STORAGE	Spare
SK3 (S73)	113	STORAGE	NEXT↑
SK4 (S62)	114	ENTRY	O and S
SK5 (S63)	115	ENTRY	. and T
SK6 (S64)	116	ENTRY	, and U
SK7 (S65)	117	ENTRY	ENTRY
SL0 (S59)	120	CURSOR	SDBD
SL2 (S60)	122	CURSOR	SDBD <
SL3 (S61)	123	CURSOR	SDBD >
SL4 (S51)	124	ENTRY	1 and M
SL5 (S52)	125	ENTRY	2 and N
SL6 (S53)	126	ENTRY	3 and O
SL7 (S54)	127	ENTRY	DEL
SM7 (S48)	130	CURSOR	HARMONIC
SM2 (S49)	132	CURSOR	STORE
SM3 (S50)	133	CURSOR	RECALL
SM4 (S40)	134	CURSOR	4 and G
SM5 (S41)	135	ENTRY	5 and H
SM6 (S42)	135	ENTRY	6 and I
SM7 (S43)	137	ENTRY	CLR

Table 4-1. Front Panel Switches (continued)

LOCATION (SWITCH NO.)	OCTAL VALUE	SECTION	NAME
SN0 (S37)	140	CURSOR	SINGLE
SN2 (S38)	142	CURSOR	SET REF
SN3 (S39)	143	CURSOR	REL/ABS
SN4 (S29)	144	ENTRY	7 and A
SN5 (S30)	145	ENTRY	8 and B
SN6 (S31)	146	ENTRY	9 and C
SN7 (S32)	147	ENTRY	EE and D
SP0 (S25)	150	CURSOR	< CURSOR
SP1 (S26)	151	CURSOR	Spare
SP2 (S27)	152	CURSOR	> CURSOR
SP3 (S28)	153	CURSOR	Spare
SP4 (S19)	154	DISPLAY	NEXT↓
SP6 (S20)	156	DISPLAY	CHANNEL PAIR
SP7 (S21)	157	DISPLAY	UNITS
SR0 (S15)	160	DISPLAY	FULL
SR1 (S16)	161	DISPLAY	SPLIT
SR2 (S17)	162	DISPLAY	OVLP
SR3 (S18)	163	DISPLAY	SWITCH UP/DOWN
SR4 (S11)	164	DISPLAY	NEXT PAIR
SR6 (S12)	166	VIEW SETUP	Spare
SR7 (S13)	167	VIEW SETUP	Spare
SS0 (S1)	170	DISPLAY	REAL TIME
SS1 (S2)	171	DISPLAY	AVG
SS2 (S3)	172	DISPLAY	MEM
SS3 (S4)	173	DISPLAY	NEXT
SS4 (S5)	174	DISPLAY	NEXT↑
SS6 (S6)	176	DISPLAY	FUNCTION
SS7 (S7)	177	DISPLAY	X Scale

4.3.1.7 Receiver. The receiver part of UART U17 accepts an encoded serial character and converts it to eight bits of parallel data, RD1-RD8. These eight bits are used to illuminate the front panel LEDs for the various functions and to enable the tone generator (see paragraph 4.3.1.8). Twenty-one LED capabilities are available, however only fourteen are used in the standard system. Refer to table 4-2. The receiver data address field is shown below.

RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1
DATA	ENA3	ENA2	ENA1	ADD4	ADD3	ADD2	ADD1

Once the data is present at RD1-RD8 (data holding register is filled), RDA goes high, then is inverted. This enables -RDAR which resets the data available flip-flop and gates the 3 to 8 line selector U11. Three bits, RD5-RD7, are used as enable lines to select which of the three addressable latches (U12-U14) to gate (to illuminate the LEDs) or to load data into the tone generator address register. Three bits of address, RD1-RD3 RD4, are common to the latches. RD8 is the data line for all LEDs. Latches U12-U14 are used to turn the LEDs on and off. The LEDs are addressed as follows:

Table 4-2. Front Panel LEDs

LED	Enable Octal Value	Disable Octal Value	Switch	Name
D1	200	000	S1	REALTIME
D2	201	001	S2	AVG
D3	202	002	S3	MEM
D4	203	003	S10	RUN
D5	204	004	S15	SINGLE
D6	205	005	S16	DUAL
D7	206	006	S17	OVLY
D8	207	007	S23	ARM
D9	220	020	S24	HOLD
D10	221	021	S36	SHIFT
D11	222	022	S37	MAIN
D12	223	023	S48	HARMONIC
D13	224	024	S59	SIDEBAND
D14	225	025	REMOTE	REMOTE

Note that an illuminated LED will remain on until the UART receives an explicit disable. For example: To Illuminate LED CR8 (D8), the UART must receive an octal 207. To disable CR8 the UART must receive an octal 007. Octal code 100 will reset all illuminated LEDs (and tone generator, if on at the same time).

4.3.1.8 Tone Generator. The tone generator is comprised of six ICs (U3-6, 10), one-half of U9 monastable multivibrator, and one speaker, SP1. Tone data is kept in 32 x 8 PROM U3 and uses 16 locations to produce 15 different tones ranging from 634 Hz to 1.422 kHz. This range covers more than one octave. The tone generator's on time periods are totally software controlled.

The PROM's chip select (pin 15) is grounded so that any positive transition on the address register clock line will latch the data. Address bit 5 (pin 14) is also grounded so that only locations 0-15 are accessed. The address data, RD1-RD4, from the UART is stored in register U10 and is latched into the PROM address register by the positive edge of a pulse selected by RD5 and RD6.

The tone generator code uses six of the UART's eight address bits. Four bits, RD1-RD4, are used as tone address bits. Bits RD5 and RD6 are the tone enables. Bits RD7-RD8 are not used. The tone generator address field is shown below.

RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1
NOT	USED	TEN2	TEN1	ADD4	ADD3	ADD2	ADD1

The tones are addressed as follows:

Tone	Base Freq	Enable Octal Value	Disable Octal Value	Tone Freq
0	153.6 kHz	060	060	000. Hz
1	153.6 kHz	061	060	317.35 Hz
2	153.6 kHz	062	060	347.5 Hz
3	153.6 kHz	063	060	357.2 Hz
4	153.6 kHz	064	060	378.3 Hz
5	153.6 kHz	065	060	400. Hz
6	153.6 kHz	066	060	424.8 Hz
7	153.6 kHz	067	060	449.12 Hz
8	153.6 kHz	070	060	477.01 Hz
9	153.6 kHz	071	060	505. Hz
10	153.6 kHz	072	060	533.3 Hz
11	153.6 kHz	073	060	564.7 Hz
12	153.6 kHz	074	060	600. Hz
13	153.6 kHz	075	060	634.7 Hz
14	153.6 kHz	076	060	673.7 Hz
15	153.6 kHz	077	060	711.1 Hz

Once the data is latched, the PROM contents are present on U4 and U5 counter inputs. The counters count up to 15, then produce a ripple-carry-out (RCO). Counter U4's RCO output performs two functions: 1) it enables counter U5 to count, and 2) both counters' RCOs are gated together to produce a positive edge to clock flip-flop U6. To clock the flip-flop, the count, from the preloaded values in PROM, goes to 255. Then, when both U4 and U5 RCOs are high, the output clocks U6.

Once the PROM has been accessed, the tone generator circuitry continues to generate a square wave output which is ANDed with U9's one-shot pulse to feed inverter U15. It is the positive going pulse (30-40 nanoseconds) from U9 that gates the tone output to speaker SP1.

During power up, the PROM address register is cleared and the data addresses location 0 (content octal 377). This sets all the counter inputs and RCO outputs to high. There is no square wave output. On power up, the PROM stays at location 0 until it receives a valid enable code (061-077).

4.4 SERIAL LINE INTERFACE BOARD

The serial line interface used in this system is Model PM-DLV11J and is manufactured by Digital Equipment Corporation. It is a 4-channel, asynchronous, serial line interface between the LSI-11/23 Q-Bus and standard I/O devices, in this case, the front panel PCB assembly.

The interface receives parallel data from the Q-Bus, converts it to a serial word and transmits it to the peripheral device. It also receives the 10-bit serial data word from the front panel PCB assembly (Control Panel Boards 1 and 2) and converts it to parallel data for output to the LSI-11/23 Q-Bus.

Refer to the vendor manual listed in Appendix A for additional details.

4.5 WINCHESTER/FLOPPY INTERFACE BOARD

The Model WINC05 is manufactured by Advanced Electronics Design, Inc., and provides DEC RX02 emulation for the 5¼-inch floppy drive at a 0.5 megabyte capacity and RL01 emulation for the Winchester at 5.0 megabyte capacity (for each of two RL01 devices).

The WINC05 has standard DEC address, vector, interrupt priority, alternate address and vectors, bootstrap, and 22-bit addressing. The bootstrap allows the user to boot using a console device, auto-boot on power up, and disable the bootstrap.

More details and discussions are found in the vendor manual listed in Appendix A.

4.6 LSI-11/23 PROCESSOR BOARD

The LSI-11/23 processor board is manufactured by Digital Equipment Corporation. The LSI-11/23 controls the 2515 system through the Q-Bus interface lines on the memory controller board. The LSI-11/23 is a 16-bit, asynchronous, high performance processor which functions using 3 chips: data, control, and memory management. In addition, it's capabilities include the DEC KEF -11A floating point consisting of two chips.

Through the Q-Bus lines the processor interfaces with the 2501 microprocessor (CPU board) and the main memory (unibus board). It interfaces with the control panel boards 1 and 2 via the serial interface board.

More complete information can be obtained from the vendor manual listed in Appendix A.

4.7 MEMORY CONTROLLER BOARD

The following text is supported by the diagrams listed below:

Block Diagram, Memory Controller Board	Figure 4-3
Schematic, Memory Controller Board	2515-4702-2D

The memory controller board arbitrates memory accesses among three ports. These are the Q-bus, microprocessor bus, and analog input FIFO control bus ports.

The only control the microprocessor has over the priority given to the three ports is the ability to disable output from the FIFO. FIFO status signals also affect the priority structure. The memory controller board also contains Q-bus slave and interrupt vector interfaces. These allow the microprocessor to interrupt the LSI-11/23 and act as a slave for three sets of Q-bus addresses (jumpered). This is used to emulate Q-bus devices.

Q-Bus Port

One of the functions of the memory controller board is to make the Unibus memory look like Q-bus memory. Two microprocessor registers, MAPMEM and OFFMEM, are used to define a simple mapping of the LSI-11/23 address space into the address space of the Unibus memory.

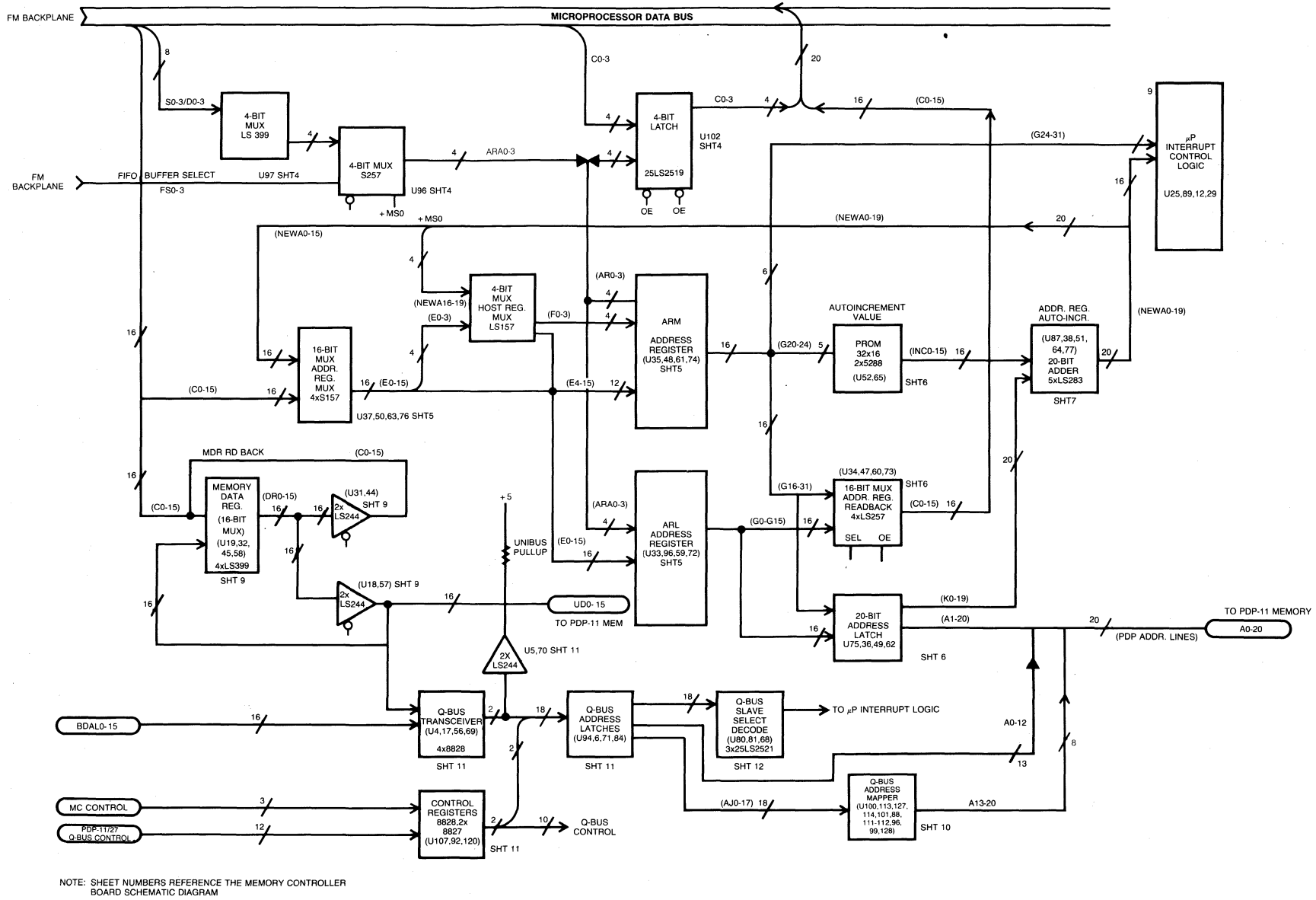
This board interfaces 18 address bits of the LSI-11 Q-bus with the 21 address bits of the Unibus memory. The board interface supports 21 of the 22 address bits allowed (A0-A20) on the Pseudo-Unibus.

Once the mapping registers are set up by the LSI-11/23, any Q-bus device may do reads and writes from the Unibus memory in the same manner as for real Q-bus memory. The LSI-11/23 memory mapping hardware can be used to access memory areas up to the 18-bit address limit. Memory access beyond this limit is only possible through the use of MAPMEM and OFFMEM or microprocessor initiated operations.

Microprocessor Port

A memory data register (MDR) and 16 memory address registers (MAR) are provided to allow the 2501 microprocessor (CPU board) to access the Unibus Memory. Each address register contains a 20-bit address, A1 through A20. A0 is always assumed to be zero during a microprocessor access.

Two 16-bit registers, ARM and ARL comprise each MAR. The MAR to be accessed is determined by control bits in the command register MCMD. Each memory access is accomplished by reading or writing one of 16 microprocessor registers called access initiators (AI). The AIs use address and address increment information contained in their corresponding MARs. These registers are called access initiators because each access to this register initiates a read/write



NOTE: SHEET NUMBERS REFERENCE THE MEMORY CONTROLLER BOARD SCHEMATIC DIAGRAM

Figure 4-3. Block Diagram, Memory Controller Board

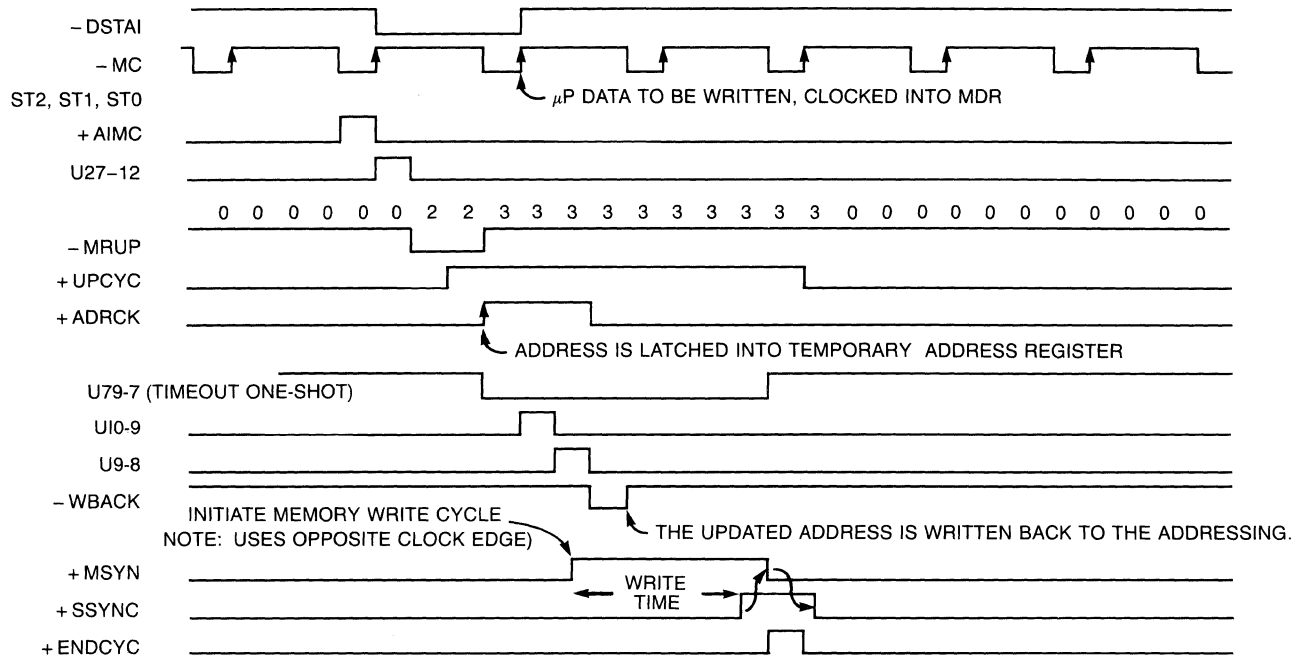


Figure 4-4. Timing Diagram, Microprocessor Write Cycle (MOV R0, AI0)

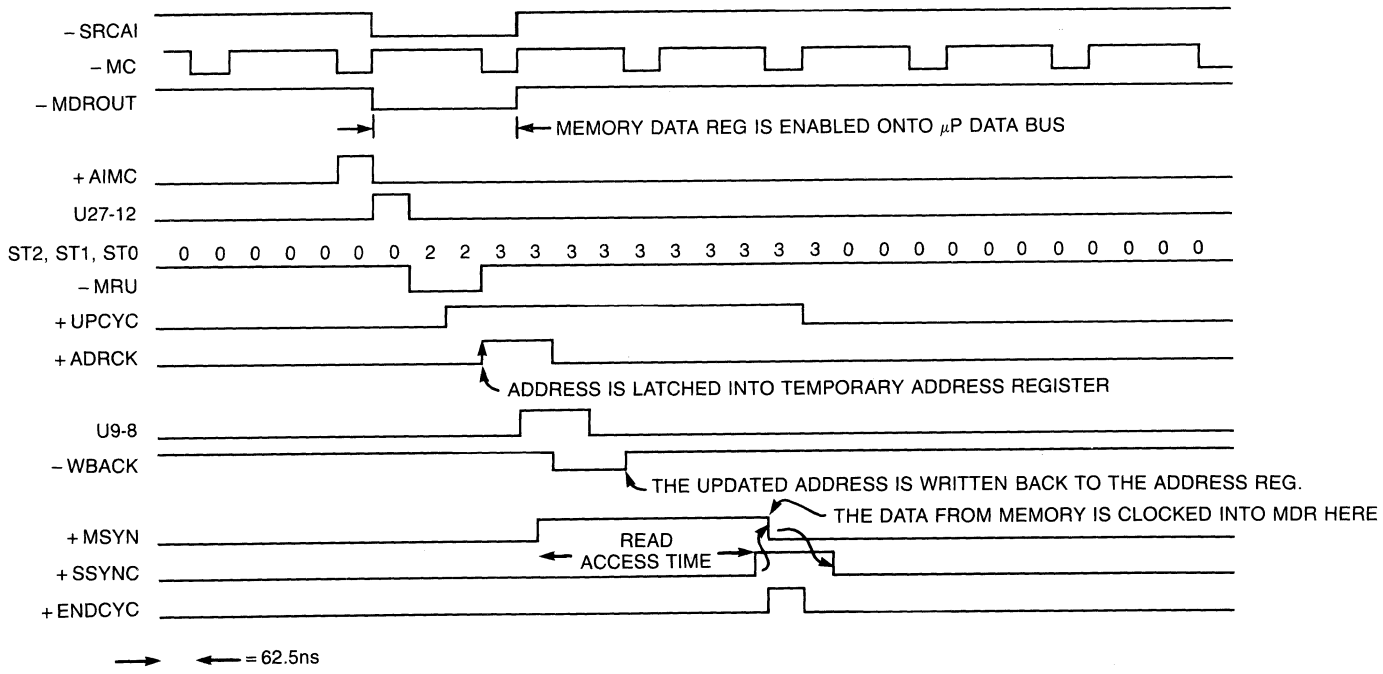
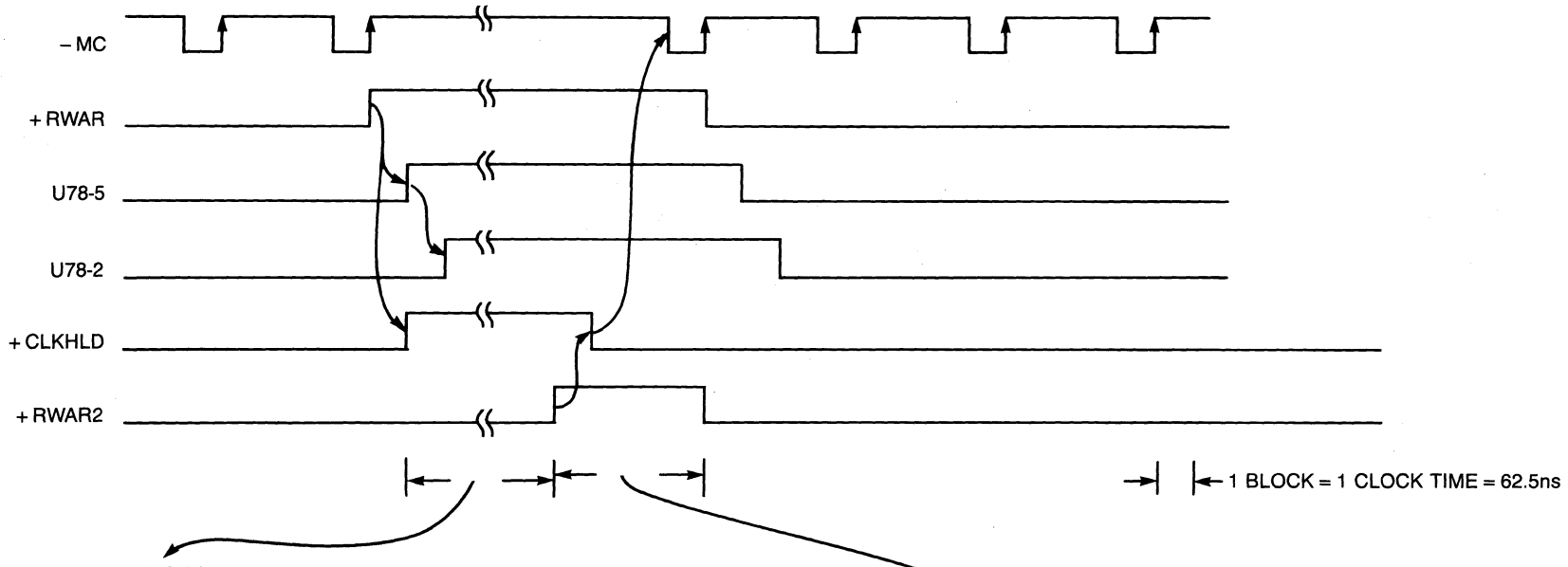


Figure 4-5. Timing Diagram, Microprocessor Read Cycle (MOV AI0, R0)



NOTE:

THIS INTERVAL WILL BE 2 CLOCK TIMES *OR* LONGER (4 CLOCKS SHOWN). DURING THIS INTERVAL WE MUST WAIT FOR EITHER OR BOTH OF THE FOLLOWING:

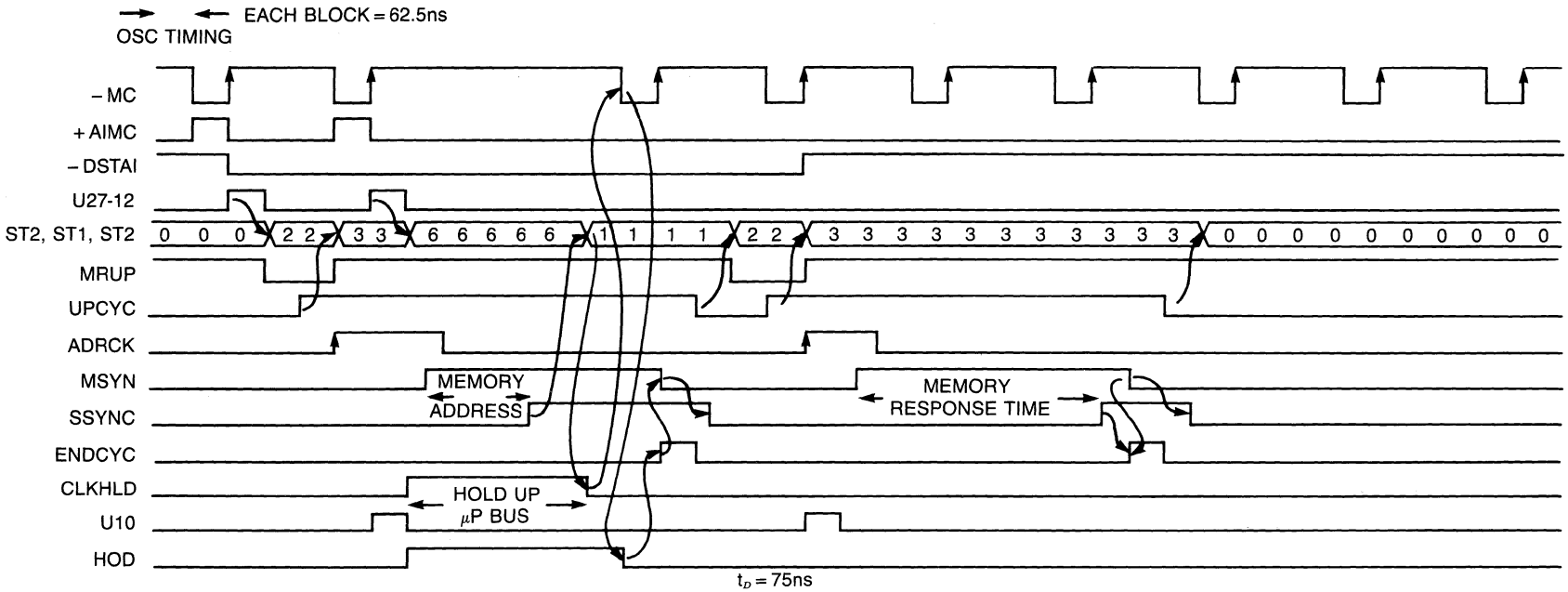
1) WAIT FOR THE ANALOG INPUT FIFO TO RELEASE THE MEMORY (THIS COULD TAKE MORE THAN ONE MEMORY CYCLE IF FIFO IS IN HOG MODE). THE REASON WE HAVE TO WAIT FOR THIS IS THAT FOR THE μP TO WRITE TO ARM OR ARL IT NEEDS TO DRIVE THE 4-BIT ADDRESS REGISTER SELECT LINES, WHICH WOULD CONFLICT WITH THE FIFO'S NEED TO DRIVE THESE LINES SO THE DATA ENDS UP IN THE PROPER BUFFER.
2) IF THE PREVIOUS μP INSTRUCTION HAS INITIATED A MEMORY ACCESS THEN WE MUST WAIT FOR THAT MEMORY CYCLE TO COMPLETE. IF WE DID NOT WAIT THEN THE FOLLOWING SEQUENCE WOULD NOT WORK:

```
MOV #6,MCMD
MOV AI3,R0
MOV RI,ARL
```

THIS IS BECAUSE THE THIRD INSTRUCTION MUST FORCE THE 4 ADDRESS REGISTER SELECT LINES TO 6 SO THAT IT WILL WRITE INTO THE CORRECT ADDRESS REGISTER BUT THIS WOULD INTERFERE WITH THE MEMORY READ INITIATED BY THE SECOND INSTRUCTION SINCE THAT READ REQUIRES THE 4 ADDRESS SELECT LINES TO REMAIN AT 3.

THE DATA TRANSFER TO OR FROM ARL OR ARM OCCURS OVER THE μP DATA BUS DURING THIS INTERVAL.

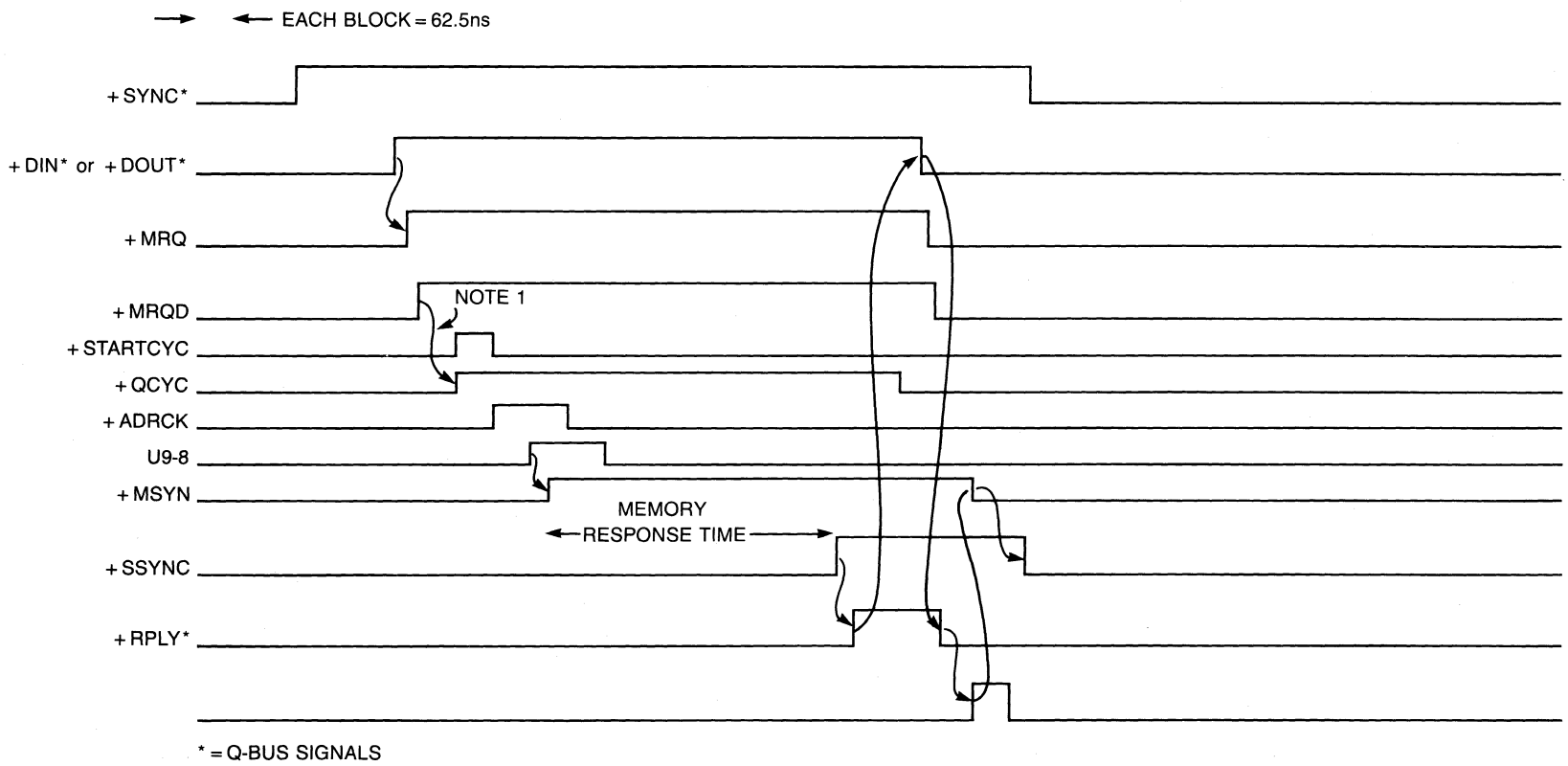
Figure 4-6. Timing Diagram, Microprocessor Timing for Reading/Writing ARM/ Address Register



THE ABOVE TIMING WOULD RESULT FROM 2 CONSECUTIVE INSTRUCTIONS IN WHICH BOTH DO A MEMORY WRITE

e.g. MOV R0,AI0
MOV R1,AI0 (ASSUMING NO INTERFERENCE FROM FIFO OR Q-BUS CYCLES)

Figure 4-7. Timing Diagram, Microprocessor Hold Timing Due to Consecutive Memory Accesses



* = Q-BUS SIGNALS

NOTE 1: THIS DELAY IS ONE CLOCK TIME MINIMUM (MINIMUM TIME SHOWN) BUT THIS MAY BE A MUCH LONGER DELAY IF THE FIFO AND/OR THE MICROPROCESSOR ARE ACTIVELY USING THE MEMORY. IN AN EXTREME CASE WHERE THE ANALOG INPUT FIFO GETS ITSELF INTO ITS "HOG" STATE FOR MORE THAN 10μs, THIS DELAY COULD BE SO LONG THAT THE LSI-11 BUS "TIMES OUT" (i.e. TRAPS TO 4).

Figure 4-8. Timing Diagram, Q-Bus Memory

operation. This means that the data is actually written during the next microcode instruction and that the data read is actually the data requested by the previous AI access. The data register MDR is used to hold the data value for microprocessor access but reading or writing this register does not cause the address in the MAR to change.

Memory Write from Analog Input FIFO Port

The analog/digital FIFO control port uses only one way transfers. The same 16 address registers handling microprocessor access are also used to supply the destination for the words coming from the FIFO. The AIs are used to store the data in much the same way as the microprocessor port except that the MDR is not used. The AI to be used is stored in the FIFO along with each data point.

The following subparagraphs contain descriptions and discussions of the major circuits contained on the memory control board.

Slave and Interrupt Interface

Three sets of Q-bus addresses in the I/O page are decoded and access to these locations by the LSI-11 causes an interrupt on the microprocessor bus. The microprocessor uses these interrupts to provide emulation of hardware devices on the Q-bus. The microprocessor responds to reads and writes of the slave interface, generates device interrupt requests and provides interrupt vector response.

4.7.1 Source/Destination Decoding

This decoding circuit receives 18 bits of source and destination data (S0-8 and D0-8) from the CPU board via the microprocessor bus.

Twelve of the bits, D3-D8 and S3-S8, are used to produce source and destination code signals -AIMC (codes 240-257) and +SD8MC (codes 260-267) and six clock and enable lines. The codes are listed numerically and described in Table 4-3. The clock and enable lines are as follows:

DSTAI	Destination is one of 16 access initiators.
DST8	Destination 8 is one of the other eight control/data registers. The high output is applied as data to an 8-bit addressable latch U115 to decode one of six possible outputs.
SRCAI	Source is one of 16 access initiators. The low output of this signal is ORed with the decoder output -SRCMDR to produce -MDROUT which clocks outputs of the memory data register feedback circuit. The high output of the signal ORed with +SRC8 produces +SOURCE to provide a DIRection control input to the microprocessor buffer circuit. The same signal is inverted to also generate -SCM0 to the CPU board master clock circuit.

SRC8	Source 8 is one of eight control/data registers. In addition to the function in the SRCAI description, this signal is used as an enable input to decoder U95.

Eight bits, S0-S3 and D0-D3, once gated out flip flop U123 are used to provide the address and control signals for the selected circuits. One function is to provide the access initiator selection logic with S0L-S3L and D0L-D3L which are multiplied to generate four bits of data MARA0-MARA3 to the address register selection logic. Three source bits, S0-S2, are the select inputs to decoder U95 which produces six source address and control signals. Three destination bits, D0-D2, are used as select inputs to 8-bit addressable latch U115 which generates six destination address and control signals. In addition, three combined outputs from both the decoder and the latch generate the address register read/write command. The address and control lines are as follows:

CLRBFI	Code 266, see table 4-3.
CLRQSI	Code 267, see table 4-3.
SRCMCMD	Source is memory command register. The four least significant bits C4-C7 are written back to be 0.
SRCMDR	Source memory data register. ORed with SRCAI to produce MDROUT. See SRCAI description.
SRCARL	Source address register least significant word. The signal selects the A or B inputs of the address register readback circuit. If low, the least significant word, ARL data, is sent to the microprocessor data bus buffer circuit and placed on the microprocessor data bus. If high, the most significant word, ARM data, is sent to the microprocessor data bus buffers.
SRCAR	Source address register. See SRCARL description above. SRCARL and SRCARM, ANDed together, produce SRCAR. Used as the output enable for the address register readback circuit.
SRCARM	Source address register most significant word. See description for SRCARL.
DSTOFF	Destination offset memory register. Used to load the 16 bits of high and mid-range offset data in U111 and U112.
DSTMAP	Destination MAPMEM register. Loads the MAPMEM register on the rising edge of the clock.

DSTMCMD	Destination is memory command register.
MDRIN	Memory data register in.
DSTARL	Destination address register least significant.
DSTARM	Destination address register most significant. This signal provides the write enable input to each of the four RAMSs in the ARM register.
RWAR	Read/write address register.
DCM0/SCM0	If any of the registers on this board is accessed the appropriate clock delay bit is asserted on the microprocessor bus for one additional clock delay.

Table 4-3. Source & Destination Codes, Memory Controller

Note: The action indicated is not begun until the last memory cycle initiated by the microprocessor has been completed (the microprocessor is put on hold if necessary). Detailed descriptions of the functions performed by these registers are provided in later sub-paragraphs.

Code	Name	MEANING AS SOURCE	MEANING AS DESTINATION
240	AI0	ACCESS INITIATORS (READ)	ACCESS INITIATORS (WRITE)
241	AI1	ACTION FOR AI0-15:	ACTION FOR AI0-15:
242	AI2	a) Read the current contents of	a) Write to the MDR.
243	AI3	the memory data register (MDR).	
244	AI4	b) Initiate a memory read at the	b) Initiate a memory write at the
245	AI5	address and in the specified	address found in the specified
246	AI6	address register (0-15).	address register (0-15).
247	AI7	c) Add Δ Adr to the address	c) Same as for source.
250	AI8	register being used. Δ Adr is	
251	AI9	determined by the INCR field	
252	AI10	of the address register.	
253	AI11	d) Cause an interrupt in the micro-	d) Same as for source.
254	AI12	processor if the address bit	
255	AI13	selected by ASEL field of the	
256	AI14	address is at the level indicated	
257	AI15	by ASELHL.	

Table 4-3. Source & Destination Code, Memory Controller (Continued)

Code	Name	MEANING AS SOURCE	MEANING AS DESTINATION
260	ARM	Read the MSW of the address register specified by the ARS field of the MCMD (memory command register).	Write the MSW of the address register specified by the ARS field of the MCMD.
261	ARL	Read the LSW of the address register specified by ARS.	Write the MSW of the address register specified by ARS.
262	MDR	Read the memory data register. All data transfers between microprocessor and memory occur through this.	Write the memory data register.
263	MCMD	Read memory command register.	Write memory command register.
264	MAPMEM	— not used —	Write memory mapping register, defines three regions (Low, Mid, High) in LSI-11 address space.
265	OFFMEM	— not used —	Write memory offset register, specifies offset addresses for the MID and HIGH regions defined by MAPMEM.
266	CLRQSI	Clear interrupt from Q-Bus slave interface.	— not used —
267	CLRBFI	Clear interrupt from buffer full or trigger event. Allow FIFO output to continue.	— not used —

4.7.2 Memory State Sequencing Circuit

This paragraph is supported primarily by sheet 2 of schematic diagram 2515-4702-2D. The purpose of the circuit is to sequence memory accesses between the 2501 microprocessor, the FIFO, and the Q-bus devices. The next device to be serviced is based on the current device being serviced and the devices requesting service.

If the FIFO is almost full, the microprocessor bus and the Q-bus are locked out and the FIFO will be serviced. If the FIFO is less than 3/4 full it is serviced in turn. If it is less than half full it is serviced only if the memory is idle (i.e., not servicing the microprocessor or Q-bus.) If all devices are constantly requesting memory, service ordering is in turns. The FIFO, if full or nearly full, is first. If not, the condition of the microprocessor or Q-bus dictates service ordering and the order (assuming the microprocessor bus condition meets the criteria) is the microprocessor first, then the FIFO, and after that the Q-bus. This order is repeated as long as all devices are constantly requesting memory.

4.7.2.1 Timing. Timing is provided by the master clock on the CPU board which provides -OSCOUT and MCOUT. These signals are buffered and supplied to the sequencing circuitry as MC and OSC.

The master clock signal, MC, cycles once for every microprocessor instruction cycle and is used to latch in source and destination registers.

The oscillator signal, +OSC, provides the 6.25 nsec per cycle clock to synchronize state changes.

4.7.2.2 Memory State and Sequence PROM. Control PROM U28 receives four bits of status data from the hold logic. These consist of an active request for memory access by either the microprocessor (MRUP) or the Q-bus (MRQD) and two FIFO status bits (FS0D and FS1D) which indicate how full the FIFO is. Two PROM outputs (MS0-MS1) indicate the present state of the memory and the device to be serviced. The third PROM output (STARTCYC) starts the memory cycle.

The condition of status signals MS0 and MS1 indicate which cycle the memory is currently in. If both MS0 and MS1 are logic 0s, the memory is idle. If MS1 is a 0 and MS0 is a 1, the microprocessor is being serviced. If MS1 is a logic 1 and MS0 is a logic 0, the FIFO is serviced. If both MS1 and MS0 are logic 1s, the Q-bus is serviced.

4.7.2.3 Microprocessor Memory Access Cycle. Assume now that the microprocessor has requested access to the main memory (Unibus memory). The request, -MRUP, goes low to the sequencing PROM (U28). If no other request is being made, the other inputs will be high. The PROM outputs memory state 01 (output MS1 is a logic 0 and MS0 is a logic 1) to the access initiator selection logic and the microprocessor data bus buffer circuit. The microprocessor cycle signal, UPCYC, becomes active for the duration of the cycle.

STARTCYC from the PROM causes the new MAR address to be computed on ADRCK. Master sync signal MSYN tells the memory to write the current data. During the memory cycle the new address is latched into MAR by -WBACK. Once the data is written the memory returns SSYN.

If the microprocessor wishes to read data from the main memory a microprocessor read signal (UPREAD) is generated in the hold logic. The Unibus data is then clocked into the memory data register and read back to the microprocessor via bus transceivers U124 and U125.

If there are no further requests for memory at this time, the sequencer PROM goes to a logic 00 (idle state).

4.7.2.4 FIFO Memory Access Cycle. The state of the analog input FIFO is determined by status bits FS1 and FS0 from the Channel Control board. The FIFO is serviced if one of three conditions prevail.

a. FIFO status bits FS1 = 0 and FS0 = 1 indicating the FIFO is less than half full. The FIFO may be serviced then only if the memory is idle.

b. FIFO status bits FS1 = 1 and FS0 = 0 indicating the FIFO is more than half full but less than three-quarters full. Then the FIFO is serviced in its turn, which is following the microprocessor.

c. FIFO status bits are both ones (11) indicating the FIFO is in excess of three-quarters full (HOG mode). The memory is required to service the FIFO exclusively and, in effect, lock out the microprocessor and Q-bus.

To accomplish the cycle FIFOCYC is sent low to the channel control board to acknowledge that a FIFO cycle is in progress. This causes the current FIFO data output to be placed on the pseudo-Unibus data lines. The data is not latched or buffered by the Memory Controller board. At the end of the memory cycle CKFIFO is sent to the Channel Control board to cause access of the next data value in the FIFO.

4.7.2.5 Q-Bus Memory Cycle. If the Q-bus requests access to the main memory the MRQD input to PROM U28 will go high. The PROM will then change the memory state outputs to logic 1s. QCYC is then generated to enable the address (A13-20) from the Q-bus memory mapper and Q-bus address latches onto the pseudo-Unibus lines. Note that the address lines on the pseudo-Unibus are inverted relative to normal usage of the Unibus. The Q-bus data to be written to the main memory is then driven onto the pseudo-Unibus. If data is to be read from the main memory onto the Q-bus, the Unibus data is driven to the Q-bus transceivers and on to the LSI-11/23. The master sync signal MSYN tells the memory that there is valid address and data on the pseudo-Unibus. Once a read or write has been completed, the Unibus memory sends SSYN low to memory state sequencing circuit (via backplane connector B1U) to advise the completion. An end of cycle (ENDCYC) signal is sent high to the sequencer PROM.

4.7.3 Microprocessor Hold Logic

This paragraph is primarily supported by sheet 3 of schematic diagram 2515-4702-2D. The microprocessor hold logic performs, controls, and reflects the states of a memory access cycle requested by the microprocessor or the analog input FIFO. If a Q-bus cycle is indicated the circuits forward the active Q-bus data request bit (MRQD) to the sequencing circuit for action.

One 256-word x 4-bit PROM, U14, and U27 latch act as a state machine. This state machine is used to provide control of intermediate operations during a memory cycle.

Since access to main memory is overlapped with execution of microprocessor instructions it is possible for a memory access request to be generated before the previous memory cycle is completed. There is also a potential timing problem if one of the microprocessor registers on this board is accessed by the microprocessor during a memory cycle. The signal, CLKHLD, is generated during these states and is sent to the CPU board to cause the processor to "hold" in the current microcode instruction until the previous memory access is completed.

There are eight states of a memory access cycle controlled by the hold logic. These states are reflected by the logic present on three state bits ST2, ST1, and ST0. The following sub-paragraphs describe the eight states.

State 0, IDLE	ST2=0, ST1=0, ST0=0. The memory is idle and no access request is pending. At this time all the inputs to the state machine are zeros and are ignored. HOLD and MRUP are zeros.
State 1, 6A	ST2=0, ST1=0, ST0=1. This is an intermediate phase during which the state machine has received an active signal indicating a microprocessor memory cycle is pending. HOLD and MRUP are zeros.
State 2, RQSTM	ST2=0, ST1=1, ST0=0. At this time the microprocessor is requesting access. -MRUP is sent low to the state sequencing circuit to initiate sequencing the cycle. During State 2, HOLD will remain a 0 but MRUP goes to 1.
State 3, MCIP	ST2=0, ST1=1, ST0=1. This state indicates that a memory cycle is in progress. During this state both HOLD and MRUP are zeros.
State 4, RQSTM, RQST2	ST2=1, ST1=0, ST0=0. This indicates that a memory request is in progress and a second request for memory access by the microprocessor is being processed. Both HOLD and MRUP go to logic 1.
State 5, RQSTM, RQSTR	ST2=1, ST1=0, ST0=1. This state is entered if a memory request is in progress and access to a register (ARL, ARM, MDR, MCMD, MAPMEM, or OFFMEM) is requested. Both HOLD and MRUP are logic 1s.
State 6, MCIP, RQST2	ST2=1, ST1=1, ST0=0. This shows that the second request for memory access made by the microprocessor is now in progress. HOLD is now a logic 1 and MRUP goes to 0.
State 7, MCIP, RQSTR	ST2=1, ST1=1, ST0=1. This indicates that a memory cycle was in progress when a register access was requested. HOLD is a logic 1 and MRUP is a logic 0.

4.7.3.1 Timing. Timing is provided by -OSCOUT and -MCOUT from the 2501 microprocessor bus via the backplane.

4.7.4 AI Selection, Microprocessor Data Bus Buffer Circuit

This paragraph will discuss sheet 4 of the schematic. The circuit performs three functions which are micro-processor data bus buffering, access initiator selection logic, and pseudo-Unibus control lines. To perform these functions the circuit uses two octal bus transceivers (U124 and U125), one quad register (U102), one quad 2-input storage multiplexer (U97), one dual 4 to 1 data selector (U42), and one quad data selector (U96).

4.7.4.1 Microprocessor Data Bus Buffering. Octal bus transceivers, U124 and U125, allow asynchronous data transmission in either direction depending on the logic level at direction control input DIR. The buffered data, C0-C15, carries the data to be written from DB0-DB15 for a destination register and carries the data to be placed on DB0-DB15 for a source register.

4.7.4.2 Access Initiator Selection Logic. The four address bits, ARA0-ARA3, determine which of the 16 MARs is to be used for the current read/write operation. For reading or writing the MAR registers the four least significant bits of U102 drive ARA0-ARA2 when -RWAR2 is low. Data selector U96 will be driven when -RWAR2 is high. The A inputs are selected if the FIFO is writing to memory and the B inputs are selected if the microprocessor is accessing memory through the access initiators. Eight bits (S0L-S3L and D0L-D3L) of latched source/destination data are sent from the source/destination decoding circuit to provide the active address to storage multiplexor U97. Dependent upon the state of the word select (SEL) input, DSTAI, U97 selects one source and one destination bit from the two 4-bit inputs. The active address is sent from U97 along four lines, MARA0-MARA3, to data selector U96. Four bits of FIFO status and address are also provided to U96 from the channel control board. If the FIFO requires service, these bits (FARA0-FARA3) carry the required address.

4.7.4.3 Pseudo-Unibus Control Lines. The LSI-11/23 processor controls the pseudo-Unibus lines through the memory controller circuits. The processor sends two bits of command, DIN (data in) and WTBT (write byte), via the Q-bus interface transceivers to data selector U42. The microprocessor sends UPWE from the hold logic which will always indicate a word access. When the memory is ready to accept an access, the memory state inputs, MS0 and MS1, select the data for output.

If WTBT is selected, control output lines C0 and C1 are both be zeros. If a word is to be written to memory C0 is a logic 1 and C1 is a logic 0. If a word is to be read, C1 is a logic 1 and C0 is ignored.

Control line C1 is also used by the Q-bus address latch circuit to control its line drivers and receivers.

4.7.5 Address Register Circuit

This paragraph discusses sheet 5 of schematic 2515-4702-2D. The circuit is comprised of five address register multiplexers (U37, U50, U63, U76 and U86) and eight 64-bit RAMs. The RAMs make up two register banks, ARM and ARL. The ARM bank consists of U35, U48, U61, and U74. The ARL bank consists of U33, U46, U59 and U72.

The address register circuit serves to decode and multiplex the selected address, data, incremented value, and commands into 32 bits and drive them on to the temporary address register and the full buffer detection circuits.

4.7.5.1 Address Register Multiplexers. The multiplexers receive the 16 buffered data bits ($\overline{C0}$-$\overline{C15}$) from the microprocessor bus and the one incremented 20-bit word new address (NEWA0-NEWA19) from the address register incrementer circuit. If the microprocessor is loading an address register, the C0-C15 bits drive the E0-E16 bits and the -DISTARM and -DSTARL signals determine whether they are loaded into ARM or ARL. If the address register is being updated by an AI operation (-WBACK) then NEWA0-NEWA19 are loaded into ARL and the least significant four bits into ARM. Multiplexer U86 supplies ARM RAM U72 with either the least significant bits of the microprocessor data word or NEWA16-NEWA19.

4.7.5.2 ARM/ARL RAMs. These 16-word by 4-bit RAMs are built using Schottky diode clamped transistors in conjunction with internal emitter coupled logic circuitry. An active low write line ($\overline{WE}$) controls the write/read operation of the memory. When the chip select (CS) and write lines are low, data on the D inputs (D0-D3) is written into the addressed memory word and preconditions the output circuitry so that true data is present at the outputs when the write cycle is complete. The outputs are always active since CS inputs are all pulled low.

The write cycle is initiated by the WE input going low during which time the four memory outputs are floating.

Four bits containing the address from MCMD (+ARA0-3) are applied to the A inputs of each RAM. Write enable is provided by either -DSTARM and -DSTARL from the source/destination decoding circuit or by -WBACK from the memory state sequencing circuit. -WBACK will load the four ARL registers as well as the four least significant bits of ARM.

The output of the RAMs is 32 bits, +G0 - +G31 containing the current address, increment value, and control bits. These bits are bused from the address register circuit to the temporary address register and to the address register incrementer circuit.

ARL Address Register Least Significant

Bit	15	14	13	12	11	10	9	8
	A15	A14	A13	A12	A11	A10	A9	A8
Output	G15	G14	G13	G12	G11	G10	G9	G8

	7	6	5	4	3	2	1	0
	A7	A6	A5	A4	A3	A2	A1	A0
	G7	G6	G5	G4	G3	G2	G1	G0

ARM Address Register Most Significant

Bit	15	14	13	12	11	10	9	8
	ASEL3	ASEL2	ASEL1	ASEL0	ASELHILO	INCR4	INCR3	INCR2
Output	G31	G30	G29	G28	G27	G26	G25	G24

	7	6	5	4	3	2	1	0
	INCR1	INCR0	INTENB	SPARE	A19	A18	A17	A16
	G23	G22	G21	G20	G19	G18	G17	G16

Descriptions

Bit 0-19 — A0-A19. 20-bit word address.

Bit 20 — A20. Not used; spare.

Bit 21 — INTENB. If bit equals 0, buffers full interrupts are enabled for this buffer.

Bit 22-26 — INCR0-INCR4. Specifies the auto-increment value to be added into the 20-bit address after every memory cycle:

$0 \leq INCR < 16$ ΔADR = INCR (0 to 16)
$16 \leq INCR < 26$ ΔADR = 2**(INCR-12) (16 to 16384)
$27 \leq INCR < 31$ ΔADR = -2**(INCR-27) (-1 to -16)

Bit 27 — ASELHL. When selected end of buffer address bit matches ASELHL, buffer full interrupt occurs if INTENB is 0.

Bit 28-G31 — ASEL0-ASEL3. Selects one of address bits A1-A16 to indicate the end of a memory buffer. ASEL3 is used as an output enable strobe.

4.7.6 Temporary Address Register Circuit

This circuit is divided into two sections: the address register readback circuitry and the temporary address registers on sheet 6.

The function of this circuit is to perform two operations:

a. Allow microprocessor readback of either ARM or ARL registers.

b. Provide temporary storage of the address from the selected MAR, demultiplex the information onto two separate buses, one of which interfaces with the pseudo-Unibus and one which buses the data to the address register incrementer circuitry so as to increment the 20-bit word address after each memory cycle.

The circuitry consists of four quad data selector/multiplexers (U34, U47, U60 and U73) with non-inverting three state outputs, five quad D registers (U75, U36, U49, U62 and U85).

4.7.6.1 Readback Register. The two 16-bit words (G0-G31) from the ARL and ARM registers are sent via the bus to the readback register. The readback register consists of four quad data selectors: U72, U60, U47, and U34. If ARM is being read back (-SRCARM) bits G16-G31 are selected to drive C0-C5. If ARL is to be read (-SRCARL) bits G0-G15 are selected.

4.7.6.2 Temporary Address Register. The temporary address register consists of five quad D registers capable of generating two sets of outputs, standard (Q) outputs and three state (Y) outputs, onto separate data buses. The 20-bit address word (G0-G19) is applied in 4-bit bytes to the D inputs of the five registers.

This register is loaded from the currently selected MAR when ADRCK goes high. The Y outputs, A1-A20, are sent via the pseudo Unibus address lines on to the main memory. The Y outputs are disabled if a Q-bus access to memory is performed. The Q outputs, K0-K19, are sent on the other bus to provide the 20-bit word to the address register auto-incrementer.

4.7.7 Address Register Incrementer

This circuit is divided into two sections. One section contains the address register auto-incrementer which consists of five 4-bit binary full adders U38, U51, U64, U77 and U87 on sheet 7. The auto-incrementer PROMs U52 and U55 (on sheet 6), supply the increment values. The other section contains the full buffer detection circuitry consisting of two data selectors, U25 and U89, with true and inverted three-state outputs, a hex inverter U11, and gates U12, U29 and U106 on sheet 7.

The function of the circuit is to compute the incremented address of the MAR with a value specified in the ASEL0-ASEL3 bits in the MAR. If the new address has the bit specified by the INCR0-INCR4 bits of MAR is equal to the ASELHI bit (of the MAR), the full buffer condition exists. An interrupt is enabled if INTENB is 0 in the MAR.

4.7.7.1 Address Register Auto-Incrementer. +INC0-INC15 from the auto-increment PROMs and +K0-K19 from the temporary address register are sent from the temporary address registers circuitry to provide the auto-incrementer with the auto-increment value and the current address. U38, U51, U64 and U77 comprise a 20-bit adder.

The incremented output, 20-bit word +NEWA0-19, is stored back in the MAR. Sixteen bits, +NEWA1 -+NEWA16, are sent to the full buffer detection circuit for monitoring of the FIFO condition.

PROMs U52 and U65 are 32W x 8B standard TTL programmable read-only memories with three state outputs. Five MSBs, +G22 - +G26, sent to the PROM inputs via the data bus, contain INCR0-4 from MAR specifying the auto-increment value to be added into the 20-bit address after each memory cycle. Each PROM generates 8 lines (total 16 lines) consisting of +INC0 -+INC15 auto-increment value to be used by the auto-incrementer.

4.7.7.2 Full Buffer Detection. Four bits in ARM (ASEL0-ASEL3) allow specification of which bit of the low 16 address bits will be used to generate a microprocessor interrupt.

Sixteen bits of the incremented data address NEWA1-16 are applied to the two data selectors (U25 and U89). Three bits, +G28-30 (ASEL0-2) in ARM, are sent to the select inputs. +G31 (ASEL3) determines which of the selectors to enable.

The state of the select inputs will determine which bit will be chosen, A1-A16, to indicate the end of a memory buffer. If the selected bit matches the condition of +G27 (ASELHL), and G21 (-INTENB) is 0 the FULBUF signal goes to the microprocessor hold logic to initiate the interrupt.

4.7.8 Q-Bus Slave and Interrupt Control Circuit

This circuit provides control and status bits in the MCMD register which allow the microprocessor to interrupt the LSI-11/23 and act as a slave for the Q-bus. The circuit consists of line driver U20, quad D register U24, flip-flop U22 and hex inverter U67. Gates U21, U23 and U55 provide load control of the MDR.

4.7.8.1 Memory Command and Control. Sixteen bits of memory command, C0-C15, are sent on the microprocessor data bus to control read and write functions. The command register is shown below and is followed by bit descriptions.

15	14	13	12	11	10	9	8
FS1	FS0	MS1	MS0	Not Used	IAKI	INOUT	INTACK

7	6	5	4	3	2	1	0
Not Used	RPLYQ	TRIGENB	ENBF	ARA3	ARA2	ARA1	ARA0

Bit

0-3 ARA0-ARA3. These bits select which address register to access when using ARL, ARM. See paragraph 4.7.5.

4 ENBF enables the analog input FIFO to write to the main memory. This constitutes the only control the microprocessor can exert over the memory access sequence.

5 TRIGENB enables trigger interrupts for all buffers.

6 RPLYQ is a reply to the Q-bus interface which is used to complete the handshake to the LSI-11/23 for slave registers and interrupt vectors.

7 Not used.

8 INTACK is the interrupt acknowledge from the LSI-11/23. (Read only.) ANDed with IRQ.

9 -INOUT. This indicates Q-bus DATA IN or DATA OUT is active. (Read only.)

10 IAKI is interrupt acknowledge from the LSI-11/23.

11 Not used.

12, 13 MS0, MS1. (Read only.) These two bits reflect the current memory state to be:

00 Memory idle.
01 Microprocessor cycle in progress.
10 FIFO cycle in progress.
11 Q-bus cycle in progress

14, 15 FS0, FS1. (Read only.) These two bits reflect the current FIFO status to be:

00 FIFO empty.
01 FIFO less than 1/2 full but not empty.
10 FIFO less than 3/4 full but more than 1/2 full.
11 FIFO is more than 3/4 full.

The signal RPLY is sent to the Q-bus to indicate that the data has been written or is now valid for reading. It is generated by either the microprocessor slave response (RPLYQ) or by the slave sync signal returning from the main memory (SSYNC).

An interrupt request to the LSI-11/23 generated by this board sets U22. When the acknowledge signal (IAKI) returns and U22 is set INTACK generates a microprocessor interrupt. If this board has not requested the Q-bus interrupt the acknowledge is passed out to the next Q-bus device (IAKO).

The signal which loads data into the MDR (CKMDR) is generated when one of the following events occurs:

1) Register load of AI or MDR (MC and MDRIN);
2) Q-bus slave write (DOUT and REPLYQ); or
3) Register load of AI register (MSYN and UPREAD).

In order for the microprocessor to emulate devices on the Q-bus, access to the decoded addresses by the LSI-11/23 generates interrupts to the microprocessor. The microprocessor must complete the slave handshake using the RPLYQ bit in MCMD.

4.7.8.2 Interrupt Routines. The following routines show the general signal flow used for interrupt routines used in device emulation.

a. DATO OR DATOB Interrupt Routine

1) Lower +ENBF
2) Wait for memory idle (MS0=MS1=0)
3) Save MDR
4) Raise +RPLYQ
5) Wait for -INOUT to go high (inactive)
6) Lower +RPLYQ
7) Receive data from MDR
8) Restore MDR, raise +ENBF
9) Return from interrupt

b. DATI Interrupt Routine (word to LSI-11/23)

1) Lower +ENBF
2) Wait for memory idle (MS0=MS1=0)
3) Save MDR
4) Load data to send into MDR
5) Raise +RPLYQ
6) Wait for -INOUT to go high
7) Lower +RPLYQ
8) Restore MDR, raise +ENBF
9) Return from interrupt

c. Procedure to cause Interrupt in LSI-11/23

1) Raise +IRQ
2) Wait for +INTACK to go high (INTACK causes an interrupt of the microprocessor.)
3) Lower +IRQ and +ENBF
4) Wait until memory is idle (MS0=MS1=0)
5) Load desired interrupt vector into MDR (save if necessary)
6) Raise +RPLYQ (send vector)
7) Wait for +IAKI to go low (receive vector)
8) Lower +RPLYQ and raise +ENBF (restore MDR if necessary)

4.7.9 Memory Data Register

The memory data register circuit is comprised of the memory, its data register, and readback three-state drivers. The primary function is to store data and commands going to and from the pseudo-Unibus data lines.

Four quad 2-input multiplexers with storage (U19, U32, U45, and U58) are used as the memory data register (MDR). Two line drivers, U31 and U44, comprise the readback register. Line drivers U18 and U57 drive the data to the pseudo-Unibus lines.

4.7.9.1 I/O Signal Flow. The MDR is loaded via the microprocessor bus (C0-C15) inputs when either the MDR register or one of the AIs is the destination. MDRIN selects the A2-D2 inputs. CKMDR loads MDR when MDRIN and MC go high.

The MDR is loaded from the pseudo-Unibus inputs (UD0-UD14) when MDRIN is low. CKMDR loads MDR when the LSI-11/23 writes to the slave interface (DOUT) and the microcode completes the slave handshake (RPLYQ).

Readback of the MDR to the microprocessor bus at U31 and U44 occurs (-MDROUT) when either the MDR or one of the AIs is the source.

The pseudo-Unibus data lines are driven by U18 and U57 when the microprocessor is accessing the main memory (UPWRITE) or when the LSI-11/23 is reading from the slave interface (DIN) and the microcode completes the slave handshake (RPLYQ).

4.7.10 Q-Bus Memory Mapper

The Q-bus memory mapper circuits contain a memory map comparator, an offset adder, OFFMEM register and MAPMEM register.

MAPMEM and OFFMEM are microprocessor registers used to define a single mapping of the LSI-11/23 address space into the address space of the Unibus memory. Once the mapping registers are set up, the LSI-11/23 processor and any Q-bus device may do reads and writes from the Unibus memory in the same manner as for "real" Q-bus memory. MAPMEM defines three memory regions. The size of each region is fixed at 4K words and each region starts on a 4K word address boundary. The lowest region maps directly but a 4K address offset specified in OFFMEM is added to the mid and high regions.

4.7.10.1 MAPMEM Regster. This register uses two octal D-type flip-flops, U98 and U99, to decode the 16-bit command word, C0-C15, from the command register. The MAPMEM register is illustrated below.

15	14	13	12	11	10	9	8
ENB HI	ENB MID	ENB LO	Not Used	IRQ	Not Used	PB4	PB3
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PB2	PB1	PB0	PA4	PA3	PA2	PA1	PA0

BITS

0-4 PA0-PA4. Starting block number of middle region. (One block = 4K words.)

5-9 PB0-PB4. Starting block number of high region. (One block = 4K words.)

10 Not used.

11. IRQ. Used to generate an interrupt on the Q-bus.

12. Not used.

13 ENBLO. Enables low region.

14 ENBMID. Enables middle region.

15 ENBHI. Enables high region.

If an interrupt request signal is active in the MAPMEM register the IRQ signal is sent active to the Q-bus slave and interrupt control circuits and to the Q-bus transceivers.

The LSI-11/23 sends -BS7L to the circuit to initiate a memory access cycle to the I/O page. INOUT is also sent from the Q-bus address decode circuit to provide an active input if +MRQ is to be generated. These two signals, the active enable in the command word and the matching enable from the comparator (PROM U127) circuit, generate the Q-bus memory request (MRQ) to the state machine in the microprocessor hold logic. The address bits are cleared out of the MAPMEM register by -INIT from the Q-bus address decode circuits.

Note: The addresses that appear on the A0-A20 pseudo-Unibus lines are inverted relative to normal Unibus operation.

4.7.10.2 OFFMEM Register. The OFFMEM register is comprised of two octal D transparent latches (U111 and U112) with edge-triggered flip-flops. The sixteen bit command word, C0-C15, is applied to the inputs of the latches in two 8-bit bytes, eight MSBs received by U111 and eight LSBs received by U112. +DSTOFF (from the source/destination decoding circuit) is sent to clock the inputs to the outputs with a 15 nanosecond pulse. If the active address is found to be in the MID region, U112 output is enabled by PROM U127 of the memory map comparator. If the active address is in the HIGH region, the PROM enables the output of U111. When enabled, the output is applied to the OFFSET ADDER inputs in two 4-bit bytes along with the five data bits (AJ13 - AJ17) from the Q-bus address latches.

15	14	13	12	11	10	9	8
HOFF7	HOFF6	HOFF5	HOFF4	HOFF3	HOFF2	HOFF1	HOFF0

7	6	5	4	3	2	1	0
MOFF7	MOFF6	MOFF5	MOFF4	MOFF3	MOFF2	MOFF1	MOFF0

Bits 8-15 HOFF0-HOFF7. Indicate the HIGH region 4K word block offset.

Bits 0-7 MOFF0-MOFF7. Indicate the MID region 4K word block offset.

Example of use of memory map registers:

Suppose MAPMEM = 160304
(ENBLO, ENBMID, ENBHI = 1) (<A0, A4> = 4) (R0, R4) = 6

Suppose OFFMEM = 6 + 254 x 256
(MOFF = 6, HOFF = 254)

Since B = 6, the HIGH region is defined to be blocks 6 through 30 of the LSI-11/23 processor address space. Since A = 4, the MID region is defined to be blocks 4 through 5. The LOW region is defined to be blocks 0 through 3.

The LOW region is mapped directly with no address change; i.e., block 0 of the LSI-11/23 address space goes directly to block 0 of the memory address space, block 1 goes to block 1, and on through block 3.

The MID region is mapped with an offset of 6 blocks (MID OFF equals 6), thus LSI-11/23 block 4 goes to 2501 block 10, LSI-11/23 block 5 goes to block 11.

The HIGH region's offset is 254 (HIGH OFF) which when added is modified to 256, therefore, LSI-11/23 block 6 goes to 2501 block 4, block 7 goes to block 5, and so on.

4.7.10.3 Memory Map Comparator. Two 4-bit magnitude comparators (U100 and U113) and one 256-bit PROM (U127) comprise the memory map comparator. The two most significant bits of the block address, PA4 and PB4, are applied directly to the PROM inputs. The other eight bits are applied in 4-bit bytes to the two comparators. The LSI-11/23 data bits AJ13 through AJ16 are also applied to the two comparators and are compared with the PB and PA bits. If the A inputs are greater than the B inputs the output is sent out on one line to the PROM. The PROM receives the two bits of compared data and the MSB of the latched address from the LSI-11/23 (via the Q-bus transceivers) and the two MSBs (PA4 and PB4) from the MAPMEM register. If the selected address is in HIGH region, the PROM sends the output enable -HIGH to the OFFMEM register. If the selected address is in the MID region, the PROM sends the output enable -MID to U112 of the OFFMEM register. In an addition, the appropriate HIGH or MID signal will be sent to the MAPMEM register memory request decode. If however, the selected address is in the LOW region the OFFMEM register is not affected and the LOW signal is sent active to the memory request decode circuit.

4.7.10.4 OFFSET ADDER. Two 4-bit binary full adders, U114 and U101, sum five Q-bus address bits (AJ13-AJ17) with an 8-bit offset value (OFF0-OFF7). If either the MID or HIGH region is being accessed, one of the two OFFMEMory registers will drive the offset bits. If the LOW region is being accessed, neither output is enabled and the OFF0-OFF7 bits are pulled high. The +LOW signal adds a "carry" to the maximum offset which is effectively no offset. The summed output is applied to line driver U88 whose output is enabled during a Q-bus memory access (-QCYC).

4.7.10.5 MRQ Generation. The Q-bus memory access request is generated by the conditions present at the request decode gates U128. If one of the three regions is detected and that region is enabled, then as long as the address is not in the I/O page (-BS7L) a memory request is generated by U23 while the address is valid (-INOUT).

4.7.11 Q-Bus Transceivers and Address Latches

The main function of the circuit on sheet 11 is to buffer the Q-bus data and control lines and to latch the Q-bus address data.

Note: Addresses that appear on A0-A20 pseudo-Unibus lines are inverted relative to normal Unibus operation.

4.7.11.1 Control Inputs. Control inputs from the Q-bus are buffered by U107, U92, and U120. A brief description of each is given below.

BIAKI	Interrupt acknowledge in
BWTBT	Write byte operation in progress
BINIT	Initialize
BSYNC	Bus cycle is starting, valid address on bus
BBS7	Current access is to the I/O page
BDIN	Data input operation in progress
BDOUT	Data output operation in progress

4.7.11.2 Control Outputs. Control outputs to the Q-bus are buffered by U107 and are listed below:

BIRQ4	Interrupt request (microcode controlled)
BRPLY	Reply to Q-bus cycle. Output data has been read. Input data is valid on bus (microcode controlled for slave/interrupt interface)
BIAKO	Interrupt acknowledge out

4.7.11.3 Data/Address Line Buffers. Transceivers U4, U17, U56, and U69 drive the data/address lines from the Q-bus in the direction desired for the current operation. When data is written to memory -DIN direction control signal is high and the BDAL bus signals drive the DAL0-DAL15 outputs. For a read operation the pseudo-Unibus data lines UD0-UD15 drive the BDAL bus lines.

4.7.11.4 Q-Bus Address Latches. The Q-bus multiplexes the same bus lines (BDAL0-BDAL15) for both data and address. Address lines BDAL16 and BDAL17 are separate. The address is valid on these lines while BSYNC is asserted and is latched into U6, U71, U84 and U94 by a buffered BSYNC signal. The lower 13 bits of address drive the pseudo-Unibus address lines (A0-A12) directly while the upper 5 bits (AJ13-AJ17) are supplied to the memory mapping circuit.

4.7.11.5 Memory Data Bus Drivers. For a memory write operation (C1) the buffered data lines DAL0-DAL15 drive the pseudo-Unibus data bus lines UD0-UD15. This occurs as long as the FIFO is not writing (-FIFOCYC) and the microprocessor is not writing nor is the slave interface being written (-DROE).

4.7.12 Q-Bus Address Decode

The memory controller provides three 8-bit equal-to comparators U80, U81, and U68 which function to decode three sets of Q-bus addresses and which are, in effect, selected slave interface devices. The three sets of Q-bus addresses possible in the memory controller are decoded therefore by spare interface decode U80 (SELSPARE), terminal interface decode U81 (SELTERM), and main communication interface decode U68 (SELI1). These interfaces decode 8 bits of address which are latched and held for the duration of the bus cycle or until -BSYNC is negated.

Ten Q-bus address selection switches are provided by switch block SW1. Seven of the selection switches VS4-VS10 are fed to the spare interface decode. Two selection switches, VS2 and VS3, are fed to the terminal interface decode. The remaining one, VS1, goes to the main communication interface decode. The states of VS1, VS2, and VS3 determine the address in the LSI-11/23 processor I/O page.

The comparators simply compare the 8-bit words for an equal-to condition. A logic low enable, provided by -BS7L and bits +AJ11 and +AJ12 NANDed together, produces an active low on the output if an equal-to condition is met. -BS7L is a signal provided by the Q-bus which indicates access to the I/O page of memory; i.e., the most significant three bits are set. In this manner U80 produces -SELSPARE, U81 produces -SELTERM, and U68 produces -SELI1.

SELSPARE decodes a group of eight words with an address 774000-777777 in the LSI-11/23 I/O page. The exact address depends on the status of address jumpers VS4 through VS10.

SELTERM decodes four words in the I/O page as follows:

01 — VS2=0, VS3=1; the normal terminal located at address 777560-777567 is indicated.

00 — VS2=0, VS3=0; the first alternate terminal located at address 776570-776567 is indicated.

10 — VS2=1, VS3=0; the second alternate terminal located at address 776570-776577 is indicated.

SELI1 decodes eight words in the I/O page if the following conditions are met:

0 VS1=0; the normal interface located at address 775000-775017 is indicated.

1 VS1=1; the alternate interface located at address 775020-775027 is indicated.

The three decode signals are gated to generate +SEL1 and +SEL0. These are then gated with the -INOUT signal to produce +SELS. These signals are now made available for use by the microprocessor interrupt logic.

4.7.12.1 System Initialize. Although not part of address decode circuit, initialization circuitry is included in this section. On power-up or BOOT conditions, or when a RESET instruction is executed by the LSI-11, the Q-bus signal -BINIT is generated. This signal is asserted for about 10 microseconds and appears on sheet 12 in its buffered form as INITQ. The differentiator circuit (U91, U106) provides a 1.0 microsecond initialize pulse INIT which is used on the microprocessor bus to reset bus hardware. This signal also causes the microprocessor to start execution at address 0.

The Q-bus signal BDCOK is also controlled by this circuitry. On power up this signal, indicating that DC power is up to full level, is held low until enough time has passed for all supply voltages to come up. This prevents glitches which occur due to a slowly rising power supply. As soon as this signal is asserted the LSI-11 begins execution of its Power Up Mode (which is normally to execute a BOOT). This same signal is asserted by the BOOT switch on the 2515 front panel. The signal from the switch is "de-glitched" by one-shot U79.

4.7.13 Microprocessor Interrupt Logic

The circuitry on sheet 13 provides generation of interrupts of the microprocessor resulting from the following events.

1. One of the Q-bus slave interfaces is being accessed.
2. Interrupt acknowledge has been returned by the microprocessor.
3. A trigger event has occurred in unloading the analog input FIFO.
4. The buffer full condition exists in unloading the FIFO.

These interrupts exist at two different priorities. Level 7 interrupt requests will be acknowledged before level 6 interrupts if they occur at the same time.

4.7.13.1 Q-Bus Slave Interrupts. If one of the Q-bus slave interfaces is accessed (+SELS) the Level 7 interrupt flip-flop (U53) is cleared and the request for interrupt is latched at U118 where it is synchronized with the master clock (-MC). This request is driven on the bus as -IR7.

When the next microcode instruction begins and priority allows the microprocessor returns the interrupt grant signal for level 7 (IG7I). If the memory controller was the device requesting service this condition is latched in U118 (-QVEC). Otherwise the grant signal is passed to the next device on the bus via signal +IG70. The -QVEC signal will place the apropriate vector on the interrupt vector lines IV0-IV7 of the microprocessor bus and the microprocessor jumps through this vector. The interrupt is cleared by access to the CLRQSI register. The same sequence of events occurs if the Q-bus acknowledges an interrupt that was requested by the microprocessor (+INTACK).

The vector address is determined by the status inputs to ROM U122.

a) +INTACK used to acknowledge the interrupt by the Q-bus slave and interrupt control circuit;

b) SEL0 and SEL1 whose logic states determine which interface (spare, terminal, or main) is to be selected;

c) AJ1-AJ3 selects the address within the interface;

d) L0 and L1 whose logic states indicate what type of operation the Q-bus is executing.

L1	L0	Function	Indicated Routine
0	0	Word to slave	DATO
0	1	Low byte to slave	DATOB
1	0	High byte to slave	DATOB
1	1	Word to master	DATI

The following are the vector addresses for the level 7 interrupts 240-277.

240-277 Q-Bus access to spare interface (selectable to be any group of 8 words between 774000-777777.
<IV4-IV0> = <A3, A2, A1, L1, L0> (see note 1)

340-357 Q-Bus access to console terminal (Normal = 777560-7, Alt #1 = 776560-7)
<IV3-IV0> = <A2, A1, L1, L0> (see note 1)

360-377 Q-Bus access to console terminal at alternate #2 (776570-7)
<IV3-IV0> = <A2, A1, L1, L0> (see note 1)

300-317 Q-Bus access to Interface I (Normal = 775000-17, alternate - 775020-37)
<IV4-IV0> = <A3, A2, A1, L1, L0> (see note 1)

227 Interrupt Grant has been returned by LSI-11.

Note 1: A1, A2, A3 - Address bits in Q-bus address.

4.7.13.2 Trigger and Buffer Full Interrupt. The buffer full condition (-FULBUF) is generated by the Buffer Full Detection circuit on sheet 7. The trigger event (-TRIGF) is signaled by a bit stored in the Analog Input FIFO (TRIG) along with the data and AI number. When this bit is set during a FIFO write (FIFOCYC) then a trigger event occurred when this data value was being placed into the FIFO. Note that there is a three point delay between the time the trigger event occurs and the time that this bit is actually loaded into the FIFO.

The sequence of events for the level 6 interrupts is nearly identical. The interrupt event (TRIGF or FULBUF) is stored in U30 and latched synchronously in U118. It is transmitted as IR6 interrupt request on the bus. The acknowledge (IG6I) causes U83 to place a vector on the bus if this board was requesting an interrupt. TRIGF and the current memory address register bits ARA0-ARA3 determine the actual vector. Access to the microprocessor register CLRBFI clears the interrupt condition. The level 6 interrupt vectors are as follows:

140-157 Trigger event interrupt
<IV0-IV3> = Access initiator with trigger event
(could be simultaneous with buffer full)

150-167 Buffer full interrupt
<IV0-IV3> = Access initiator with buffer full condition

4.8 UNIBUS MEMORY BOARD

The main memory used in the 2515 system is Model MD1700 Series Display System manufactured by Cambex Corporation. It is a 512 kilobyte MOS memory on one 6-wide board compatible with the Digital Equipment Corporation PDP-11 systems.

In the 2515 system, the memory occupies slot 5 of the card cage. It communicates directly only with the memory controller board. All accesses to the memory are controlled and sequenced by the memory controller.

Details regarding the board components and general operation can be found in the vendor manual listed in Appendix A.

4.9 CPU BOARD

The following text is supported by the diagrams listed below:

CPU Board Functional Block Diagram	Figure 4-9
CPU Board Schematic	2501-4700-2D

The CPU board, located in slot 6 of the card cage, is a 6-wide board. This board comprises most of the 2501 microprocessor, a special purpose, high speed computer constructed from approximately 140 integrated circuits. The microprocessor has 32-bit wide instruction paths and 16-bit wide data paths. The instructions are stored in PROM. The microprocessor is driven by a 16 MHz clock. A typical instruction takes 300 nanoseconds to execute.

The microprocessor is physically located on two printed circuit boards, the CPU board and the PROM board. Everything on the CPU board is considered part of the microprocessor, while only a part of the PROM board is considered part of the microprocessor. The PROM microinstruction bus interfaces with the CPU board.

Additional devices can be interfaced to the microprocessor by the use of the general purpose, high speed microprocessor bus. Included as a part of this bus is a 16-bit data bus used for transferring data within the microprocessor and to external devices. A 9-bit source field and a 9-bit destination field signal various units on the data bus to either supply data to the bus or to receive data from the bus. Data transfers and most bus activity are synchronous with the master clock, a clock pulse which comes at the end of each microprocessor instruction executed. Bus devices can interrupt the microprocessor via seven interrupt request lines and seven interrupt grant lines. The bus device transmits the address where the interrupt is to occur by means of the 8-bit interrupt vector bus.

The microprocessor has a dedicated hardware stack, 16 levels deep, used to implement subroutine calls, returns, and interrupts. The current instruction counter, status and priority level values are stored on this stack during the execution of an interrupt or subroutine. The values are then restored from the stack at the completion of the subroutine or interrupt servicing routine.

Conditional jumps, subroutine calls and returns can be executed by the microprocessor. The condition tested is either an internal condition (status register) or a condition signaled by an external device connected to the microprocessor bus. The device whose condition is being tested is selected by means of a code on the 9-bit source bus, which also serves as a condition select bus.

The basic arithmetic capability of the microprocessor is contained within a 4-chip, LSI arithmetic unit (Am 2901). Principal functions and characteristics of the Am 2901 are summarized in Appendix C. A 256-word memory, addressable in several modes, is used to provide a fast storage supplement to that included in the Am 2901 unit.

An 8-bit map register implements a multiway jump capability and provides a method of swapping data bytes on the 16-bit data bus. The map register also has special hardware which aids in the normalization operation of floating point arithmetic.

The results of the Am 2901 unit's operations set flags in the 8-bit status register. The status register bit can then be fed back into the Am 2901 by certain shift operations or tested by conditional jump instructions.

4.9.1 Bus Line Electrical Characteristics

There are four types of lines used: 1) tri-state, 2) open collector, 3) line driver, and 4) totem pole, or regular TTL output.

Tri-state lines are used for the transmission of data. The tri-state drive will enable the line at some time following the clock and the receiving component will take the data from the bus at the next clock edge.

Open collector lines are used where an ORing of several different drives is desired, as in the interrupt request lines, or where the slow rise time of open collector lines can be tolerated, as in the master clock delay control lines. The open collector lines have a 390 ohm pullup resistor and are driven by U39 or by U11, U13, U15 and U64 with the inputs wired to the enable.

Line drivers are used for only two lines, the master clock (-MCOUT) and the 16 MHz oscillator (OSCOUT). These lines are driven by U99 and U102, 50-ohm line drivers located on the microprocessor board. Master clock and OSCOUT are line drivers to minimize clock skew and to prevent ringing.

Totem pole, or regular TTL output is used for lines where there is no common bussing and moderate speed requirements. This includes the interrupt grants, which are chained, and a few other lines.

4.9.2 Microprocessor Basic Instruction Formats

The five basic types of instruction formats are:

Jump (Type 1)

JSR Jump to Subroutine (Type 2)

Conditional return (Type 3)

Immediate data (Type 4)

General Operation (Type 5)

Jump, JSR and conditional return (Types 1, 2 and 3) are change of control type instructions which are conditional based on the code specified in the 9-bit source/condition field. If the condition test is not passed, the next sequential instruction is executed. If the condition test is passed, jump (Type 1) and JSR (Type 2) instructions result in a jump to the address specified in the 16-bit jump target field. For conditional return (Type 3), a return is executed if the condition test is passed. The return is made to the last address stored in the subroutine stack by either a JSR (Type 2) instruction or an interrupt.

The immediate data (Type 4) instruction sources data from the instruction itself to the data bus. The data is accepted by bus destination specified in the 9-bit destination field.

The general operation (Type 5) instruction is where both a source and destination on the data bus are specified by the respective 9-bit source and destination fields in the instruction. The source device selected puts data on the bus and the destination device accepts the data. Variations of the code in the source and destination fields generally cause the more complicated arithmetic functions to take place.

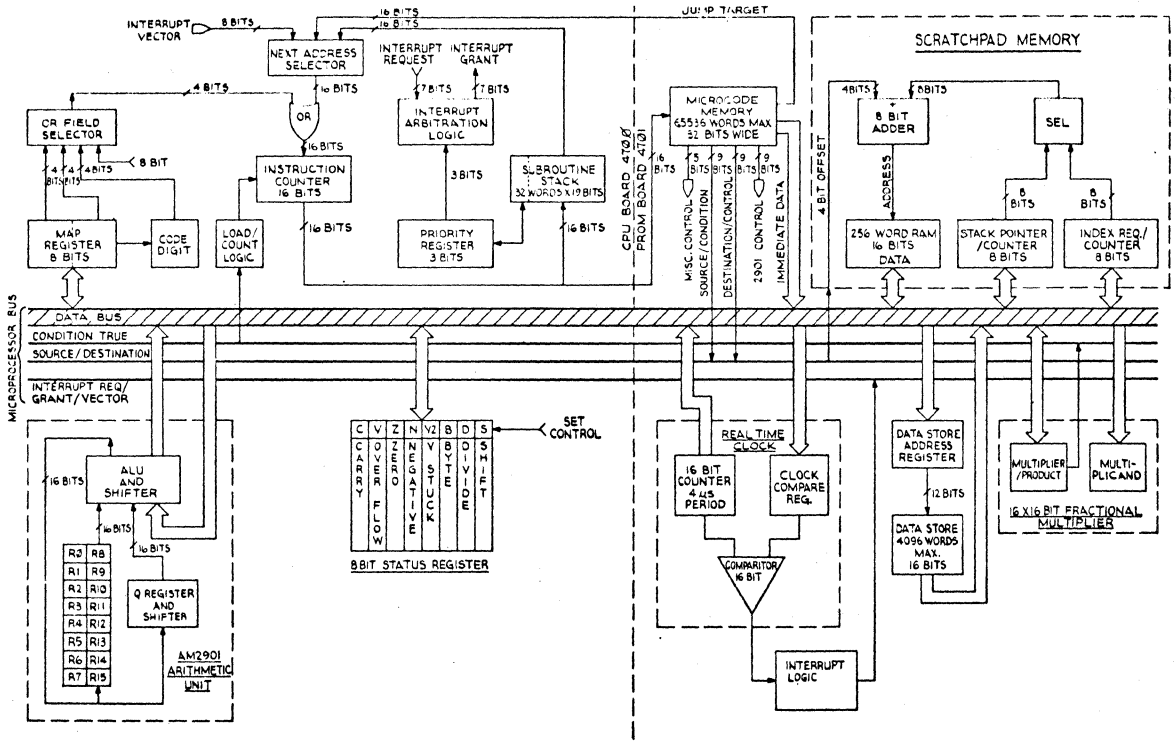


Figure 4-9. Processor Block Diagram (CPU and PROM)

Table 4-4. Basic Instruction Formats

```
30  26  22  18  14  10   6   2
J00EXCCCCCCCCCC00TTTTTTTTTTTTTTTT    Type (1) Jump
J10EXCCCCCCCCC00TTTTTTTTTTTTTTTT    Type (2) JSR
J01EXCCCCCCCCCC00XXXXXXXXXXXXXXXX   Type (3) Conditional Return
01REX00IIIIIIIDDDDDDDDDIIIIIIIII    Type (4) Immediate Data
M0REXSSSSSSSSSDDDDDDDDDNNNNNNNNN    Type (5) General Operation
```

KEY

J - Reverse jump sense bit. Reverses logical sense of condition.

E - E bit. Reverses sign of carry input to ALU

X - An unused bit.

C - The condition code field.

T - The jump target field.

I - The immediate data field. Least significant seven bits are in first field, most significant nine bits in second field.

S - The nine bit source field.

D - The nine bit destination field.

N - The nine bit Am 2901 control field.

R - The return bit. Cause unconditional return combined with (4) or (5) type instruction.

M - The multiple precision bit. When on the ALU carry in is fed from the carry bit in the status register.

4.9.2.1 Arithmetic General Operation Instructions (Type 5). The arithmetic operations are all General Operation (Type 5) instructions involving the Am 2901 arithmetic unit as the selected device in either the source field, destination field or both. Full control of the Am 2901 is only achieved when both the source and destination fields are devoted to selecting the Am 2901. In other cases enough bits are not available for full control. In practice this means that instructions involving shifts are not possible when the source or destination is external to the Am 2901. The source or destination select code for the Am 2901 is of the form: 10XXXXXXX. This means that only seven control bits are available if the Am 2901 is not both the source and destination. The 9-bit Am 2901 control field is partly appropriated to provide sufficient control when only one of the source and destination fields is available and this is the reason for the limited control.

The Am 2901 requires a 4-bit A field and a 4-bit B field to specify the source and destination address in the 16-word register file contained within the Am 2901. The nine-bit, Am 2901 control field determines the arithmetic and logic unit (ALU) operation, destination of data and the type of shifting to be applied, if any.

EXAMPLE 1: Am 2901 Is The Source, Destination Other Device

Source Field:

Bit	8	7	6	5	4	3	2	1	0
Data	1	0	B2	B1	B0	A3	A2	A1	A0

Am 2901 Control Field:

Bit	8	7	6	5	4	3	2	1	0
Data	B3	I7	I6	I5	I4	I3	I2	I1	I0

The register field and B addresses for the Am 2901 are shown as A0-A3 and B0-B3. The Am 2901 control field bits are shown as I0-I8, to conform to the manufacturer's notation. In example 1, the bit normally assigned to I8 has been used to provide B3, since there are not enough bits in the source field. The Am 2901 unit sees a zero for bit I8.

EXAMPLE 2: Am 2901 Is Destination, Source Is Other Device:

Destination Field:

Bit	8	7	6	5	4	3	2	1	0
Data	1	0	A2	A1	A0	B3	B2	B1	B0

Am 2901 Control Field:

Bit	8	7	6	5	4	3	2	1	0
Data	I8	I7	I6	I5	I4	I3	A3	I1	I0

In example 2, the A and B addresses are obtained from the destination field with the exception that A3 is obtained from the bit which would normally be control field bit I2. The Am 2901 sees a one for bit I2.

EXAMPLE 3: The Am 2901 Is Both Source and Destination:

Source Field:

Bit	8	7	6	5	4	3	2	1	0
Data	1	0	ST	D1	D0	A3	A2	A1	A0

Am 2901 Control Field:

Bit	8	7	6	5	4	3	2	1	0
Data	1	0	X	C1	C0	B3	B2	B1	B0

In example 3, both the source and destination fields are available. The Am 2901 control field is fed directly to the Am 2901, without using any bits to generate A and B addresses. The A and B addresses come directly from the least significant parts of the source and destination fields. The remaining bits in the source and destination fields are used to provide the following control bits:

(1) ST The store control bit,
(2) D1-D0 The shift control bits, and
(3) C1-C0 The ALU function modification bits.

The meaning of the shift control bits D1-D0 is shown in Table 3-1 and Figure 3-5. The store control bit, ST, must be a one for the result to be stored in the register file or the Q register of the Am 2901. The ST bit is only relevant for case 3 above. For case 1 and case 2, the storage or non-storage of the result depends upon the code in the Am 2901 control field. The control field code can also be used to inhibit storage of the result for case 3 above, but using this method of inhibiting the storage of the result is incompatible with shifts.

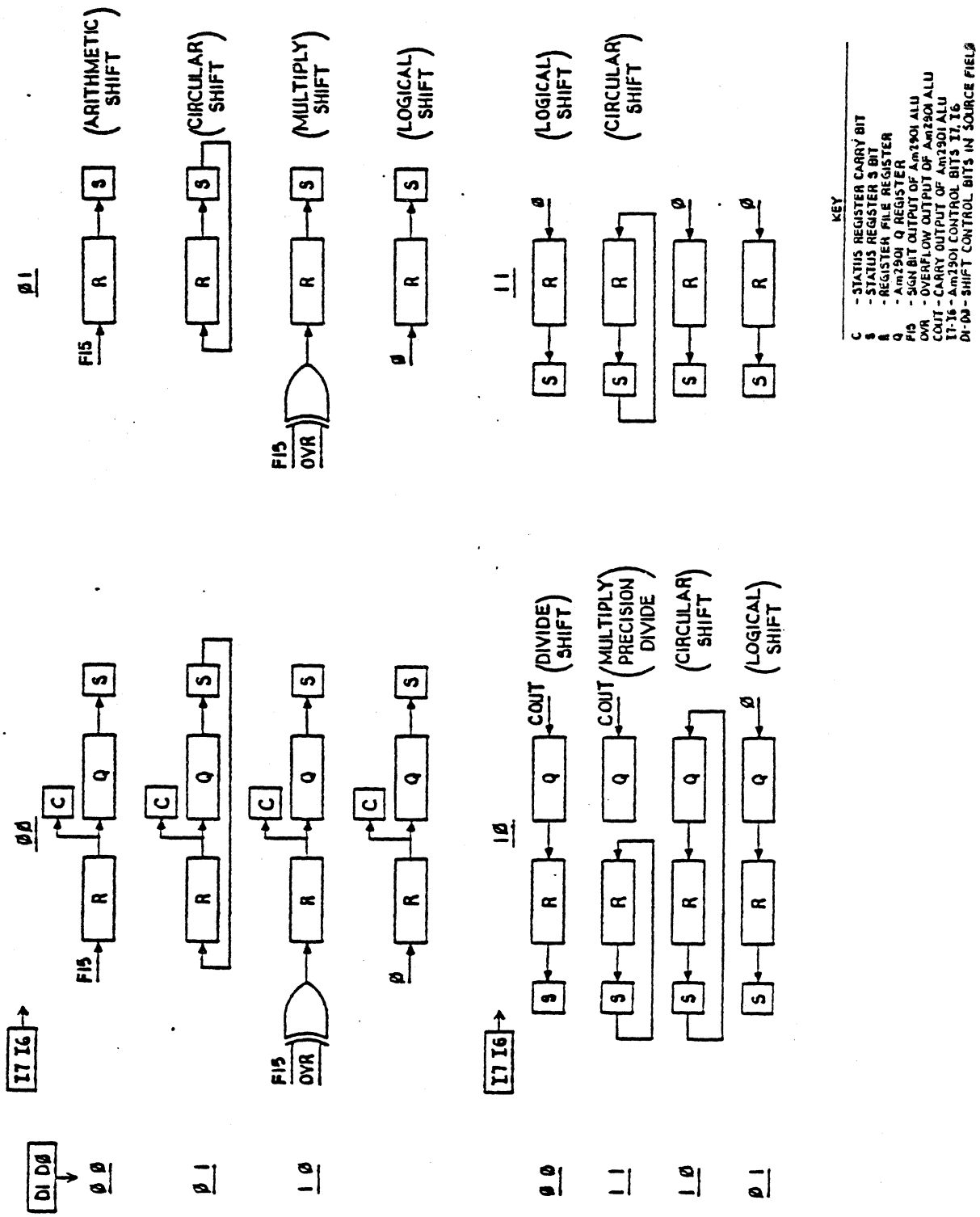


Figure 4-10. Am 2901 Shift Modes

Table 4-5. Shift Control

CONTROL BITS I7	I6	D1	D0	SHIFT FUNCTION	
0	0	0	0	F15 → R-REG	→ Q-REG → S → C
0	0	0	1	S → R-REG	→ Q-REG → S → C
0	0	1	0	XFO → R-REG	→ Q-REG → S → C
0	0	1	1	0 → R-REG	→ Q-REG → S → C
0	1	0	0	F15 → R-REG → S	
0	1	0	1	S → R-REG → S	
0	1	1	0	XFO → R-REG → S	
0	1	1	1	0 → R-REG → S	
1	0	0	0	S ← R-REG	← Q-REG ← COUT
1	0	0	1	S ← R-REG	← Q-REG ← 0
1	0	1	0	S ← R-REG	← Q-REG ← S
1	0	1	1	S ← R-REG ← S	Q-REG ← COUT
1	1	0	0	S ← R-REG ← 0	
1	1	0	1	S ← R-REG ← 0	
1	1	1	0	S ← R-REG ← 0	
1	1	1	1	S ← R-REG ← S	

Key to Shift Control

F15	Sign bit of ALU output.
S	Status register S bit.
C	Status register carry bit.
COUT	Carry out of ALU.
XFO	Exclusive OR of ALU F15 and OVR outputs. This input repairs an overflow that happens during ALU opertion.
I7, I6	Bits in Am 2901 control field.
D1, D0	Shift control bits.

NOTES

(1) S and C are flip-flops.

(2) F15, COUT, XFO are outputs from current ALU operation.

(3) The status register D bit is set by the COUT for certain C1, C0 control codes.

(4) The shifter is located between the ALU output and the registers.

The ALU function modification bits, C1-C0, cause certain modifications of the Am 2901 operation depending upon either the D bit of the status register (divide control) or the least significant bit of the Q register (multiply control). The effects of the possible C1-C0 codes are as follows:

0 0 - No modification to the Am 2901 control field.

0 1 - Control field bit I3 is inverted if the status register D bit is a one. The effect of inverting I3 is to change an addition to a subtraction, or vice-versa. The D bit provides divide control.

1 1 - I3 is inverted if the D bit is a one and at the end of the cycle the D bit is set to the value of the ALU carry out.

1 0 - Control bit I1 is inverted if the Q0 output is a one. The effect of inverting I1 is to change the ALU source input "A" from the register file to the constant zero, or vice versa. This is used to implement a multiply step, where either the multiplier, or zero, is added depending upon the least bit of the Q register.

The M and E bits influence the operation of the Am 2901 for cases 1, 2 and 3 above. For instructions not involving the Am 2901, the M and E bits are ignored. Together, the M and E bits control the carry input to the ALU. If both M and E are zero, the carry input is zero for an add operation and one for a subtract operation. This is the normal mode of operation. If the M bit is set to a one, the carry in is fed from the status register C bit. This allows multiple precision adds and subtracts. If the E bit is a one, the carry in is inverted, whatever its source. This allows increment and decrement operations. ALU operations which are not add or subtract are not influenced by the M or E bit settings.

4.9.2.2 Immediate Data Instruction (Type 4). The immediate data instruction places 16 bits of data on the bus. This data is specified within the instruction word. The data is stored at the bus destination specified by the destination field in the instruction word. The least significant 7-bits of data come from bits 24-18 of the instruction word. The most significant 9-bits of data come from bits 8-0 of the instruction word. Specifying the Am 2901 as the destination of the immediate data is a special case, because the Am 2901 control field is taken up by the most significant part of the data. If the Am 2901 is selected as the destination the control field seen by the Am 2901 will have the code: 01D011111. The "D" means that the bit follows the data specified. However this uncontrolled bit does not influence the effective action of the Am 2901 which is to store the bus data in the register file location specified in the destination field as the B field. It is not possible to store immediate data directly in the Q register.

4.9.2.3 Unconditional Return (Immediate Data Type 4 and General Operation Type 5. If bit 29 of the instruction word is a one, for Immediate Data (Type 4 or General Operation (Type 5) instructions, an unconditional return is executed at the same time as the instruction. This means that the next instruction executed is not the next sequential instruction, but an instruction located at an address taken from the subroutine stack.

An unconditional return loads the instruction counter and the priority register from the subroutine stack. This results in a return to the instruction following a previous JSR (Type 2) or a return to the instruction following the last instruction executed before an interrupt. Usually a different priority is loaded only if the return is from an interrupt servicing routine, although it is possible that the interrupt priority had been explicitly changed in the subroutine and is restored to the previous value as a result of the return.

4.9.2.4 Jump and JSR (Type 1 and 2) Instructions. If the condition test is passed as selected by the source/condition field, the reverse jump sense bit to the target address is executed. The JUMP (Type 1) is a simple transfer of control. The JSR (Type 2) saves the return address and the current priority on the subroutine stack for a later return.

4.9.2.5 Conditional Return (Type 3) Instruction. The condition is tested as for Jump (Type 1) and JSR (Type 2) instructions. A return is executed if the condition is true, otherwise the execution continues with the next instruction.

4.9.2.6 Map Register ORing. A 4-bit field can be stored with the target address for a Jump (Type 1) or JSR (Type 2) instruction or with the return address for a Conditional Return (Type 3) instruction. There are four choices of 4-bit Conditional Return fields, selected by special condition codes. A proper setting of the reverse jump sense bit assures that a jump or control transfer always takes place (otherwise the instruction would be a NOP). The effective destination address, where the next instruction is located, is the OR of the normal destination and the 4-bit field ORed into the least significant 4 bits. The four choices available are as follows:

(1) Map register lower nibble,
(2) Map register upper nibble,
(3) Status Register B-bit nibble,
(4) Code digit of the map register.

These choices are discussed more fully under the sections on the map register and status register.

The significance of the ORing is that a multiway jump (which is a function of data) can be accomplished. If the destination address has its least significant 4 bits zero it is a 16-way jump. If one of the destination bits is a one, then the jump is reduced to an 8-way jump, and so forth.

4.9.3 Interrupts and System Priority Level

The 3-bit priority register holds the current system priority level, from 0 to 7. An interrupt may have a hardware priority from 1 to 7. An interrupt will take place immediately if and only when the interrupt priority (determined by hardware) is higher than the system priority. If several interrupts are pending, the highest priority interrupt takes precedence. Precedence between interrupts of the same priority is determined by hardware interconnect, or in some cases, by the interrupt routine. If the system priority is 7, no interrupts can take place, because there is no interrupt priority of level 8. If the system priority is 0, then all interrupts can take place.

When an interrupt takes place, the current instruction counter and priority level are stored on the subroutine stack. The instruction counter is loaded with an interrupt vector address, between 0 and 255, as determined by hardware wiring. The instruction located at the interrupt vector address is then executed. Usually it is a jump instruction pointing to the interrupt servicing routine.

Higher priority interrupts generally can interrupt servicing routines for lower priority interrupts. The exception would be if the interrupt servicing routine changes the interrupt priority level.

4.9.3.1 Instruction and Interrupt Conflicts. Change of control instructions generally conflict with interrupts, since both want to change the instruction counter, and in some cases the priority level and subroutine stack. In the case of a conflict with a Jump (Type 1), JSR (Type 2) or Unconditional Return (Type 3) instruction, the instruction is aborted by the hardware and the interrupt takes place. The address stored on the subroutine stack will be that of the aborted instruction, so that the aborted instruction is read from control memory a second time after the return from the interrupt and executed. In the case of an Immediate Data (Type 4) or General Operation (Type 5) instruction combined with an unconditional return, the instruction is executed normally, but the associated unconditional return is not executed. The unconditional return is deferred until the return from the interrupt routine. The interrupt sequence in this case does not modify the subroutine stack, leaving it as is, since the return address is already on the top of the stack. These conflicts are usually not visible to the microprogrammer, but are explained here to aid understanding of the interrupt sequence.

4.9.3.2 Changing the Priority Level. The priority register can be set to one of eight possible values by using one of eight different condition codes in a Jump (Type 1), JSR (Type 2) or Unconditional Return (Type 3) instruction. This method of setting the priority register is used because it avoids the conflict of an interrupt and an instruction both trying to set the priority register to different values at the same time. The conflict is avoided because Jump (Type 1), JSR (Type 2) or Unconditional Return (Type 3) instructions are aborted (to be executed later) if they conflict with an interrupt. The condition test is controlled totally by the reverse jump sense bit. The priority will be set whether or not the condition test is passed.

Normally a Jump (Type 1) instruction is issued to set the priority. If a JSR (Type 2) is used, the priority stored on the subroutine stack will be the old priority (if the condition test is passed and the JSR (Type 2) instruction executes. This feature can be used to change the priority only for the duration of one subroutine. A Conditional Return (Type 3) instruction will only change the priority to the value specified in the condition code if the return does not execute. If it executes, the priority comes from the subroutine stack.

4.9.4 Microprocessor Registers

In the sections that follow, the use and function of the various registers which may be accessed as sources and destinations on the bus are described. When mnemonics are used they are the mnemonics used by the Am 2901 microassembler.

4.9.4.1 Status Register. The status register is an 8-bit register which may be read and written as a data register attached to the bus, but in addition, the various bits which make up the status register may be set and cleared independently by hardware to reflect the status of the microprocessor. When read or written as data, the status register communicates with bits 8-15 of the data bus. The other 8 bits can be used, optionally, by the index register. The source and destination code are as follows:

662 (SR) - If used in the source field the status register is output to the high byte of the bus. Zero is output to the low byte. If used as a destination, the high bus byte is stored in the status register, the low bus byte is ignored.

663 (XRSR) - May be used in either the source or destination fields. The status register communicates with the high bus byte, the index register with the low byte.

The 8-bits of the status register are defined as follows:

Bit	15	14	13	12	11	10	9	8
	S	D	B	V2	N	X	V	C

Status Register Key Bit	Function
S	The S bit is set by various shift instructions (see figure 4-10).
D	The D bit is set as a part of the divide algorithm. See the arithmetic general operation (Type 5) instructions for details.
B	The B bit is set to zero when the low byte of the bus is stored in the map register. The B bit is set to one when the high byte of the bus is stored in the map register. The function of the B bit is to aid in constructing a stream of bytes from a stream of words.

V — The V bit is set or reset when the Am 2901 performs an add or subtract operation. If the result is wrong because of an overflow the V bit is set to one, otherwise zero. Non-Am 2901 operations, or non-add or subtract operations do not affect the value of the V bit.

V2 — The V2 bit is set under the same conditions as the V bit, but it is reset from a one to a zero only by storing data in the status register. It is an overflow bit that "sticks" on. The purpose is to check a chain of calculations for overflow, without having to waste the execution time for a test after every operation.

N — The N bit signifies a negative output from the Am 2901 ALU. It is set on, or off, on any operation where the Am 2901 is a source or destination. In other operations its value is not affected.

Z — The Z bit indicates a zero output from the Am 2901 ALU. It is set or reset under the same conditions as the N bit. For both the N and Z bit the output of the ALU is tested before any shift of the data.

C — The C bit is set with the carry out of the Am 2901 ALU on an add or subtract operation except on the double right shift operation where it is set to the value of the bit shifted out of the register file part of the double right shift. If an operation is both add or subtract and double right shift, the double right shift controls the C bit. Operations other than those mentioned, or non-Am 2901 operations do not affect the value of the C bit.

In the case of a conflict in setting the status register, where a direct store from the bus is trying to set all bits and internal conditions are trying to set one or more bits, the direct store from the bus takes precedence.

Normally, the status register and the index register are saved on the data stack during interrupts. The combined mode of accessing the index register and the status register makes this faster.

4.9.4.2 Status Register Condition Codes. Condition codes exist so that conditional jumps may be made on each of the status register bits. In addition there exist condition codes for certain logical functions of the status register bits.

In the following table the 9-bit condition codes are given as octal numbers. A minus sign means that the reverse jump sense bit is a one; that is, the logical condition is reversed. The mnemonic names are those of the Am 2901 microassembler.

Table 4-6. Status Register Condition Codes

Mnemonic Code	Octal Numbers	Definition
LE	(640)	Less than or equal to zero. The Z bit is on, or else the N bit is on and the V bit is off or the N bit is off and the V bit is on.
QT	(-640)	Greater than zero. The logical reverse of LE.
LT	(641)	Less than zero. The N bit is on and the V bit is off or the N bit is off and the V bit is on.
GE	(-641)	Greater than or equal to zero. The logical reverse of LT.
CC	(642)	The carry bit is clear (zero).
CS	(-642)	The carry bit is set.
VC	(643)	The overflow bit (V) is clear (zero).
VS	(-643)	The overflow bit is set.
NE	(644)	Not equal to zero. The Z bit is clear.
EQ	(-644)	Equal to zero. The Z bit is set.
PL	(645)	Plus. The N bit is clear.
MI	(-645)	Minus. The N bit is set.
V2C	(646)	V2 is clear.
V2S	(-646)	V2 is set.
BC	(647)	The B bit is clear.
BS	(-647)	The B bit is set.
DC	(650)	The D bit is clear.
DS	(-650)	The D bit is set.
SC	(651)	The S bit is clear.
SS	(-651)	The S bit is set.
NEV	(652)	Never true. Will not jump.
U	(-652)	Unconditional. Always jumps.
MRB	(-654)	If the B bit is on, the number 4 is ORed with the jump destination. If the B bit is off, zero is ORed with the destination, for no net change.

4.9.4.3 Map Register. The microprocessor map register is an 8-bit register which may be stored into from either the high or low byte of the data bus. The map register can supply either the high or low byte to the data bus while zeroing the other bus byte. Because of its ability to work from or to either bus byte the map register is useful for swapping bus types and performing shifts by 8 places on multiple precision numbers. In addition, multiway jumps may be performed where either the upper or lower 4-bits of the map register are ORed in with the destination address. Always computed by hardware as a function of the current contents of the map register is the code digit, a 4-bit number which also may be used to control multiway jumps. The code digit functions are defined in the following table:

Table 4-7. Map Register Code

Map Register	Code Digit	
76543210	3210	
0xxxxxxx	0110	(6)
10xxxxxx	0111	(7)
110xxxxx	1000	(8)
1110xxxx	1001	(9)
11110xxx	1010	(10)
111110xx	1011	(11)
1111110x	1100	(12)
11111110	1101	(13)
11111111	1110	(14)

x - Indicates don't care bit.

As can be seen from the table above the code digit is related to the number of shifts needed to normalize a negtive number stored in the map register.

The source and destination codes associated with the map register are as follows:

MRLO (666) Store the lower bus byte in the map register. (Destination)

MRHI (665) Store the upper bus byte in the map register. (Destination)

MRLO (665) Read the map register to the lower bus byte, zero to the upper bus byte. (Source)

MRHI (666) Read the map register to the upper bus byte, zero to the lower byte. (Source)

The following condition codes are associated with the map register:

MRC	(-665)	OR the code digit with the jump destination.
MRL	(-656)	OR the lower 4-bits of the map register with the jump destination.
MRU	(-657)	OR the upper 4-bits of the map register with the jump destination.

4.9.4.4 Scratchpad and Stack Memory. The scratchpad unit consists of a 256 word by 16-bit memory which may be addressed in several different ways with the aid of two 8-bit addressing registers, the index register and the stack pointer register. The stack pointer register is used to implement a last-in, first-out stack in a part of the 256 word memory. The index register can be used to address the memory by 16 word pages, or it can be used to implement a stack function. When either of the address registers is used to implement a stack, the register is incremented before any data is stored into memory and decremented after data is removed from memory. The memory address is always the sum of the contents of the address register and the last 4 bits of the source or destination code. The actual increment or decrement takes place at the end of the memory access cycle. The pre-increment is accomplished by temporarily adding one to the address during the cycle to simulate an increment at the beginning of the cycle.

The significant source and destination codes are as follows:

XR	(661)	As a destination store the lower 8-bits of the bus in the index register. As a source put the lower 8-bits on the bus and zeros in the upper 8 bus bits.
XRSR	(663)	Use as source or destination to transfer between the index register in the lower 8-bits and the bus.
XSTK	(664)	Use as a source or destination to communicate between the lower 8 bus bits and the stack pointer register. If a source, the upper 8 bus bits are set to zero.
STK	(637)	Use as source or destination to push or pop from the stack defined by the stack pointer register (XSTK). The top of the stack is located at the address pointed to by XSTK. The stack works by incrementing the stack pointer after data is read from the scratchpad. When data is written into the scratchpad the stack pointer is decremented before the write operation. This works the same as the stack in the PDP-11.
SP0,SP1-SP15	(600,601-617)	Use as a source or destination to address the 16 locations of the scratchpad starting at the location pointed to by the index register.

XST0,XST1-XST14	(620, 621-636)	Address the location pointed to by the index register with 0 to 14 added and a pre-increment if a source or a post-decrement if a destination. Can be used to establish a stack or to move successive words in or out of memory in a loop. Note that this stack function is different from STK, in that the pointer register (XR) is post decremented for a write and pre-incremented for a read.

It is impossible for the scratchpad to be both a source and destination in the same instruction. The scratchpad memory and either of the addressing registers are permitted to both be addressed in the same instruction. No confusion will result from this if it is remembered that the pre-increment is only simulated and that a load to the addressing register suppresses any counting.

4.10 PROM BOARD

This paragraph is supported by the illustrations listed below.

Block Diagram, PROM Board	Figure 4-11
Schematic Diagram, PROM Board	2501-4701-2D

The PROM board contains the program microinstruction memory and the stack-scratchpad memory discussed in the paragraph on the Processor, and several miscellaneous functions including real time clock, interrupt interface, and fractional multiplier. These circuits are described individually in the following paragraphs. Note that printer interface circuits existing on this board are not used in the 2515 system and will not be described herein.

4.10.1 Program Instruction Memory

This memory contains instructions which are read out and executed by the 2501 microprocessor. Input is 16 address lines from the instruction counter which enter at the card edge (IC0-IC15). The least significant 10 bits are buffered and inverted by U40 and U42 (S240). Buffered address lines RA0-RA9 are fed directly to the PROM chips. The most significant bits of the address lines are decoded to provide chip select signals to various banks of PROMS (see Figure 4-12). Output is 32 PROM data lines, labeled ROMDAT 0-8, D0-D8, S0-S8 and ROMDAT 27-31.

There are 6 banks of PROMS, U16-U39, on the board. Each bank provides 512, 1024 or 2048 instructions of 32 bits (see Table 4-8).

Table 4-8. Relation of PROMs to Output Data

	OUTPUT BITS				BANK OCTAL ADDRESSES BY PROM TYPE		
Bank 1	Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31	512x8	1024x8	2048x8
Bank 0	U32	U20	U38	U26	0-777	0-1777	0-3777
Bank 1	U31	U19	U37	U25	1000-1777	2000-3777	4000-7777
Bank 2	U30	U18	U36	U24	2000-2777	4000-5777	10000-13777
Bank 3	U29	U17	U35	U23	3000-3777	6000-7777	14000-17777
Bank 4	U28	U16	U34	U22	4000-4777	10000-11777	20000-23777
Bank 5	U33	U21	U39	U27	5000-5777	12000-13777	24000-27777

Note: Address decoding is jumper selected for the PROM size used. In the 2515 system the 2048 x 8 PROM is used.

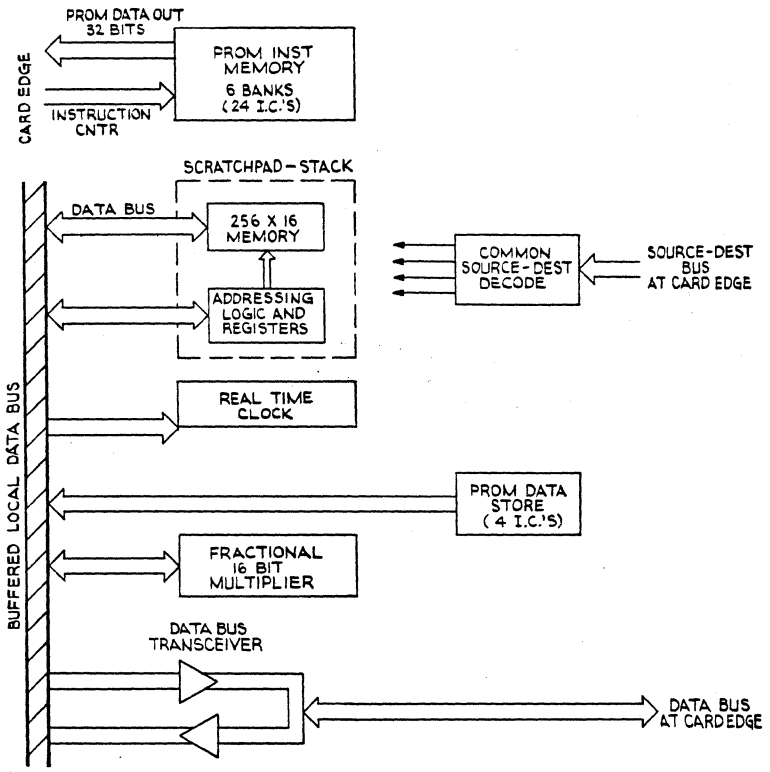


Figure 4-11. PROM Board Block Diagram

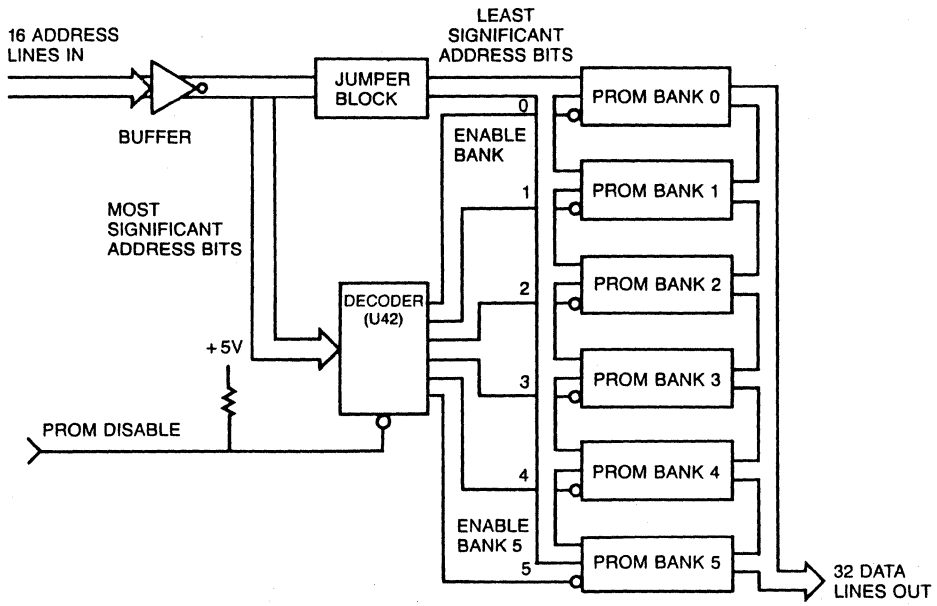


Figure 4-12. Instruction Memory Block Diagram

4.10.2 Stack Scratchpad Memory

This unit is built around a 256 x 16 bit fast bipolar memory. The memory can be written from or to the microprocessor data bus in various ways. The index register (XR) and the stack pointer register (XSTK) are each 8-bit up-down counters used to generate and hold the 8-bit addresses used to address memory (see Figure 4-13). All addresses are generated modulo 256. Address 0 follows address 255. There are two modes of addressing available using the index register (XR). In one mode the index register is set to an address which represents the base of a 16 word page. Any of 16 words can then be written or read with 16 source and destination codes (SP0-SP15) without resetting the index register. In the other mode of addressing (the index register) a 15 word page is accessible and the index register is incremented before a word is read or after a word is written, thus shifting the whole page up or down. This pre-increment, post decrement mode of addressing may be used to read successive words in a loop or to implement a stack using the index register.

The stack pointer register (XSTR) has only one mode of addressing. A store with destination code 637 does a push to the stack. A read with source code 637 does a pop from the stack. The stack pointer is incremented for a pop (post increment) and decremented for a push (pre-increment) (see Figure 4-14). The stack grows toward lower addesses. When the stack pointer is used to address memory the address is formed by adding together the contents of the stack pointer register, the least 4 bits of the source or destination code which is always 15 and a 1 if the operation is a pop. All actual incrementing and decrementing of the counters is always done at the end of the read or write cycle on the master clock. But when a pre-increment is desired a one is added to the address with an adder during the cycle so that the effective address is incremented at the start of the cycle, although the counter only counts up at the end of the cycle.

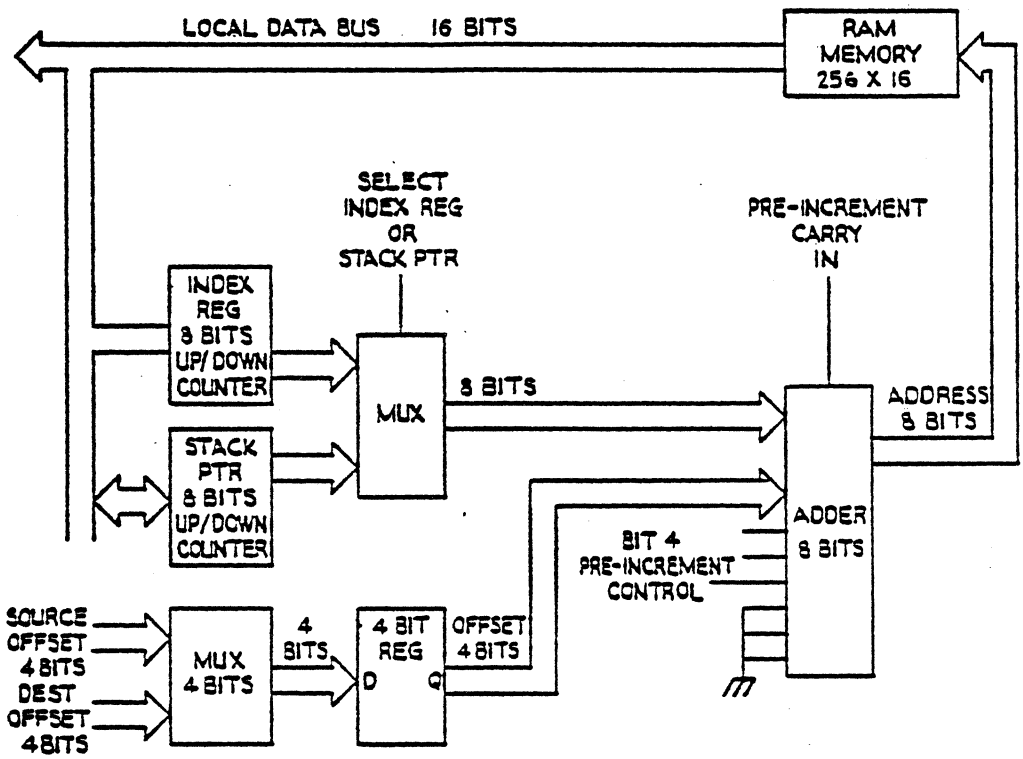


Figure 4-13. Scratchpad Stack Block Diagram

Table 4-9. Source and destination Codes

Source/ Destination Code			4SELS	Count Stack PTR	Count Index Reg	Addressed Source	Location DEST	Mnemonic
6	0	0	0	0	0	XR+16	+0	SP0
6	0	1	0	0	0	XR+16	+1	SP1
6	0	2	0	0	0	XR+16	+2	SP2
6	0	3	0	0	0	XR+16	+3	SP3
6	0	4	0	0	0	XR+16	+4	SP4
6	0	5	0	0	0	XR+16	+5	SP5
6	0	6	0	0	0	XR+16	+6	SP6
6	0	7	0	0	0	XR+16	+7	SP7
6	1	0	0	0	0	XR+16	+8	SP8
6	1	1	0	0	0	XR+16	+9	SP9
6	1	2	0	0	0	XR+16	+10	SP10
6	1	3	0	0	0	XR+16	+11	SP11
6	1	4	0	0	0	XR+16	+12	SP12
6	1	5	0	0	0	XR+16	+13	SP13
6	1	6	0	0	0	XR+16	+14	SP14
6	1	7	0	0	0	XR+16	+15	SP15
6	2	0	0	0	1	XR+16+1	XR+16+0	XST0
6	2	1	0	0	1	XR+16+2	XR+16+1	XST1
6	2	2	0	0	1	XR+16+3	XR+16+2	XST2
6	2	3	0	0	1	XR+16+4	XR+16+3	XST3
6	2	4	0	0	1	XR+16+5	XR+16+4	XST4
6	2	5	0	0	1	XR+16+6	XR+16+5	XST5
6	2	6	0	0	1	XR+16+7	XR+16+6	XST6
6	2	7	0	0	1	XR+16+8	XR+16+7	XST7
6	3	0	0	0	1	XR+16+9	XR+16+8	XST8
6	3	1	0	0	1	XR+16+10	XR+16+9	XST9
6	3	2	0	0	1	XR+16+11	XR+16+10	XST10
6	3	3	0	0	1	XR+16+12	XR+16+11	XST11
6	3	4	0	0	1	XR+16+13	XR+16+12	XST12
6	3	5	0	0	1	XR+16+14	XR+16+13	XST13
6	3	6	0	0	1	XR+16+15	XR+16+14	XST14
6	3	7	1	1	0	XS+16	XS+15	STK

Additional Associated Source & Destination Codes

S661)
S663) Read index register to lower 8-bits of bus

D662)
D663) Store lower 8-bits of bus to index register

S664 Stack pointer to lower 8 bus bits

D664 Lower 8 bus bits to stack pointer

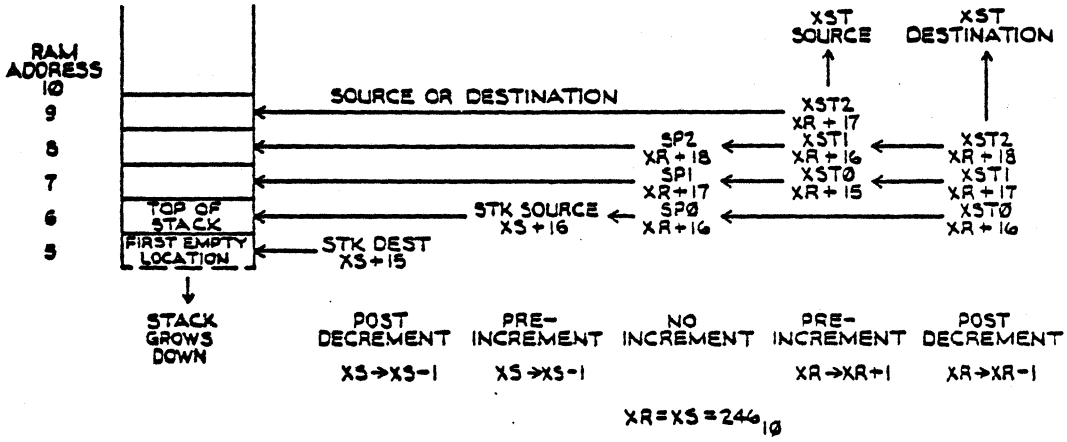


Figure 4-14. Example of Scratchpad Addressing Modes

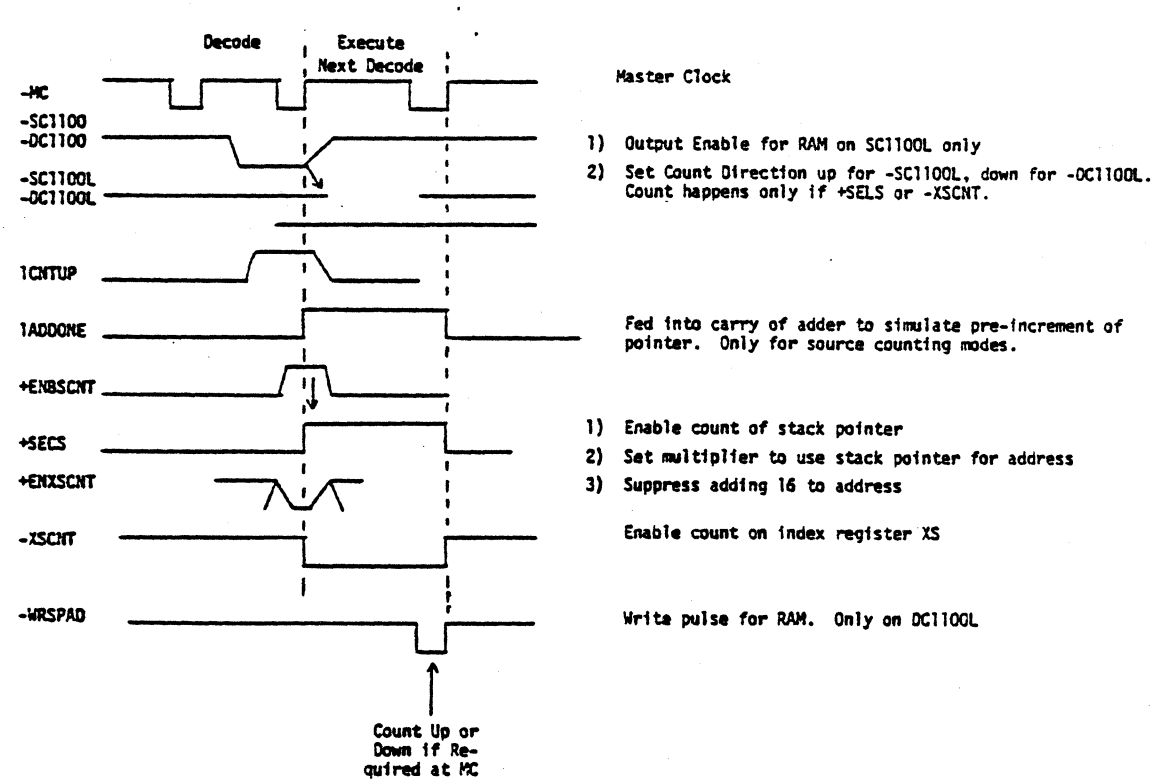


Figure 4-15. Stack/Scratchpad Generalized Timing Chart

4.10.3 Real Time Clock

The real time clock is a 16 bit counter which counts every 4 microseconds. The 4 microsecond period signal which drives the 16 bit counter is derived from the 16 MHz master clock available on the backplane (see figure 4-16). Actual counting happens at the master clock, rather than exactly at the 4 microsecond clock. The first master clock after the 4 microsecond clock makes a transition is used to count the real time clock. Thus, there is some jitter in the time at which the clock counts (never more than one microsecond) but there is no cumulative error. The clock counter is a ripple counter which takes about 250 nanoseconds to count due to the time for the carry to ripple from one end to the other (see figure 4-17). Because the real time clock counts synchronously with the master clock it can be read directly onto the data bus (through a buffer) without fear that it will be changing its value as it is being read. However it does take 250 nanoseconds to settle down after the master clock. Therefore it is necessary to stretch any instruction involving the clock as a source so that the clock will be stable by the time the data is clocked into the destination. When the clock is supplied to the bus, a delay of 6 clocks (375 nanoseconds) is enabled. The clock is never cleared or reset except when the microprocessor is initialized. In that case, it is cleared to zeros. The period of the clock is 262.144 milliseconds with the 16 MHz clock on the backplane.

Associated with the real time clock is a 16-bit clock compare register (U105 + U106) and a 16 bit equal-not equal comparator (U107 and U108). The clock compare register may be loaded from the microprocessor bus. Just before the clock is counted the contents of the clock are compared with the contents of the clock compare register. If they are equal, a +EQUAL signal is generated. The +EQUAL signal may cause an interrupt of the microprocessor or cause action in the printer interface. Both the count and the compare happen at the rising edge of the +CNTRU signal. An equal compare signal is captured by flip-flop (U116) at this time.

The final register associated with real time clock is the 8-bit clock hold register. The object of the clock hold register is to overcome the effects of interrupt latency so that the firmware can tell the exact time at which an interrupt signal (I0) was requested. The clock hold register captures the least 8 bits of the clock when interrupt flip-flop (U113) I0 is enabled (part of interrupt-printer interface). The data is captured synchronously with the master clock. Since only the least 8-bits are captured the interrupt latency must be less than 255 clock counts, or 1 millisecond, for the firmware to be able to reconstruct the time at which the clock hold register was clocked.

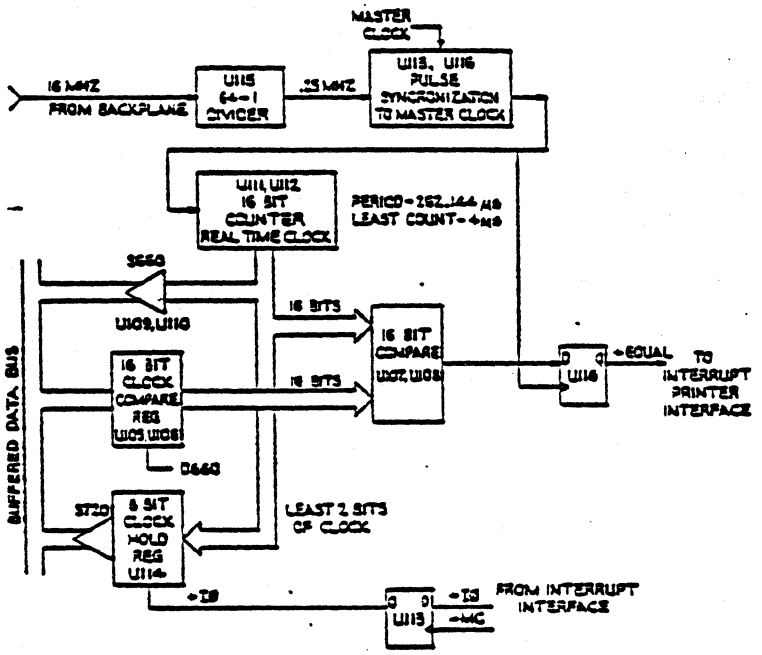


Figure 4-16. Real Time Clock Block Diagram

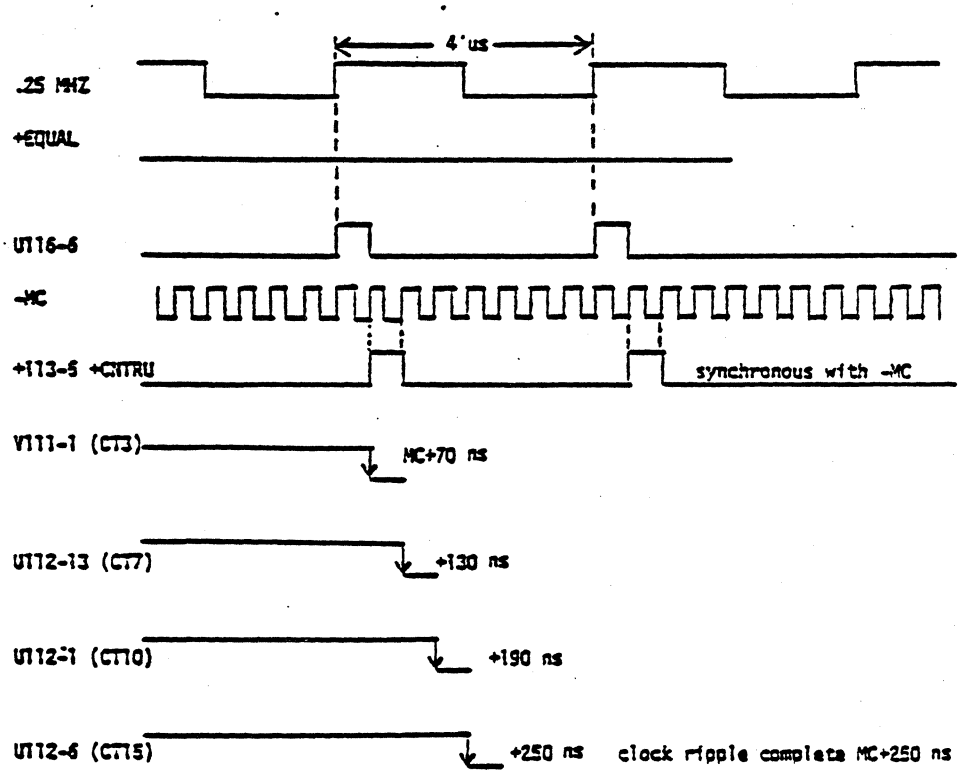


Figure 4-17. Synchronization with Master Clock

4.10.4 Interrupt Interface

There are 4 interrupt flip-flops in the interrupt interface, designated I0 -I3 (see figure 4-18). If any of these flip-flops are set active a 2501 microprocessor interrupt is requested (204, base 8, priority 6). The flip-flops are not reset by the interrupt but must be reset by the firmware which can read the state of each and reset each. To prevent multiple interrupts it is the responsibility of the firmware to reset all the flip-flops before lowering the interrupt priority in the interrupt servicing routine. The microprocessor firmware has the option of disabling any or all of the interrupt flip-flops by leaving the reset bits at zero, thus holding the flip-flop corresponding to the reset bit in a reset condition. The flip-flops are set by negative to positive transition on the clock input. The clock inputs to the various flip-flops are as follows:

I3 Set when an equal compare condition is detected in the real time clock. Principal function of I3 is to turn off the printer heaters at the end of a burn but may also be used to generate an interrupt at a time held in the clock compare register.

I2 Set when the real time clock overflows. That is when the count changes 65536 to 0. Purpose is to allow firmware to extend the clock to multiple word length.

I1 Set by a transition on input pad T2. The sense of the transition recognized can be inverted by changing control bit SN1. An interrupt can also be caused by cycling SN1 through a transition, holding the input pad T2 constant (e.g., nothing connected to T2). The major function of I1 is to create an artificial interrupt by cycling SN1 to call the task arbitrator.

I0 The clock input to I0 is from a 1 of 8 selector. In addition, the polarity of the input can be reversed by control bit SN0. Thus I0 is multifunctional. The functions which can be selected are:

(1) Print zone on T80 printer,
(2) Dot row strobe on T80 printer,
(3) Character strobe on T80 printer,
(4) DECIN or pad T2, the same signal as I1 uses,
(5) Paperfeed button on case of T80 printer pressed,
(6) Input pad T4,
(7) Input pad T5,
(8) Equal compare, the same signal as used by I3.

The main functions of I0 are to turn on the printer heaters at the dot row strobe and to cause an interrupt if the operator pushes the paperfeed button. Which of the 8 inputs is selected is controlled by the binary value in the 3 control bits SEL0, 1, 2. These may be set by the firmware. The clock input to I0 is filtered so that pulses shorter than about 200 nanoseconds are ignored. This is to prevent noise on long printer cables from causing false triggers.

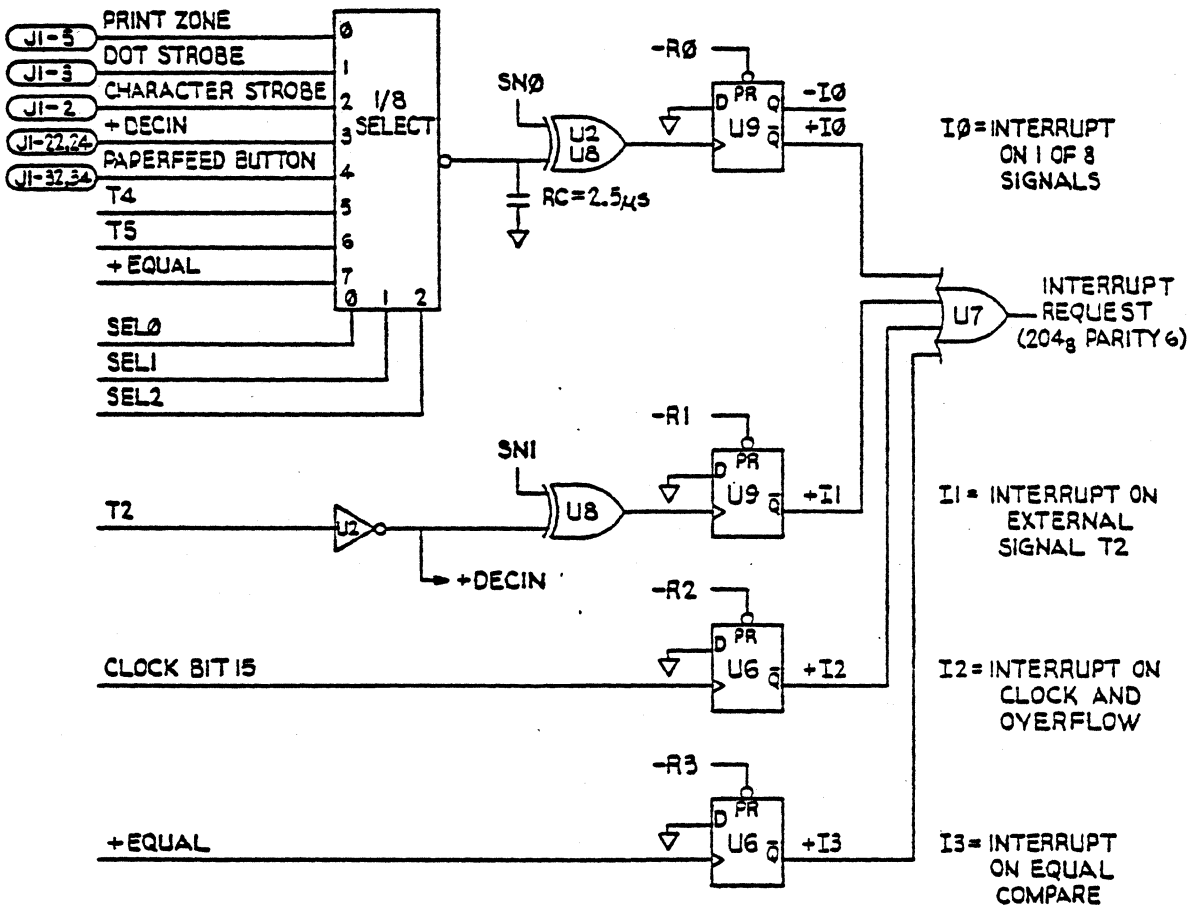


Figure 4-18. Interrupt Interface Block Diagram

4.10.5 Fractional Multiplier

The fractional multiplier uses 2 AMD chips (25LS14) to do the multiply. A 16 bit shift register holds one of the multipliers and the result. To perform a multiply the first number is loaded in the shift register (D722). See Figure 4-19. This clears the 4-bit cycle counter (U98). The second number is loaded into the multiplier at D723. Loading the second number starts the multiply which requires 33 clocks of the 16 MHz clock. The first clock loads the least significant bit from the shift register to the sign extend flip-flop (U100) and loads a one bit into the multiplier so that the result will be rounded. The following 32 clocks perform the multiply. After the sign bit enters the sign extend register, the contents of the sign extend register are frozen for the remaining clocks, to extend the sign as shown in figure 3-16. The result is read from shift register S722. The result is scaled so that 32768 1.0, 16384 = .5, etc. If 16384 x 16384 are multiplied, the result will be 8192. Positive and negative numbers are acceptable and the sign will follow the rules of arithmetic. The multiplication -32768 x 32768 will result -32768 which is incorrect and an overflow condition due to the fact that +32768 cannot be represented in 16-bits in 2's complement notation. Rounding always takes place and is effected by adding +1/2 to the least significant bit. A test at condition code 723 indicates if the multiplier has completed the multiplication.

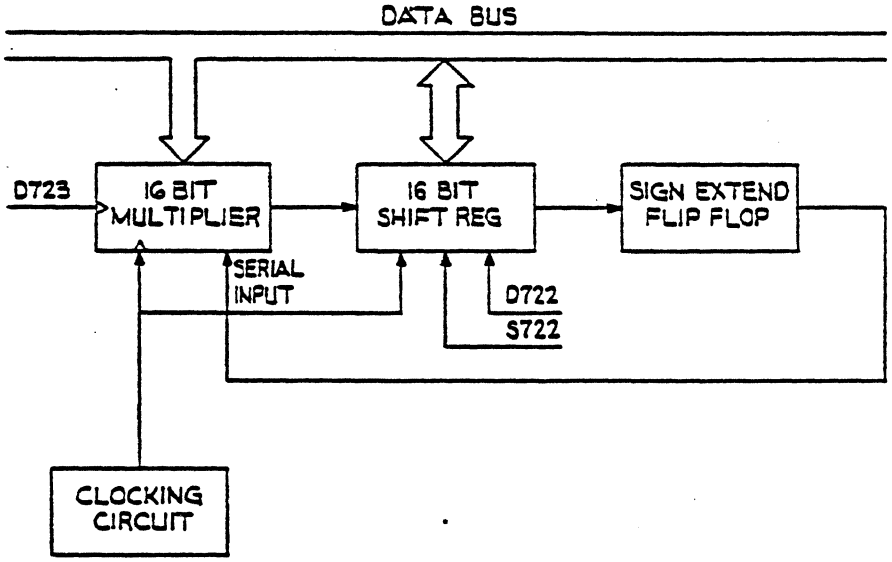


Figure 4-19. Fractional Multiplier Block Diagram

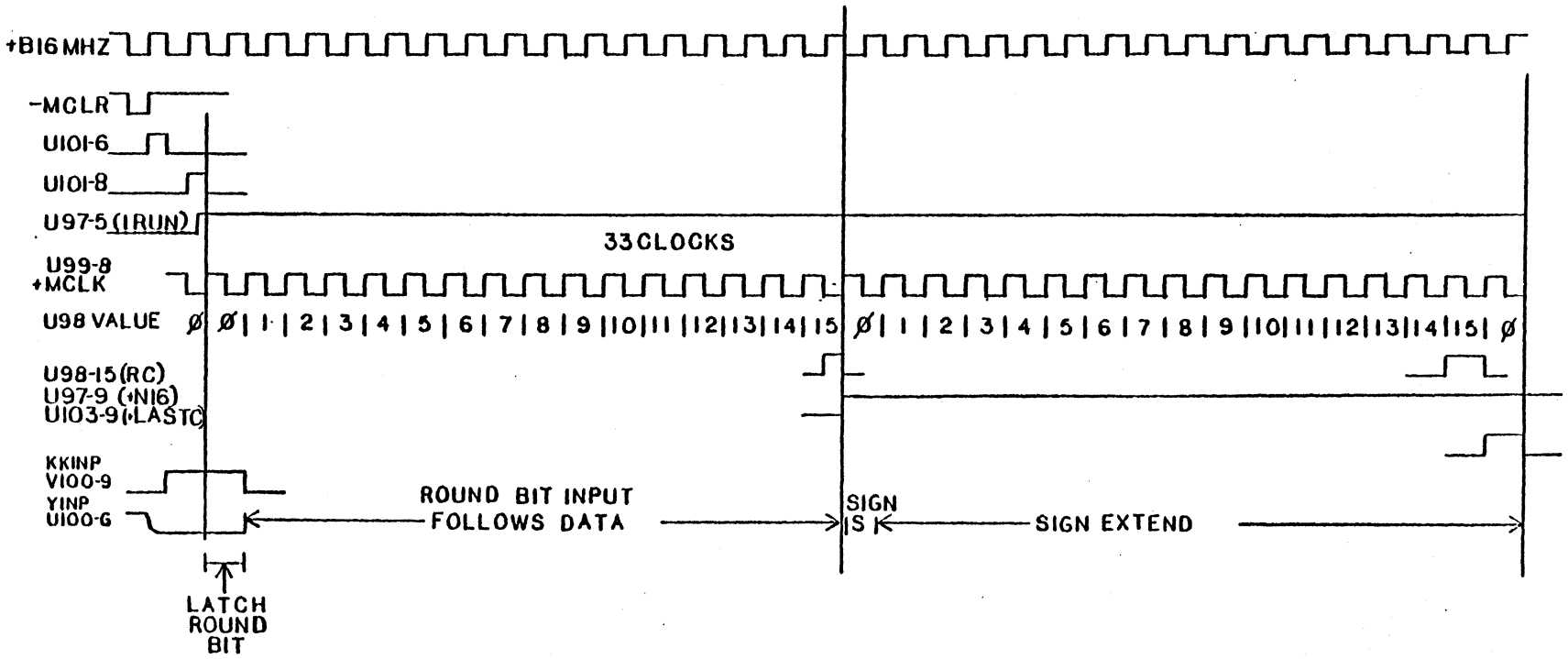


Figure 4-20. Multiply Timing Chart

4.11 MEDIUM RESOLUTION DISPLAY

The following text is supported by the diagrams listed below as well as illustrations within the text.

Simple Block Diagram, Medium Resolution Display	Figure 4-21
Functional Block Diagram, Medium Resolution Display Board	Figure 4-22
Schematic Diagram, Medium Resolution Display	2515-4706-2D

The Medium Resolution Display, hereinafter referred to as the display board, is a 6-wide board occupying slot 8 of the cardcage.

This single board graphics display controller (see figure 4-21) provides the interface between the 2501 microprocessor, the keyboard, the line printer, the vector generator and the associated display. The board includes a standard serial TTL keyboard interface which can be optionally converted to parallel by the addition of one integrated circuit. It also includes a Centronics compatible printer interface.

The display format is 640 horizontal by 480 vertical pixels. Individual pixels may be read or written by the 2501 microprocessor. On command, hardware scrolling provides the capability of moving the display up or down two or more scan lines.

The display memory comprises two frame buffers, separately accessed and displayed, to improve animated displays. An additional mode, SUPERMODE, allows superimposing both frame buffers on the CRT.

Video output is a standard composite video signal driving a 75 ohm line. Additional TTL level horizontal and vertical sync outputs are provided. The vertical rate, 50 or 60 Hz, is hardware selectable.

The major circuits to be discussed include the keyboard interface, printer interface, interrupt logic, source/destination register addresses, control, timing, vector generator, memory modifier, 64K RAM memory and control, and video output. Refer to figure 4-22.

4.11.1 Keyboard Interface

The keyboard interface is used to communicate between the microprocessor bus and a VT100 compatible keyboard.

The keyboard interface circuits consist primarily of a universal asynchronous receiver/transmitter (UART) U45, latches U89 and U110, and a timing circuit. The UART performs all the receiving and transmitting functions between the 2501 microprocessor and the keyboard. The UART is capable of receiving and transmitting serial and parallel data. In the 2515 system a serial keyboard is used, however the parallel output capability is also illustrated in the schematic on sheet 1.

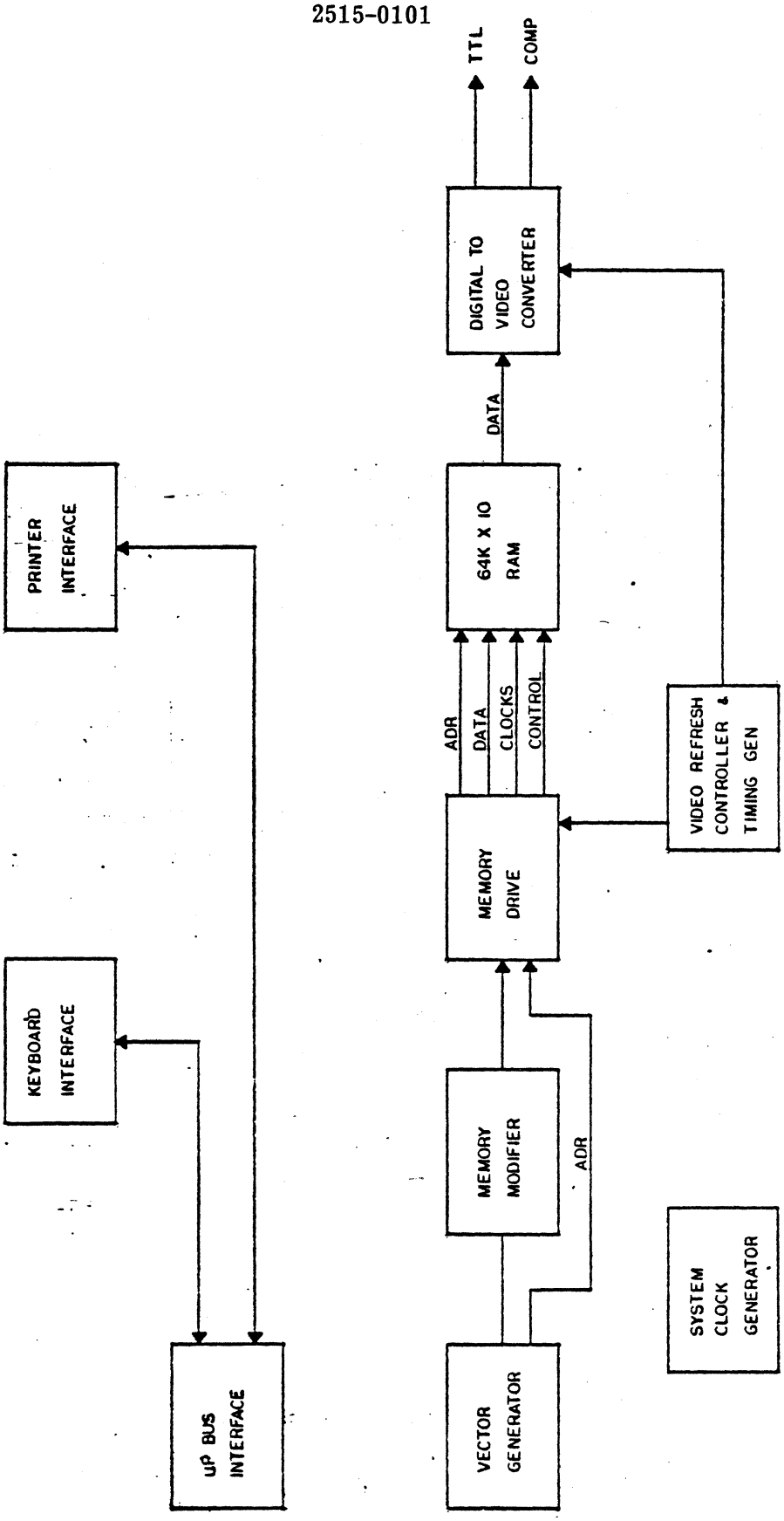


Figure 4-21. Simple Block Diagram, Medium Resolution Display

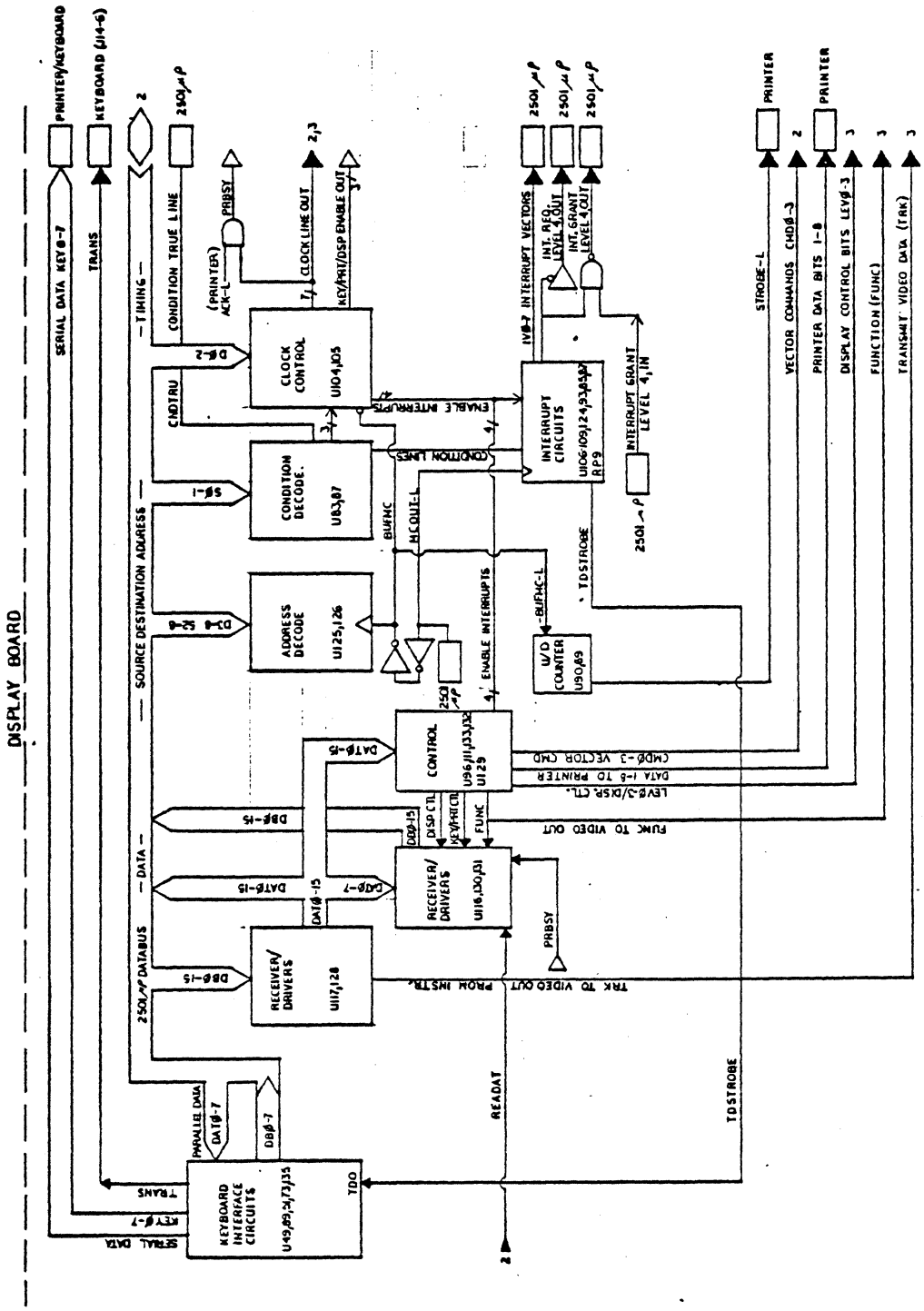


Figure 4-22. Functional Block Diagram, Medium Resolution Display (Sheet 1 of 3)

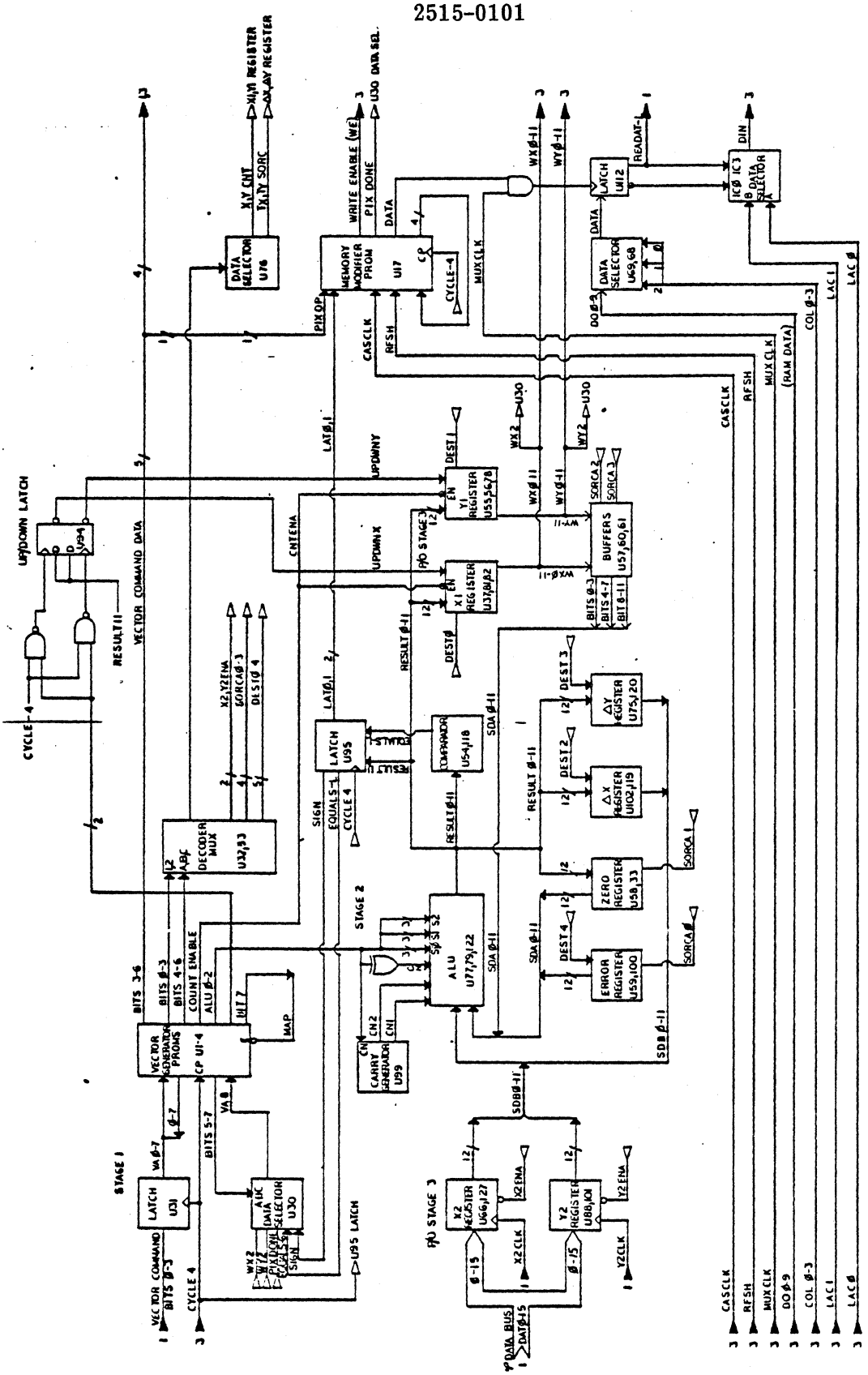


Figure 4-22. Functional Block Diagram, Medium Resolution Display (Sheet 2 of 3)

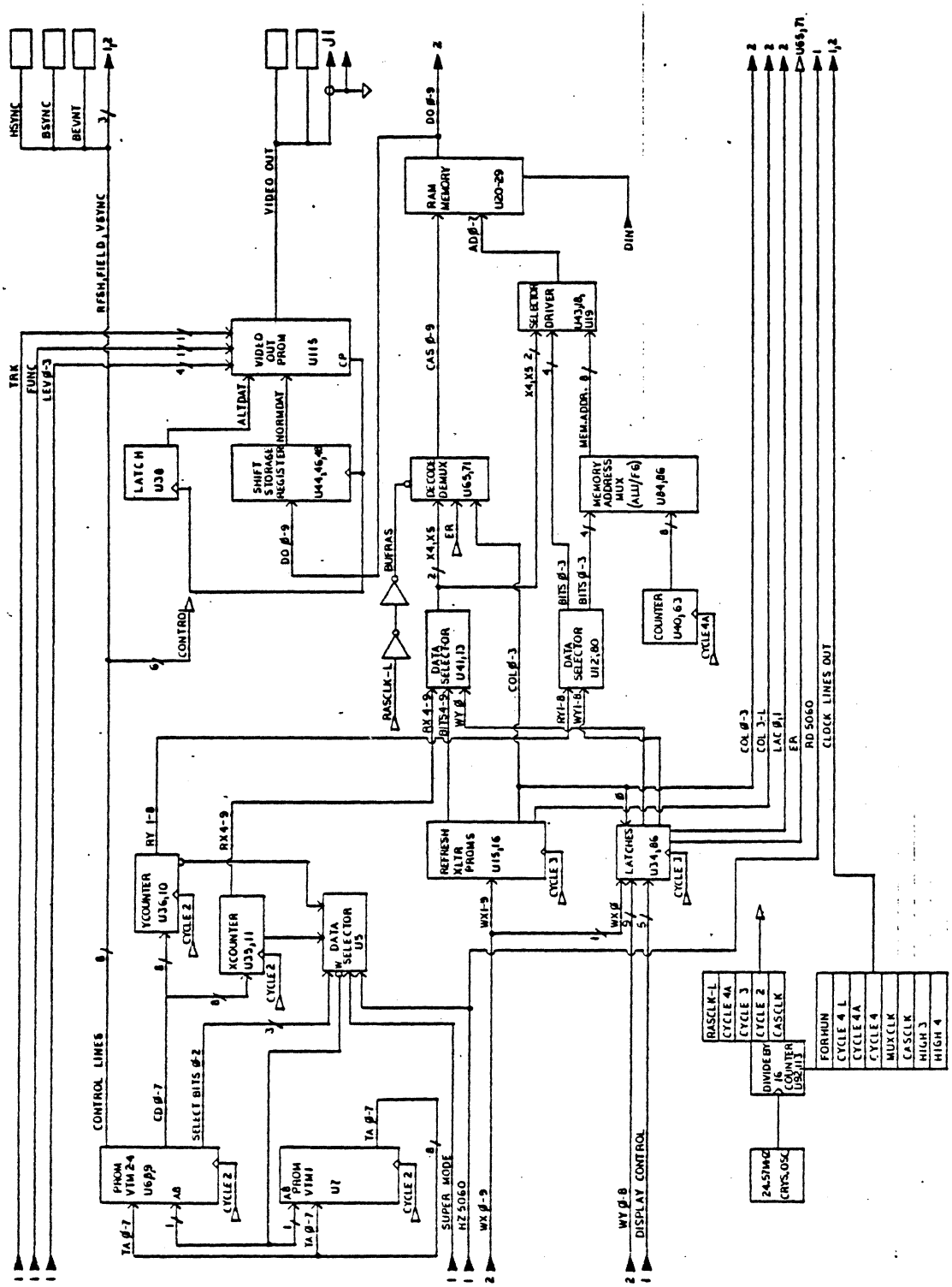


Figure 4-22. Functional Block Diagrams,
Medium Resolution Display
(Sheet 3 of 3)

The UART is reset during power up or by activating the BOOT/HALT switch. The FORHUN clock is disabled on reset. To do this, -BDCOK (the Q-bus DC okay signal) is gated and produces +BDCOK, a positive level. When -BDCOK goes low U130 line driver is cleared which inhibits the FORHUN clock. In this way, any characters being passed to the UART are latched into latch U89 thereby preventing character loss. The output of the FORHUN clock is applied to the UART. This action sets the serial output transmission line (TSO/TRANS) and buffer empty line (TBMT) to high and resets the data available (RDA) and over run (ROR) outputs to low. Serial bit input streams are accepted on the RSI input during the high to low transition of the clock. The data strobe (TDSTROBE) is pulsed and TBMT goes low. Transmission can now proceed. The data available (RDA) output goes high once the serial input has been loaded and clocked into the UART holding register. Also, KEY (RDA) is sent out to advise status (source base 0). KENA-L is returned to enable the output latch and resets RDA to a low level. The output, DB0-DB7, is sent to the 2501 microprocessor via the data bus. TBMT goes high and BMT goes out to indicate the empty status of the buffer.

Eight bits of parallel data, DAT0-DAT7, are returned from the 2501 data bus to the keyboard via the UART. The TBMT signal produced by the display board as a status bit to indicate whether the UART is ready to accept another character (base 0, bit 2). Data is first loaded into the holding register, then is transferred to a transmitter shifting register. Once the data is transferred TBMT goes high and the UART is ready to accept another character. Once the data strobe is pulsed TSO goes low as the start bit and transmission of serial output to the keyboard commences.

4.11.1.1 Clock Circuit. The base clock frequency is 24.57 MHz and is used to generate all the clocks used on the display board. The CYCLE4 clock, DOTCLK, and RASCLK are described in this subparagraph.

FORHUN is generated by dividing the base frequency by ten. It is then applied to a pair of binary counters and a flip-flop and that reduces FORHUN by 2**9 (or divided by 512). This new clock frequency is applied to the transmit and receive clocks of the UART which perform an internal divide by 16. This reduces the baud rate to 300 bits per second (or 10 characters per second).

The microprocessor data bits DATO through DAT7 are loaded into the destination address base 1 KEYDAT register and are then latched into U89 by KEYCLK and present on transmit register inputs. TDSTROBE and a negative pulse load data into the UART for transmission.

The data generated by the keyboard is received by the UART on RSI and converted to parallel data and present on RD1 through RD8. Once the key data is present on RD1 through RD8 the RDA (ready data available) signal goes high, loading data in U108. The UART's RDA signal is also used to tell the microprocessor that there is key data available through source register base 0, bit 7. The signal KENA-L enables the data onto the microprocessor bus.

CYLE4 Clocks

The base clock feeds U92 decade counter and also clocks a pair of flip-flops. The U92 counters output QA-DOTCLK base frequency divided by two which is used in the vector generator circuitry. This output is buffered by U64 and a pair of inverters (U62) creating two phases of the same clock: CYCLE4-L, CYCLE4A, and CYCLE4. The CYCLE4 clock is done this way so as not to exceed device fanout.

DOTCLK

The DOTCLK is generated by the QA output of the decade counter. It clocks four 8-bit shift registers that are used to take a 10-bit parallel work (D0 through D9) from RAM and shift it out serially. Two sections are controlled by LDRITE and LDLEFT. The QH serial output from U46 is used as the ALTDAT line on the video output PROM.

RASCLK

The QD output of U92 is the base frequency divided by eight and is used to read or write data into RAM. In addition, the RASCLK output feeds the D input of flip-flop U111 and is used to generate several clock cycles such as: MUXCLK, MUXCLK-L, CASCLK, CYCLE2, and CYCLE3. The Q output of U111 is wrapped around and feeds the D input of the second half of the same circuit. The first flip-flop's Q output generates MUXCLK and the Q-NOT output MUXCLK-L. The transitions of MUXCLK and MUXCLK-L are delayed by one base clock cycle. MUXCLK is connected to the D input of the second stage, which is delayed by two base clock cycles and this generates CYCLE3 and an inverted sense by U62 named CASCLK. The Q-NOT output is inverted by U62 and called CYCLE2.

4.11.2 Printer Interface

The printer interface allows the 2501 microprocessor to transmit 8 bits of parallel data and a strobe to the printer logic. Assuming that the printer switch is ONLINE, a SELECT signal is sent to the display board printer interface circuits to advise the printer is ready to receive data. If paper needs to be replaced in the printer a PAPEREND signal is sent to the source address register (base 0, bit 5) on the display board and data transmission is inhibited until the paper is replenished.

The SELECT signal is received and passed to the 2501 microprocessor to indicate select status. The printer destination address is decoded and the printer clock signal (PRCLK) is generated to preset PRBSY latch (BUSY) and to load the strobe and clock the data. An acknowledge (ACK-L) is sent from the printer to the printer interface to indicate that it is ready to receive another character. The ACK-L signal is approximately 5 microseconds. If an interrupt has been previously set, the positive transition of ACK also clocks the interrupt flip-flop and IR4 is generated (see also paragraph 4.11.3.1).

The printer interface has three status bits that determine the state of the printer via source address 710.

Address 710

Bit 4	PRINTER SELECT 1 = Printer is installed and operational. 0 = Printer off or not installed.
Bit 5	PAPER OUT 1 = Printer is out of paper. 0 = Printer has not run out of paper.
Bit 6	PRINTER BUSY 1 = Printer has not yet received the last data word transmitted. 0 = Printer is ready for new data.

Printer data is written into destination register U128 (address 716) and is clocked by PRCLK of U103. PRCLK is enabled by the source/destination address logic. This loads a present count into U90 which generates STROBE-L and presets the printer busy flip-flop U90 (thereby setting PRBSY high).

After data is transmitted the printer returns ACK to clock U130 and set the flip-flop into a ready state (low). When reading the source address 710, a tri-state buffer will be enabled to generate CNDTRU (condition true, printer ready) to the microprocessor.

If an interrupt has been previously set, the positive transition of ACK will also clock the interrupt flip-flop and generate -IR4.

4.11.3 Interrupt Logic

The interrupt hardware includes latch U109, priority encoder U108, buffer U124, vector select jumpers and gating circuitry (see sheet 3 of the schematic).

The display board can generate an interrupt from any of four events; printer ready, key available, vertical sync, or vector generator command done. Each of these events represents the transition of one of the status lines which can be read in a source register. When acknowledged, the interrupting board will send an interrupt handler pointer (or vector) on the appropriate lines. Each of the four interrupting events corresponds to a unique vector within a group of four sequential values. Similar to the way source and destination register addressing is decoded, interrupt vectors are formed by adding the number associated with the event to a base value which is selected with wire-wrap jumpers. Standard values for these interrupt vectors have been assigned as shown in the table. All interrupts generated by this board occur at level 4. Once the interrupt vector is put on the bus, the two least significant bits of the vector are used to clear the corresponding interrupt latch.

Table 4-10. Interrupt Vectors

Vector	Vector Address	Source Register Signal Reference	Interrupt Enables	Event
IV Base +0	214	Base + 0, bit 6	0	Printer changes from busy to ready
IV Base +1	215	Base + 0, bit 7	1	New keyboard data is available
IV Base +2	216	Base + 2, bit 13	1	CRT refresh cycle has finished a complete field (odd or even)
IV Base +3	217	Base + 2, bit 15	0	Vector generator completes a command in process

The generated vector addresses (214 through 217) depend on the function of the interrupt. When -IR4 is asserted low and IG4I is high the output of U93 goes low. This low output latched into UART U108. This enables the vector address onto the microprocessor bus and gates decoder U83 to clear the interrupt request flip-flop.

The interrupts are software enabled by the two least significant bits, 0 and 1, of source/destination control registers KPCSW (base +0) and DSPCSW (base +2).

Enabling interrupts is done by writing data to the appropriate destination register, 710 or 712, bits 0 or 1. Data is then loaded by CSWCLK which generated by the source and destination address control logic. This latches the interrupt enable data. This data can be read back through the appropriate source address register. The latched data bits are passed to the D input of the flip-flop which is then clocked by the corresponding event. For example: EICMD enable interrupt on command done is set and is clocked by the positive transition of CMDONE-L.

The outputs of the interrupt flip-flops feed U108 and are latched on the positive transition of MCOUT-L. In turn, the four bits of U108 interrupt data is connected to the inputs of a priority encoder. The GS output asserts -IR4 interrupt request level 4. A0 and A1 are used as select lines to U83 to allow the interrupt flip-flops to be cleared and is also used to determine the vector address.

4.11.3.1 Printer Interrupt. When the printer is ready to receive a data word U134 flip-flop gates PRBSY (bit 6) low to control register U111. The control register transmits EIPR, printer interrupt enable, high to the interrupt logic. The data is clocked by ACK-L and the output latched by U109.

4.11.3.2 Keyboard Interrupt. If a key is struck on the keyboard KEY is sent high to control register U130 and the interrupt logic. EIKEY is also sent to the interrupt logic from control register U111 (base +0). The data is clocked by KEY and the output latched by U109.

4.11.3.3 Vertical Sync Interrupt. At the beginning of a vertical synchronization period (interlace, even or odd field) the display control register enables a vertical sync interrupt (base +2, bit 1) and EIVSYNC goes high to the interrupt logic. The video timing circuit sends VSYNC to clock the data which is subsequently latched by U109.

4.11.3.4 Vector Generator Interrupt. The vector generator interrupts when it completes a command and is ready to receive another and bit 15 of source address base +2 goes low. The display control register enables the command done interrupt and EICMD is sent high as data to the interrupt logic. CMDONE-L is then sent from the vector command register U129 to clock the data and the output is then latched by U109.

Once an interrupt has been latched by U109, the output is passed to U108 priority encoder. The encoder uses the data to produce three outputs.

One output generates a buffered level 4 interrupt request (-IR4) low to the 2501 microprocessor. The same output is NANDed with the interrupt grant (-IG4I) and returned. The output is latched to enable U83 and reset appropriate interrupt latches. The other two outputs are used to select lines to decoder U83 and to provide two vector select data bits to buffer U124. The interrupt vector, IV0-IV7, is sent to the 2501 microprocessor to specify the address of the interrupt. (If the interrupt grant is not to be used, it is passed out and off the display board as -IG4O onto the next circuit board assembly in the system.)

Note that interrupt requests which are outstanding when the corresponding interupt enable control bit is changed from a logic 1 to a logic 0 are withdrawn. Interrupt causing events which occur when interrupts are disabled do not cause an interrupt when interrupts are subsequently enabled.

4.11.3.5 Interrupt Grant Signal. The signal IG4I feeds U85 and if an interrupt request has not been asserted (low) the interrupt grant signal is passed to the next board on the bus via IG4O.

4.11.4 Address Decode

The address decode circuitry (see sheet 3) includes two 8-bit comparators U125 and U126, latch U104, two decoders U83 and U105, flip-flop U134 and bus buffer gate U87.

Eighteen source and destination lines are sent from the 2501 microprocessor via the address bus, thirteen of which are applied directly to the 8-bit comparators. The source lines are S0-S8. The destination lines are D0-D8.

The microprocessor source and destination addresses are used to access the display command and data registers. Addresses are decoded through two 8-bit comparators U120 and U121. The address select jumpers are set on SD2 through SD8 and the current address is 710.

The six most significant bits of destination addresses, D3 through D8, are connected to 8-bit comparator, U120. The comparator's equals output and the three least significant bits, D0 through D2, are latched into U102. Outputs are applied to the select and gate lines of U103 to load all the destination registers.

Source addresses S2 through S8 connect to U121, with the equals out. S0 and S1 are used as gate and select lines to U88. Gating U88 generates two functions. It asserts the CNDTRU bit to the microprocessor bus and enables reading the source registers or it just generates CNDTRU to the microprocessor bus.

4.11.5 Source/Destination Registers

Three source and seven destination registers provide the address status, and control for the display board. These registers are basically software addresses (base) plus the register number and the active register bit (or bits; see example). Functional descriptions follow for each of the registers.

Address Example: 710 0 (0-7)

-Active register bit
-Register number
-Source base address

4.11.5.1 Source Address Registers. The three source address registers used are KPCSW (address base 0), KEYDAT (address base +1), and DSPCSW (address base +2). Base +0 register address is 710, base +1 is 711, and base +2 is 712.

a. Keyboard/Printer Status, KPCSW Register, **Base +0.**

This register uses the eight least significant bits (0-7) of a 16 bit word. The most significant bits (8-15) are not used.

15 - 8	7	6	5	4	3	2	1	0
— Not Used —	key avail	printer busy	paper out	printer select	key error	key send ready	key intrpt enable	print intrpt enable

Bit 0 Printer interrupt enable (EIPR)

1 = Interrupts from the printer are enabled (see destination base +0, bit 0).
0 = Printer interrupts are disabled.

Bit 1 Keyboard interrupt enable (EIKEY)

1 = Interrupts from the keyboard are enabled (see destination base +0, bit 1).
0 = Keyboard interrupts are disabled.

Bit 2 Keyboard send ready (BMT)

1 = The keyboard transmit buffer is empty and can accept another data word (see destination address base +6, bits 0-7). Keyboard interface is double buffered.
0 = The keyboard transmit buffer is full and cannot accept more data. Sending data while the buffer is full results in loss of one or more words of transmitted data.
Power-on = 1.

Bit 3 Keyboard receiver overrun error (ROR)

1 = The keyboard receiver buffer was full when another character was received. Indicates that one or more characters have already been lost. Error should be avoided by promptly reading the keyboard receive register (source address base +1). (Characters arrive at a time-averaged maximum rate of 30 characters per second.) If overrun error has occured, reading the keyboard receive register resets this bit to 0.
0 = No keyboard overrun error has occured since the last read of the keyboard receive register (source address base +1).

Bit 4 Printer selected status (SELECT)

1 = Local "printer on-line" switch is activated on printers which have this feature. With printers lacking this feature, this bit is permanently set to 1.
0 = Printer has been deselected locally.

Bit 5 Paper out (PAPEREND)

1 = Printer has run out of paper. Some printers may also automatically de-select when this occurs (see source address base +0, bit 4). If the printer does not implement this status line, it is permanently set to 1.
0 = Printer has not run out of paper.

Bit 6 Printer busy (PRBSY)

1 = Printer has not yet received and acknowledged the last data word sent to it. Sending a new data word using destination register address base +1 will result in an error and loss of data.
0 = Printer is ready to receive new data.
Power-on = 0.

Bit 7 Key available (KEY)

1 = Data from the keyboard is available in source register address base +1.
0 = The keyboard receive buffer is empty.
Power-on = 0.
Reading source register base +1 resets this bit to 0.

b. Keyboard Data, KEYDAT Register, **Base +1.**

This register uses the eight least significant bits (0-7) of a 16-bit word. The most significant bits (8-15) are not used.

15 - 8	7	6	5	4	3	2	1	0
— Not Used —	RK7	RK6	RK5	RK4	RK3	RK2	RK1	RK0

Bits 0-7 Keyboard receive data (REC)

When the "key available" status bit is set (source address base +0, bit 7), this register contains a data word received from the keyboard. The keyboard interface is double buffered; however, when the buffers are full and another character is received before this register is unloaded, a keyboard overrun error results (see source address base +0, bit 3).

c. Display Status, DSPCSW Register, **Base +2.**

This register uses all 16 bits of the address word.

15	14	13	12	11	10	9	8
cmd not done	blank	Field	LEV 3	LEV 1	LEV 1	LEV 0	two tone

7	6	5	4	3	2	1	0
super mode	Pixel data	Actual 50/60 Hz	Un-blank	Op/ same	View plane	vsync intrpt enable	cmd done intrpt enable

Bit 0 Command done interrupt enabled (EICMD)

1 = The vector generator will interrupt when it completes a command except NOP.
0 = The vector generator will not interrupt.
Power on = 0.

Bit 1 Vsync interrupt enable (EIVSYNC)

1 = An interrupt request will occur at the beginning of each vertical sync period.
0 = Vertical sync interrupts are disabled.
Power on = 0.

Bit 2 View plane select

Indicates which image plane has been selected for viewing, see destination base +2, bit 2.
1 = Plane 1 is displayed.
0 = Plane 0 is displayed.

Bit 3 Access opposite or same plane

Indicates which image plane has been selected for access by the microprocessor; see destination base +2, bit 3.
1 = Indicates plane which is not visible.
0 = Indicates plane which is refreshing CRT during the operation.

Bit 4 UNBLANK

Indicates the state of the blanking control bit (destination base +2, bit 40.
0 = Indicates the blank state
1 = Indicates the normal viewing mode.

Bit 5 Actual 50/60 Hz rate selected (RD5060)

Indicates the vertical field refresh rate selected by the combination of the hardware jumper and the 50/60 Hz control bit (destination base +2, bit 5).
1 = 50 Hz (European)
0 = 60 Hz (American)

Bit 6 PIXEL data

Contains the data read during the last image memory pixel operation. At the completion of a "pixel read" command to the vector generator, this bit is the result.

Bit 7 SUPERMODE

Indicates the state of the supermode select bit, see destination base +2 bit 7.
1 = Two image planes visible.
0 = One image planes visible.

Bit 8 TWOTONE

Indicates the state of the twotone select bit, see destination base +2, bit 8.
1 = Dual intensity mode has been enabled.
0 = Dual intensity mode has ben disabled.

Bits 9-12 Intensity level

Indicates the state of the intensity control bits, see destination base +2, bits 9-12.

Bit 13 Field

Indicates which field is currently being refreshed on the CRT, odd or even.
1 = Odd
0 = Even

Bit 14 BLANK

Indicates the current state of the blanking signal (note: this is not the same as "unblank", destination base +2, bit 4).
1 = Blanked
0 = Not blanked

Bit 15 Vector generator command not done

Indicates the status of the vector generator.
1 = Vector generator is busy executing a command, do not change parameters or commands.
0 = Vector generator is idle and ready to receive and begin executing a new command.

4.11.5.2 Destination Address Registers. The seven destination address registers used are KPCSW (address base +0), KEYDAT (address base +1), DSPCSW (address base +2), VECCMD (address base +3), VECX (address base +4), VECY (address base +5), and PRDAT (address base +6).

a. Keyboard/Printer Control, KPCSW Register, **Base +0.**

Only bits 0 and 1 of this register are used for destination control.

15 - 2	1	0
— Not Used —	enable intrpt on key	enable intrpt on print

Bit 0 Enable interrupt from printer (EIPR)

1 = Printer will interrupt when "printer busy" (source base +0, bit 6) goes low indicating printer has received one word and is ready for another.
0 = Printer can not interrupt.
Power-on = 0

Bit 1 Enable interrupt from keyboard (EIKEY)

1 = Keyboard will interrupt when "key available" (source base +0, bit 7) goes high indicating a key has been struck and its value is available to be read from source register base +1.
0 = Keyboard can not interrupt.
Power-on = 0

b. Keyboard Data, KEYDAT Register, **Base 1.**

Only the eight least significant bits of this register are used.

15 - 8	7	6	5	4	3	2	1	0
— Not Used —	TK7	TK6	TK5	TK4	TK3	TK2	TK1	TK0

Bits 0-7 Transmit data to keyboard (TRK). Eight bits of data which will be automatically sent to the serial keyboard. Before writing to this register, test "keyboard send ready" (source base +0, bit 2) to be sure the interface is ready to receive new data.

CAUTION

After loading this register, the microprocessor must allow at least one bus cycle to pass before attempting to reload the register. This normally would be accomplished automatically by testing the ready status bit before every keyboard load operation.

c. Display Control, DSPCSW Register, **Base 2.**

This command register can be accessed by writing data into the destination register base (address 712). The register feeds hardware responsible for screen setups and manipulation.

15	14	13	12	11	10	9	8
— Not Used —			LEV 3	LEV 2	LEV 1	LEV 0	two tone

7	6	5	4	3	2	1	0
super mode	Not Used	50/60 Hz	un-blank	access OP/ same	view plane	enable intrpt on V sync	enable intrpt on CMD done

Bit 0 Enable interrupt on command done (EICMD)

1 = The vector generator will interrupt when "command not done" (source base +2, bit 15) goes low, indicating that execution of the last command is complete and the vector generator can accept a new command.
0 = The vector generator can not interrupt.
Power-on = 0

Bit 1 Enable interrupt on vertical sync (EIVSYNC)

1 = Video controller will interrupt at the start of the vertical sync period when "vsync" (source base +2, bit 13) goes high.
0 = Video controller can not interrupt.
Power-on = 0

Bit 2 Select viewed plane (PLANE)

Two separate image buffers exist but only one can normally be viewed on the CRT at a time. This control bit selects the plane which is used to refresh the CRT image so it is visible.
1 = Plane 1 is displayed.
0 = Plane 0 is displayed.

Bit 3 Access plane (SAMEOP)

The microprocessor can read, write and draw vectors in either plane. This bit, in conjunction with "select view plane" selects which of the planes to be accessed by subsequent image plane operations (reading, writing).
1 = Read and write operations affect the plane which is not visible during the operation.
0 = Read and write operations access the plane which is refreshing the CRT during the operation.
Power-on = 0

Bit 4 UNBLANK

0 = Forces the video signal to black level, blanking the CRT screen. The image buffer memories are unaffected.
1 = Normal viewing mode, the current viewable plane is visible on the CRT.

Bit 5 50/60 Hz (RD5060)

Selects 50 Hz or 60 Hz vertical sync rate for European or American standard television. In 50 Hz mode, additional blank lines are inserted above and below the visible picture to slow the field refresh rate to 50 Hz. There is provision for a wire jumper on the circuit board to select the 50/60 Hz mode (inserted = 60 Hz). This software controlled line can be used to override the jumper selection.
0 = Mode is selected by jumper on board.
1 = Mode is the opposite of that selected by the jumper.
Power-on = 0

Bit 6 Not used

Bit 7 SUPERMODE (Superimpose mode)

1 = Causes both image planes to be superimposed on the CRT screen. Does not affect the function of "viewplane" (base 2, bit 2) or "access op/same" (base 2, bit 3), except that the plane which is normally not visible is now visible also. The superimposed mode slows down pixel memory accesses; this is transparent to the programmer, but drawing images takes longer.
0 = Normal mode, only one plane is visible.
Power-on = 0

Bit 8 TWOTONE (Dual intensity mode)

1 = Enable dual intensity mode. When SUPERMODE (base +2, bit 7) is enabled (making both image planes visible), setting twotone to 1 causes the two planes to be shown at different intensities. The normal plane (selected by base +2, bit 2 "select viewed plane") appears at normal brightness; the opposite plane appears at half intensity. The pixel intensity levels of the two planes are added to yield the displayed image. When SUPERMODE is disabled, Twotone has no effect.
0 = Disable dual intensity mode. When SUPERMODE is enabled, the two planes are OR'ed together and displayed at one intensity level. When SUPERMODE is disabled, TWOTONE has no effect.
Power on = 0

Bits 9-12 Intensity level (LEV0-LEV3)

Treated as a two's complement number, (with base +2, bit 12 as the sign, base +2, bit 11 as the most significant bit, and base +2, bit 9 as the least significant bit). These four bits give software control to increase (positive values) or to decrease (negative values) the CRT display contrast. This is accomplished by adjusting the voltage level

of the visible image portion of the video output signal. In typical use, these bits would be set to 0 and the contrast and brightness controls on the CRT monitor would be adjusted for a good picture under normal conditions. Then, these intensity level bits can be used to raise or lower the contrast to suit the prevailing conditions or operator preference, without needing to change the CRT settings. The range of adjustment in typical use is from barely visible (bit 12 set, bits 9-11 cleared) to the point where the CRT saturates (bit 12 cleared, bits 9-11 set).
Power-on = 0

Bits 13-15 Not Used

d. Vector Command, VECCMD Register, **Base +3.**

15 - 8	7	6	5	4	3	2	1	0
— Not Used —	Res'vd	OPRN 1	OPRN 0	Not Used	CMD3	CMD2	CMD1	CMD0

Bits 0-3 Vector generator command (CMD0-CMD3)

These four bits select one of eleven operations to be performed by the vector generator. If the vector generator is ready for a new command, writing to destination register base +3 initiates execution of the operation selected. Parameters X and Y must be loaded prior to issuing a command (see destination base +4). Operation codes are:

0 No operation
1 Move to position x, y(beamp)
2 Draw vector to x, y
3 Not assigned
4 Read/write pixel
5 Fast erase from current position to x, y
6 Fast erase from 0, 0 to x, y
7 Move-relative Δx, Δy
8 Fine erase from current position to xy, y
9 Scroll up two scan lines
10 Scroll down two scan lines
11 Scroll up 2 y scan lines
12 Not assigned
13 Not assigned
14 Not assigned
15 Not assigned

Power-on = 0
More detailed information on the vector generator is presented in paragraph 4.11.7.

Bit 4 Not used

Bit 5-6 Pixel Data (OPRN0, OPRN1)

These two bits control what data will be written to the image memory during vector generator pixel operations. The codes are:

Bit 6	Bit 5	
0	0	Read only
0	1	Clear (write 0)
1	0	Set (write 1)
1	1	Complement (invert current value)

Power on = 0

These pixel data bits will be reset to a logic 0 upon completion of a vector generator operation.

CAUTION

Do not change these bits while the vector generator is busy.

Bit 7 This bit cannot be written under software control.

Bits 8-15 Not used.

e. Vector X Coordinate, VECX Register, **Base +4.**

15 - 12	11	10	9	8	7	6	5	4	3	2	1	0
- Not Used -	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

Bits 0-11 X coordinate.

This parameter is used by the vector generator in executing its next command. Bits 0-11 equal X0-X11, two's complement binary, normally in the range of 0 to +639 (decimal).

CAUTION

Do not change these bits while the vector generator is busy.

Bits 12-15 Not used.

f. Vector Y Coordinate, VECY Register, **Base +5.**

15 - 12	11	10	9	8	7	6	5	4	3	2	1	0
- Not Used -	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Bits 0-11 Y coordinate.

This parameter is used by the vector generator in executing its next command. Bits 0-11 equal Y0-Y11, two's complement binary, normally in the range of 0 to +511 (decimal). Only values 0 to +479 are visible.

CAUTION

Do not change these bits while the vector generator is busy.

Bits 12-15 Not used.

g. Printer Data, PRDAT Register, **Base +6.**

15 - 8	7	6	5	4	3	2	1	0
— Not Used —	P7	P6	P5	P4	P3	P2	P1	P0

Bits 0-7 Printer data.

These eight bits of data, when loaded, are sent to the printer. Writing to this register also sets the printer busy bit, PRBSY (source address base +0, bit 6).

Bits 8-15 Not used.

4.11.6 Condition - True

The condition line is tested by reading one of three source registers: base +0, base +2, or base +3. This sets the bus CNDTRU line (see sheet 3 of schematic) to the state of the function selected according to the address used. The 2501 microprocessor can test either logic state (1 or 0) of the CNDTRU line. Table 4-11 presents line and mnemonic assignments.

Normal use of this line by the microprocessor is as the conditional input for a jump or a return instruction. The three functions on the display board which are tested in this manner are also available in the status registers, therefore the table refers to the appropriate signal in the source register descriptions in paragraph 4.11.5.1.

Table 4-11. Condition Line Assignments

Address	Condition	Source Register
BASE + 0	PRINTER BUSY	Base + 0, bit 6
BASE + 3	KEY AVAIL	Base + 0, bit 7
BASE + 2	CMD NOT DONE	Base + 2, bit 15

BASE = 710

Address	Mnemonic	Function
710	PRRDY	Jump if printer is ready to receive another character
710	PRBSY	Jump if printer is busy
713	NOKEY	Jump if no data is available from the keyboard
713	KEY	Jump if data is available from the keyboard
712	VGDONE	Jump if vector generator is done
712	VGBUSY	Jump if vector generator is busy

4.11.7 Vector Generator

The vector generator serves as the interface between the 2501 microprocessor and the image buffer memory, the latter of which can be examined or modified only through the vector generator. (See vector generator stages 1-3 on schematic sheets 4-6).

Parameters from other registers and the command register are used in the execution of vector generator commands. Specifically, destination register base +4 (VECX) is used as an X coordinate value, destination register base +5 (VECY) is used as a Y coordinate value, and bits 5 and 6 of destination register base +3 (VECCMD) describe how the memory is to be modified (read only, set, reset, complement). In addition, the vector generator maintains an internal register called the "current position". This register holds an X, Y coordinate pair, usually the last point referenced by the vector generator. The effect of each operation on this register is explained in the description of the operations.

The vector generator program resides in four 512-word x 8-bit PROMs. Three of the PROMs (U1, U2, and U3) are used in the hardware manipulations. The fourth PROM (U4) is the address pointer. Bits CMD0-CMD3 provide the starting address for the vector generator PROMs. The first 16 addresses in PROM U2 contain the data address jump table, i.e., pointers to the microcode routines for the vector generator functions.

The vector generator commences an operation when a vector command is loaded on bits CMD0-CMD3 and is then applied to the address bits of all the vector generator PROMs. If, for example, PROM U4 outputs a map enable bit for U2, U2 then outputs the next address onto the PROM lines sequentially until the command sequence is completed. Then the PROMs jump to an idling address to wait until the next command.

Detailed descriptions of all the available commands follow in paragraph 4.11.7.1.

4.11.7.1 Vector Generator Commands. The vector command register uses four bits, CMD0-CMD3, which are used as the starting address for the vector generator PROMs. All the instructions used in vector operation perform four types of functions: scrolling up or down, line drawing, read/write pixels, and erasing.

Bits 0-3 of the vector command register (U128) are written into destination address (713), register U129. The destination address data is latched by the positive edge of SCROLLCLK. Once latched, the data applied to U31 flip-flop is loaded on the positive edge of the CYCLE4 clock.

All available commands are listed below followed by a description of each command.

Command Number	Command Name	Parameters Used	Function
0	NOP	none	No operation
1	MOVE	x, y	Move current position (CP)
2	VECTOR	x, y, data, CP	Draw vector from current position to x, y
3	-	-	Not assigned
4	RWPIX	x, y, data	Read or write pixel at x, y
5	ERASBLK	x, y, data, CP	Fast erase from current position to x, y
6	ERASEXY	x, y, data	Fast erase from 0, 0 to x, y
7	RELMOV	x, y, CP	Move current position by Δ x, Δ y
8	FINERA	x, y, data, CP	Precise erase from current position to x, y
9	SCROLLUP	none	Scroll up 2 scan lines
10	SCROLLDWN	none	Scroll down 2 scan lines
11	SCROLLN	y	Scroll up 2*(y+1) scan lines
12-15	-	-	Not assigned

Following are detailed descriptions for each available vector command operation.

Command	Description
0, NOP	Idle state; perform no operation.
1, MOVE	Move current position point to X, Y. This loads internal current position register with the contents of the X and Y destination registers (base +4 and base +5). It does not modify any image memory locations, and does not change contents of X and Y destination registers.
2, VECTOR	Draw vector from current position to X, Y. This command calculates the appropriate pixel locations near a line between the current position and the contents of X and Y (destination base +4 and base +5) and modifies them according to DATA (destination base +3, bits 5 and 6). Data may instruct vector to modify memory by writing ones, writing zeros, complementing the current state of each pixel, or not to modify memory (read). Upon completion, the current position register is set to X, Y and the X and Y registers will be unchanged; data will be reset to read. Pixels modified by the vector command include the current position and the ending point, which may be especially significant for complement vectors. Vectors are drawn from the current position to the ending point, regardless of direction. When redrawing a vector (drawing a vector using the same end points as a previously drawn vector), the vector command will modify exactly the same pixels it modified when the vector was first drawn only if the vector is drawn in the same direction as it was previously. Thus, a vector may be erased by drawing its complement only if the same current position and ending point are used. Vectors are calculated using 12 bit, two's complement signed numbers; however, only positive numbers between 0 and 639 decimal for X, and between 0 and 479 for Y are allowed for the display. Drawing vectors outside these limits may result in unexpected wraparound images.
3	Not assigned, do not use.

4, RWPIX — Read or write pixel at X, Y.

Using the memory modification instructions in data (base +3), bits 5 and 6), the pixel at location X, Y will be set, reset, complemented or simply read. When the pixel is read or complemented, its state (or former state, in the case of complementing) is available as bit 6 of source register base +2 at the completion of this command. The contents of registers X and Y are unchanged; the current position is changed to equal X, Y.

5, ERASBLK — Fast erase from current position to X, Y.

This command modifies (erases) a rectangular region of the image in memory according to data (base +3, bits 5 an 6). Because of the way this operation is performed, only the set and reset (write 1 or 0) modes of data are rational; complement and read-only modes have hard-to-predict results. Executing this command, the vector generator modifies ten pixels simultaneously during each memory operation. Each group of ten consists of a pixel with an arbitrary Y coordinate and an X coordinate which is 0 or a multiple of 10, and the nine pixels immediately to its right on the CRT screen. The location and size of the rectangle to be erased are defined by the internal current position register and the X and Y destination registers (base +4 and base +5). The axes of the rectangle are parallel to the X and Y axes. The lower left corner of the rectangle is located at 10* ((current position x) modify 10), (current position Y). The upper right corner is at ((destination register X) destination register Y).

CAUTION

The current position X and Y values must be less than the respective destination register X and Y values. In addition, Y coordinates must be in the range 0 to 511 decimal, and the only allowable destination X coordinates are (10N+9), where N = 0 to 63. In other words, the upper right hand coordinate of the erased rectangle must be accurately specified. Failure to meet these restrictions may produce unpredictable or undesirable results.

Destination registers X and Y are unchanged; data is cleared and the current position register is modified to 10* ((destination register X) modify 10), (destination register Y). The ERASBLK command will erase a region of the screen faster than writing all the pixels in that area individually.

6, ERASEXY — Fast erase from 0, 0 to X, Y.

This command operates identically to command 5, ERASBLK except that the current position is automatically reset to 0 at the start. Refer to the description of ERASBLK for details of operaton. The fastest way to erase the entire visible image is to execute the ERASEXY command with X = 630 decimal and Y = 470 decimal.

7. RELMOV — Move current position ΔX and ΔY.

Adds the signed value in destination registers X (base +4) and Y (base +5) to the current position register X and Y respectively. The contents of destination registers X and Y are unchanged.

8, FINERA — Precise erase from current position to X, Y.

This command modifies (erases) a rectangular region of the image in memory according to data (base +3, bits 5 and 6). Because of the way this operation is performed, only the set, reset, and complement modes are meaningful. The location and size of the rectangle to be erased are defined by the internal current position register and the X and Y destination registers (base +4 and base +5). The axes of the rectangle are parallel to the coordinate axes. The lower left corner of the rectangle is located at the current position; the upper right corner is at destination registers X, Y. The current position X and Y must be less than or equal to destination X and Y respectively. In addition, X coordinates must lie between 0 and 639; Y must be between 0 and 511 decimal. Destination registers X and Y are unchanged; data is cleared and the current position is set to destination registers X, Y. This command allows erasing a precisely located, precisely sized rectangle but is much slower than ERASBLK and ERASEXY commands.

9, SCROLLUP — Scroll screen up two scan lines.

This command rotates the screen image buffer so that the picture moves up two scan lines. The top two lines move into the bottom of an invisible 32 line additional buffer region. The top two lines of this invisible region wrap around and appear as the bottom two visible lines. The scroll operation affects all image planes, viewable or not, read/write accessible or not. The scroll operation moves the current image, not the coordinate axes. The current position register and the X and Y registers are unaffected. Data is cleared. Notice that the scrolling operation does not change the relation between the current position and the edges of the CRT image, but changes the relation between the current position and the lines previously drawn on the screen.

10, SCROLLDWN Scroll screen down two scan lines.

This command rotates the screen image buffer so that the picture moves down two scan lines. The bottom two lines move into the top of an invisible 32 line buffer. The bottom two lines of the invisible region wrap around and appear as the top two visible lines. The scroll operation affects all image planes, viewable or not, read/write accessible or not. The scroll operation moves the current image, not the coordinate axes. Note that the scrolling operation does not change the relation between the current position and the edges of the CRT image, but changes the relation between the current position and lines previously drawn on the screen. The current position register and the X and Y registers are unaffected. Data is cleared.

11, SCROLLN Scroll up two times Y lines.

This command rotates the screen image buffer so that the picture moves up by a number of scan lines determined by the value in destination register Y (destination base +5). The number of scan lines moved is equal to twice the number in register Y. The top 2*Y lines move into the bottom of an invisible 32 line buffer. The top 2*Y lines of this buffer wrap around and appear as the bottom 2*Y visible lines. The scroll operation affects all image planes, viewable or not, read/write accessible or not. The scroll operation moves the current image, not the coordinate axes. The scrolling operation will not change the relation between the current position register and the edges of the CRT screen, but changes the relation between the current position and lines previously drawn. The X and Y registers are unchanged. The current position register is modified. Data is cleared. This scroll up command can also be used to scroll down by using a number for Y which rotates the image slightly less than a full screen. To move down 2*N lines, Y = 256 -N.

12-15 Not assigned; do not use.

4.11.7.2 Display Memory. The display memory is comprised of ten 64K x 1 RAMs. The vector commands are written into memory and then transferred to the screen. The screen displays are generated by a series of 10-bit (pixel) words. Only one RAM is written to or read from at any time. All others are being refreshed as RASCLK-L is cycled.

Addresses are multiplexed for row and column addresses. Refreshing the RAM is done during the cycle when RASCLK-L is low and CAS 0-9 are high.

Data into a RAM is written one RAM and one bit at a time. The falling edge of RASCLK-L loads row addresses. The falling edge of CAS0-9 loads the column addresses. The write enable goes low previous to the CAS line going low. At this time, data is written into the RAM.

4.11.7.2 Memory Modifier. At the end of a vector read function, memory data (DO0-DO9) is loaded into data selectors U68 and U69. Three bits of select control (COL0-COL3) from the memory control circuit select the vector to be modified. The selected data, if gated high (from U112 flip-flop) produces READAT-L. Data can then be read from the source register. The low output of U112, along with the READAT-L bit, generates DIN providing the data the RAM memory. Write enable (WE) from the memory modifier PROM enables the RAM memory for a read or write operation. The memory modifier PROM also provides the vector generator PROMs the status with a PIXDONE signal once the command has been completed. Refer to figure 4-23.

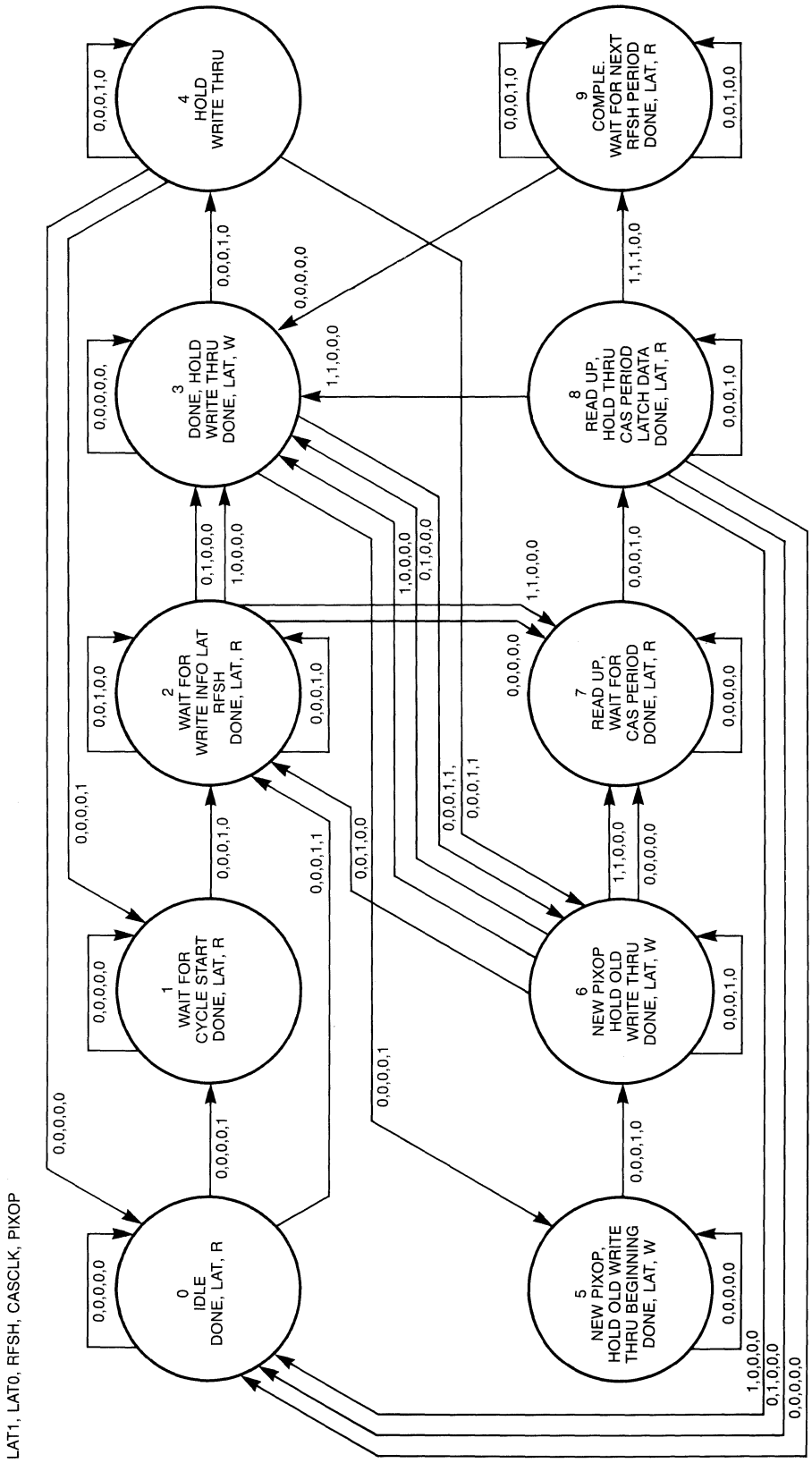


Figure 4-23. Memory Modifier State Diagram
Medium Resolution Display

4.11.8 Video Output and Timing

The screen is divided into two fields, even and odd. Sections of PROM microcode that perform video output are used for writing the even or odd fields.

The odd field is created first. Each time a line is done it begins at the top at line 239 by the Y counter. The X counter increments from 0 to 64 ten times. Then the X counter generates a horizontal sync pulse, and the Y counter decrements to 237. This is then repeated.

Once the Y counter gets to line 0 and has gone through the increment process to 64 ten times, the odd field is completed and a verticle sync pulse is generated. Then at the upper left corner of the screen, it begins over again at line 240.

On the even field, when the bottom right-most lower corner is reached again, and the field is completed, a vertical sync pulse is generated.

During this activity, the generated output is written into RAM in 10-bit bytes. During this time also, the screen and RAM are being refreshed. Refer to the timing illustrations shown in Figures 4-24, 4-25, and 4-26.

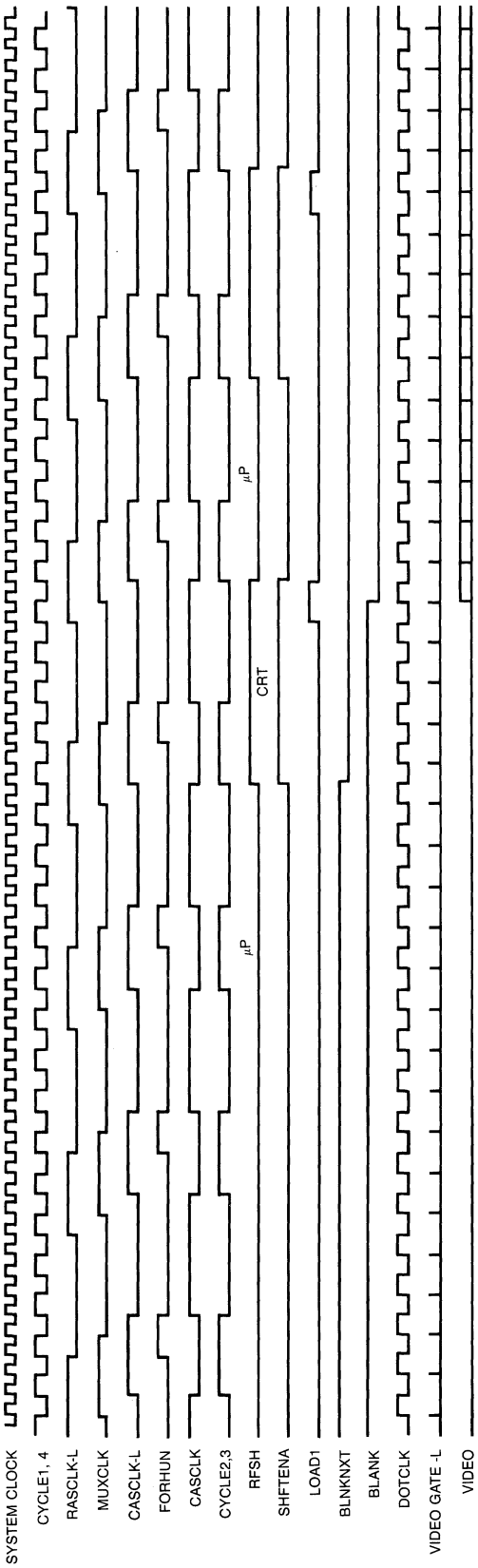


Figure 4-24. Timing Diagram, High Speed Section
Medium Resolution Display

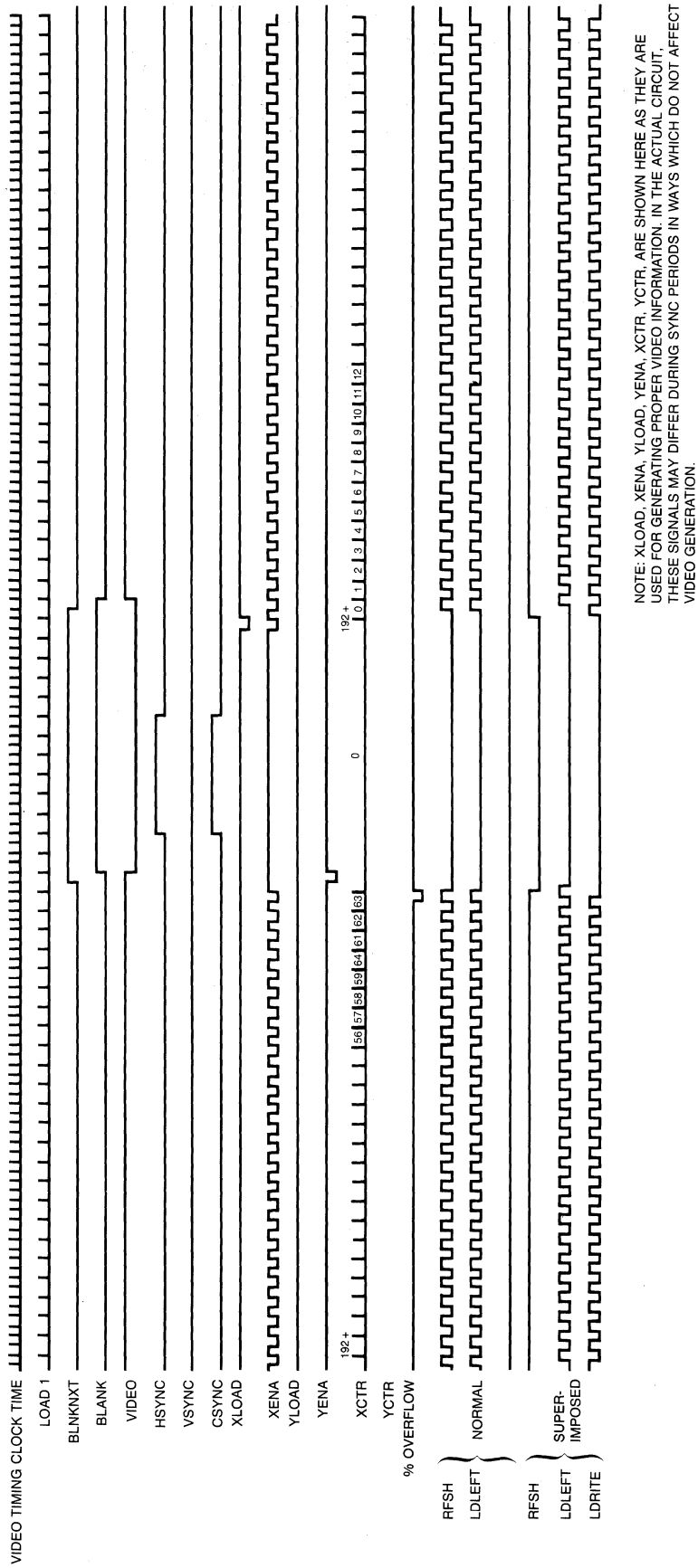


Figure 4-25. Timing Diagram, Video-Horizontal Rate Medium Resolution Display

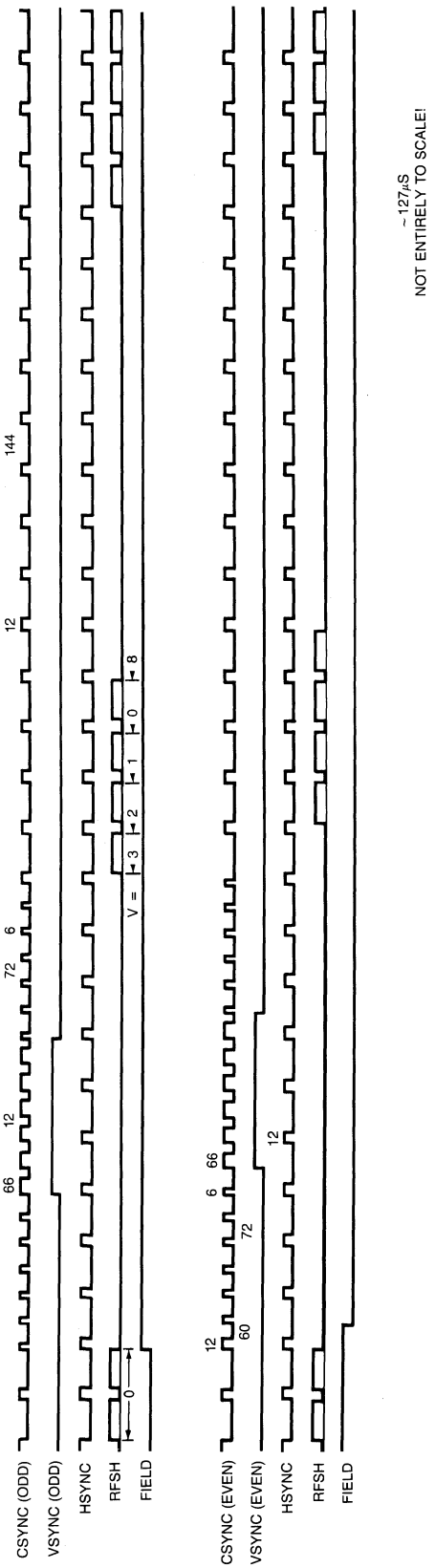


Figure 4-26. Timing Diagram, Vertical Interval
Medium Resolution Display

4.12 UTILITY BOARD

The following text is supported by illustrations and tables within the text as well as the major diagrams listed below. For specific information regarding parts or the major diagrams, refer to the Diagrams and Parts Lists Manual, Publication No. 2515-0102.

Block Diagram, Utility Card	Figure 4-27
Schematic Diagram, Utility Card	2515-4705-2D

The utility board occupies slot 9 of the card cage and is a 6-wide board. It contains test signal generator circuitry designed to generate signals for use in testing electrical and mechanical systems. It also contains an arithmetic booster and microprocessor interface circuitry to perform the necessary arithmetic computations in conjunction with the 2501 microprocessor. The functions are performed by approximately 118 integrated circuits. The hardware will be described as required in the following text.

4.12.1 Initialize and Master Clock

Initialize and the master clock are derived from the CPU via the 2501 microprocessor bus. -MCOUT and -INITL signals provide the master clock and reset to the interface circuits from whence they are distributed.

4.12.2 Booster Section

The booster section consists of the microprocessor interface circuitry and a booster processor circuit. The interface circuitry provides the control, timing, source and destination data, and the microprocessor data bus necessary for the utility board to perform computations and to function compatibly within the system. The booster processor is made up of four major parts, usable separately and jointly by the 2501 microprocessor. These four parts are a 16-bit latch, a 16-bit control register, a 1024-word by 16-bit RAM, and a 16 x 16 bit parallel multiplier/accumulator.

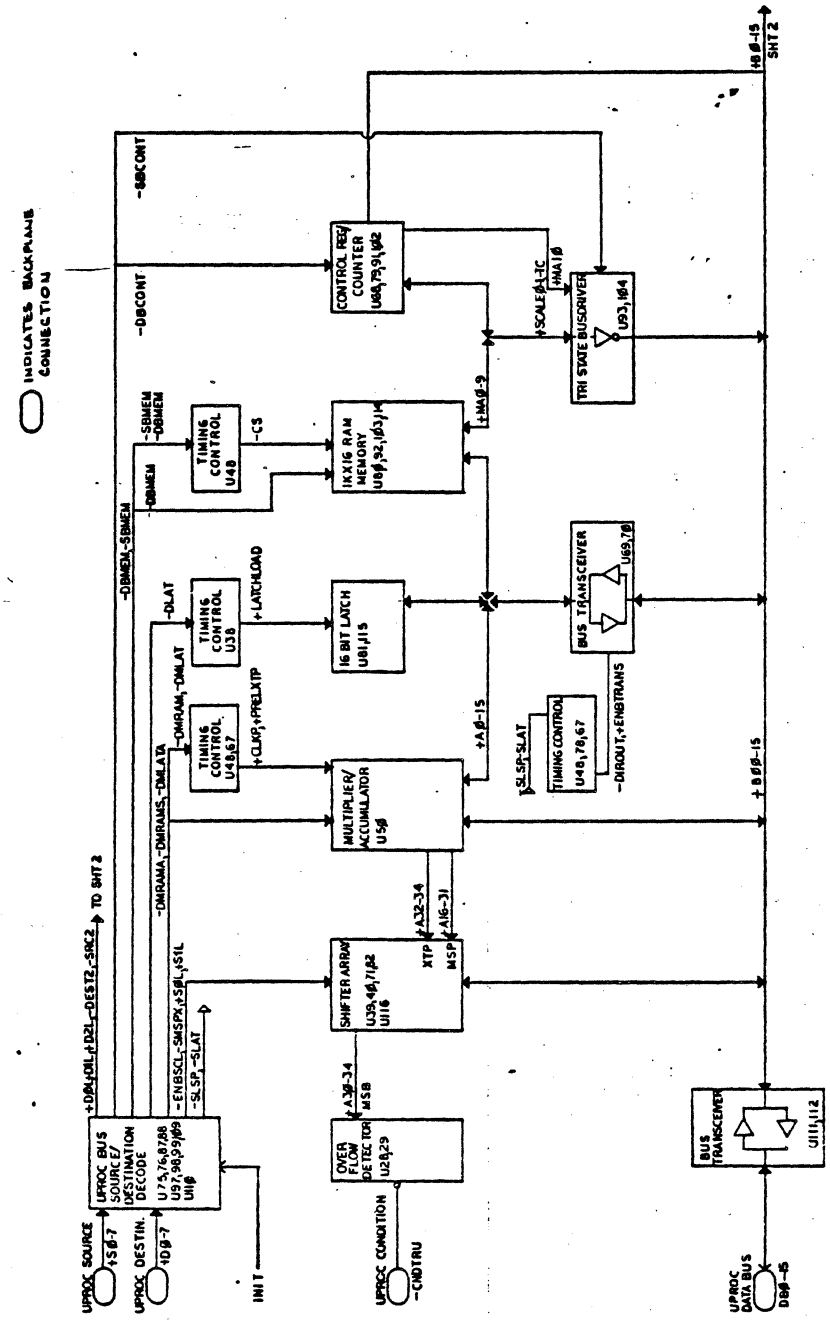


Figure 4-27. Block Diagram, Utility Board
(Sheet 1 of 3)

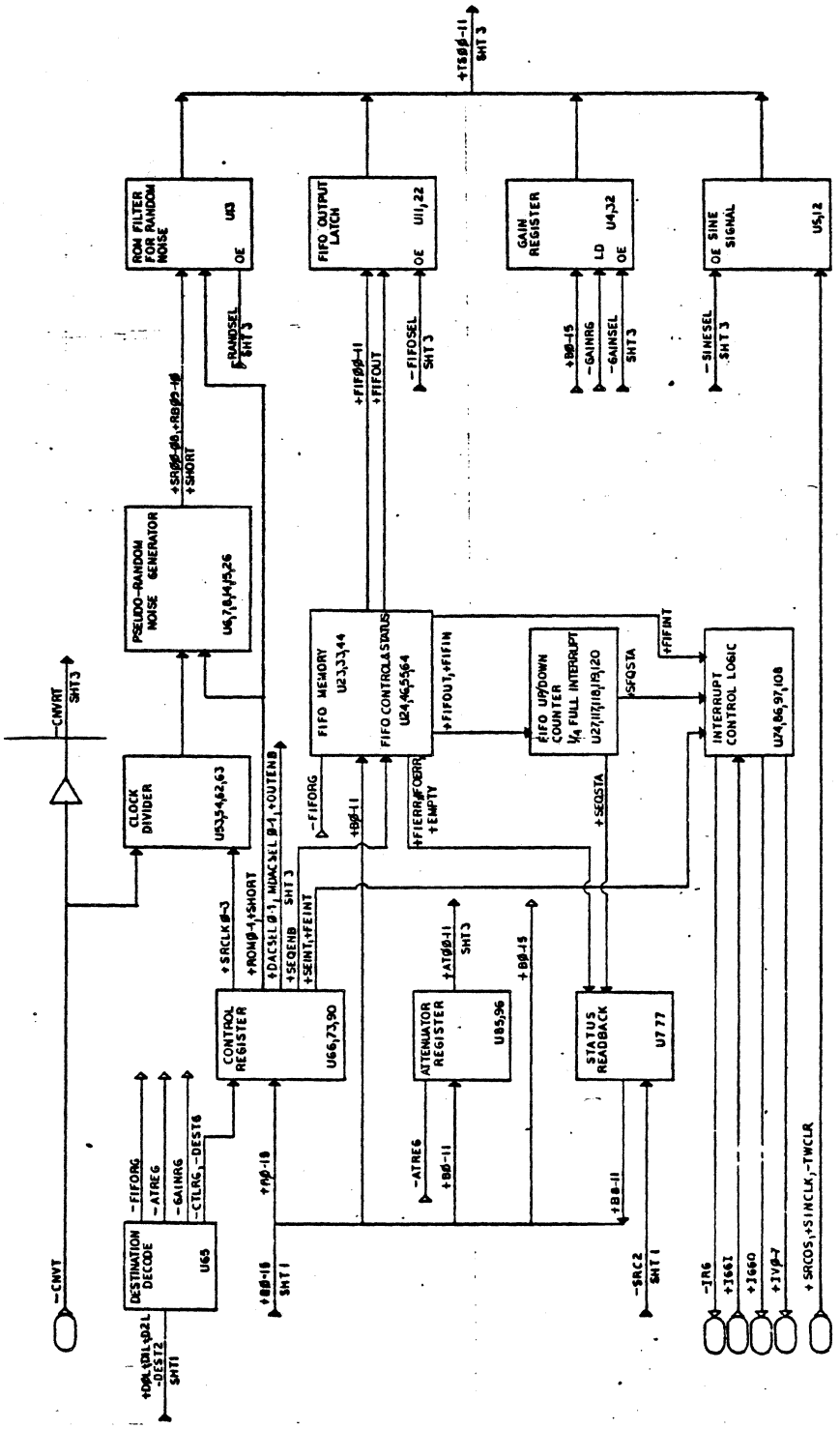


Figure 4-27. Block Diagram, Utility Board
(Sheet 2 of 3)

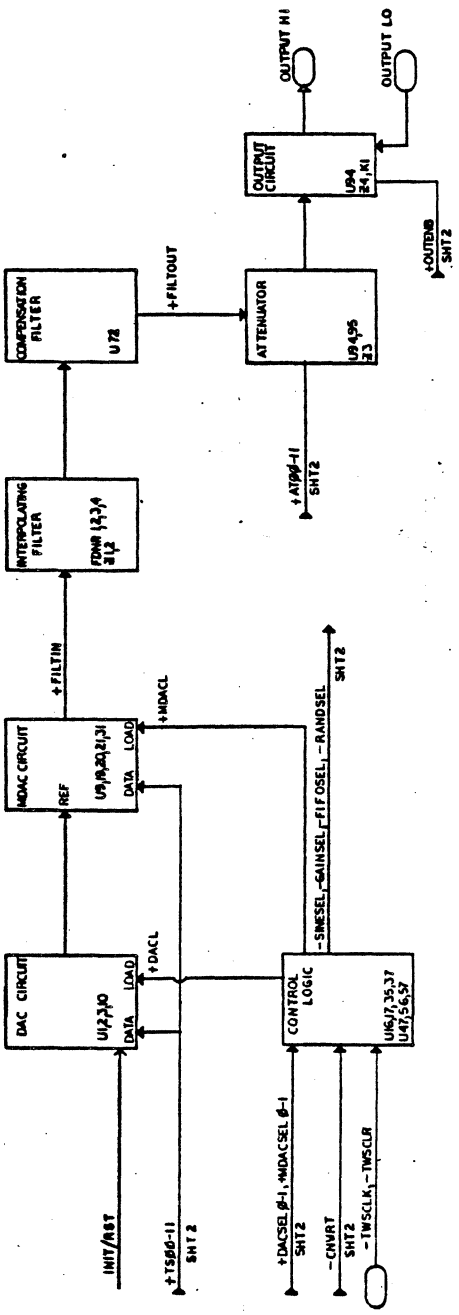


Figure 4-27. Block Diagram, Utility Board
(Sheet 3 of 3)

4.12.2.1 Microprocessor Interface. There are eight ICs and one 256 x 4 PROM used to provide control, timing and overflow detection. Normally, three clock cycles are required for microprocessor operation. The interface circuitry is capable, however, of stretching the clock from three to four intervening cycles for both source and destination data. The address and data are latched on the rising edge of the master clock signal, -MC. The clock cycle is separated by a 62.5 ns interval. The 16-bit data carried on the microprocessor data bus must be valid within 60 ns of the master clock's rising edge. Refer to Figure 4-28.

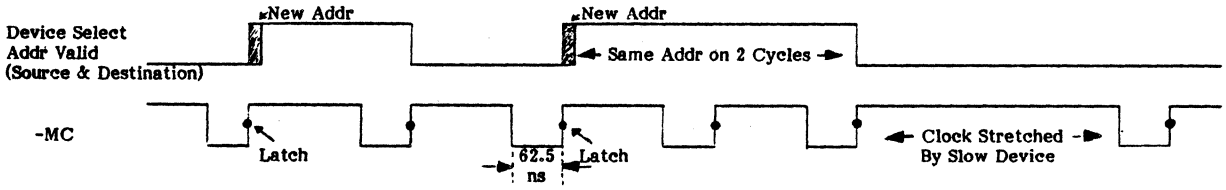


Figure 4-28. Latch Timing

The microprocessor interface circuits decode the 9 bits each of source and destination which provide the various circuits with timing control and commands. Refer to Table 4-12.

Table 4-12. Source/Destination/Condition Code Descriptions

	Source Codes	Destination Codes	Condition Codes
200	MCONT- Read or Write the Control Register		
201	MEM - Read or Write the Booster Memory		
202	LAT - Read or Write the Booster Latch		
203	LSP - Read LSP of Accum.	MRAM - Multiply by RAM	
204	MSP0 - Read MSP of Accum.	MRAMA - Multiply by RAM & add	
205	MSP1 - Read MSP downshift 1	MRAMS - Multiply by RAM & sub	
206	MSP2 - Read MSP downshift 2	MLAT - Multiply by Latch	
207	MSPX - Read MSP downshift 0, 1, 2, 3	MLATA - Multiply by Latch & add	

Table 4-12. Source/Destination/Condition Code Descriptions (continued)

Source Codes	Destination Codes	Condition Codes
210	MCONTH - Same as MCONT but only modifies upper 4 bits	VM0-False if: P30=P31=P32=P33=P34
211		VM1-False if: P31=P32=P33=P34
212	STP - Preload XTP	VM2-False if: P32=P33=P34
213	RMRAM - MRAM with round	VMX-Overflow bit for MSPX
214	RMRAMA- MRAMA with round	P34 -Most significant bit of Accum.
215	RMRAMS - MRAMS with round	
216	RMLAT - MLAT with round	
217	RMLATA - MLATA with round	

4.12.2.2 16-Bit Latch. The latch is primarily used for storage and is readable and writable using source/destination code LAT. It is comprised of two 8-bit shift/storage registers.

4.12.2.3 16-Bit Control Register. The control register consists of four synchronized 4-bit up/down counters and is readable and writable using the source/destination code MCONT. The least significant 10 bits of this register is a counter which provides the address for the RAM memory. The other 6 bits are used for control.

The control register is loaded by destination decode signal -DBCONT on the positive edge of -MC. For readback, the contents of the control register are put on the bus during the -SBCONT decode signal by drivers U93 and U104.

The 10-bit counter (MA0-10) will be clocked by the rising edge of -MC gated by the -CS decode that selects the booster memory for source, destination, or multiplier source. The counter action will depend on the contents of the +CNT0-1 bits.

The control register format is shown below.

15	14	13	12	11	10	9	8
SCALE1	SCALE0	CNT0	TC	MA10	MA9	MA8	MA7
7	6	5	4	3	2	1	0
MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

The control register bit descriptions are as follows:

Scale Control Bits 14, 15

Scale 0	Scale 1	Action for MSPX	Action for VMX
Bit 14	Bit 15		
0	0	Output bits 15-30	False if P30=P31=P32=P33=P34
0	1	Output bits 16-31	False if P31=P32=P33=P34
1	0	Output bits 17-32	False if P32=P33=P34
1	1	Output bits 18-33	False if P33=P34

Counter Control Bits 12, 13

Cont1	Cont0	Counter Control
Bit 13	Bit 12	
0	0	Hold
0	1	Count down circular
1	0	Count up circular
1	1	Count up

Bit 11	Two's complement control (-TC). If 0, signed; if 1, unsigned.
Bit 10	Loaded but not used.
Bits 0 - 9	Loaded into the memory address counter; contains memory address for RAMs.

4.12.2.4 RAM Memory. There are four 1024-word x 16-bit RAMs comprising the booster scratchpad memory. Each receives the ten least significant bits containing the address from the control register counters. The address is written into memory during the overlap of the low chip select ($\overline{CS}$) and low write enable ($\overline{WE}$) inputs. If the chip select and write enable transitions occur simultaneously the output buffers remain in a high impedance state. Write enable must be high for a read cycle and the data at the current address is read out during the overlap period when write enable is high and chip select is low. Total read or write access time is 150 nanoseconds.

Memory is loaded during the -DBMEM destination decode on the rising edge of -MCT0A bus signal. Memory readback is accomplished through bus drivers U69 and U70 enabled during -ENBTRANS with the -DIROUT signal directing the data onto the microprocessor bus.

The memory can be read or written at the address in MCONT by using the source/destination code MEM. The counter may either remain unchanged or do a post-decrement or a post-increment (mod 16) depending on the state of the counter control bits in MCONT (see source/destinate decode descripton in table 4-12). In the count up "circular" mode the carryout of the least significant 4 bits is inhibited. So, if octal 435 is loaded into the counter, successive reads would access words 435, 436, 437, 421, 422..... This implements the 16-bit circular buffer.

4.12.2.5 16 x 16-Bit Parallel Multiplier/Accumulator. The multiplier/accumulator is a 64-pin dual-in-line chip with a total multiply-accumulate time of typically 115 nanoseconds. The multiplier consists of 16 data bits supplied by the CPU via the data bus. The numbers may be interpreted as signed or unsigned depending on the state of TC (two's complement control). An integer multiply is performed to obtain a 32-bit result which is sign extended to 35 bits, then loaded, added or subtracted into a 35-bit accumulator. Provision has been made to implement rounding by adding a logic 1 into bit 15 of the accumulator. Refer to Figures 4-29, 4-30, and 4-31.

Note that if the multiplier and multiplicand are interpreted as signed 2's complement fraction integer (i.e., binary point between bits 14, 15) then the binary point of the result will be between bits 29 and 30 of the product, and the accumulated output can contain numbers with the range $-16 \leq x < 16$ with 30 fraction bits.

Data Inputs The data inputs are loaded into the X and Y inputs (or registers) at the rising edge of clock X or clock Y (CLKXY). The multiply/accumulate begins at this time. The LSP outputs are timeshared with the Y data inputs.

Output The output is divided into three parts: LSP (least significant part), MSP (most significant part), and XTP (extended product). The product generated is loaded into the output registers at the rising edge of CLKP.

TC When TC is high the inputs are two's complement numbers. When TC is low the inputs are unsigned magnitude numbers. The TC signal is loaded at the rising edge of CLKXY. The TC signal must be valid over the same period that the input data is valid.

RND When round control RND is high, a logic 1 is added to the MSB (bit 15) of the LSP multiplier to round up the product in MSP and XTP. The RND signal is loaded into the control register at the rising edge of CLKXY.

ACC When accumulation control ACC is high, the contents of the output registers are added to the next product generated and the sum is stored back into the output registers at the rising edge of the next CLKP. When ACC is low, multiplication without accumulation is performed and the next product generated is directly stored into the output registers.

SUB This is the addition/subtraction control and when both ACC and SUB are high, the contents of the output register are subtracted from the next product generated and the difference is stored back into the output registers on the rising edge of the next CLKP. When ACC is high and SUB is low, addition is performed. When ACC is low, SUB is inactive (don't care condition).

PREL The preload control. All output buffers are at high impedance (disabled) when PREL is high. When TSL (three-state least), TSM (three-state most), or TSX (three-state extended) control is also high, the initial contents of the corresponding register can be preset to the preload data applied to the outputs at the rising edge of CLKP. If TSL, TSM, or TSX is low while PREL is high, the contents of the corresponding output register remain unchanged. The LSP, MSP, or XTP output buffers are disabled (high impedance state) when TSL, TSM or TSX is high. These are direct control signals. The output drivers are enabled when TSL, TSM or TSX is low, and PREL is low.

There are ten destination codes and operations which cause a multiply to be performed. These are listed below.

Operation	**Multiply Performed**
MOV RO, MRAM	<ACCUM> = RO x Memory (Note: This operation will cause the accumulator to be loaded with the product of RO and the content of Memory at the current address. After being used, the address will change according to the counter control bits in MCONT.)
MOV RO, MRAMA	<ACCUM> = RO x Memory + <ACCUM>
MOV RO, MRAMS	<ACCUM> = RO x Memory - <ACCUM>
MOV RO, MLAT	<ACCUM> = RO x Latch
MOV RO, MLATA	<ACCUM> = RO x Latch + <ACCUM>

The above five operations can also be performed with the rounding bit being added in: RMRAM, RMRAMA, RMRAMS, RMLAT, RMLATA.

The accumulator may be loaded using these operations. There are no destination decodes that will load the accumulator registers directly except for the extended product, XTP.

The upper three bits of the accumulator may be loaded without affecting the other 32 bits by the following operations:

MOV RO, XTP ACCUM bit 34 = bit 0 of RO

ACCUM bit 33 = bit 15 of RO

ACCUM bit 32 = bit 14 of RO

(Note: This bit ordering is best for saving and restoring the state of the booster, which is also the only intended function for the preload XTP function.)

CAUTION

When one of the 10 destination codes performing multiply are used, the XTP destination code must not be used in the immediately succeeding instruction.

Five source codes and one condition code are used to read out the contents of the accumulator. See Figure 4-31. These are as follows:

LSP Sends bits 0-15 to the data bus

MSP0 Sends bits 15-30 to the data bus (normal fractional multiply result)

MSP1 Sends bits 16-31 to the data bus

MSP2 Sends bits 17-32 to the data bus

MSPX If SCALE0-1 in MCONT = 0, sends bits 15-30 to data bus
If SCALE0-1 in MCONT = 1, sends bits 16-31 to data bus
If SCALE0-1 in MCONT = 2, sends bits 17-32 to data bus
If SCALE0-1 in MCONT = 3, sends bits 18-33 to data bus

P34 A condition code. Since the most significant accumulator bit cannot be put onto the data bus, this code is provided. The condition will be found TRUE if bit 34 of the accumulator is 1.

CAUTION

The foregoing six codes will produce unpredictable results if one of the 10 multiply destination codes are in the immediately preceeding instruction.

The following sequence may be used by an interrupt routine intended to use the booster to save and restore the booster state.

```
To Save State:  MOV MCONT,      SP0       ;Save Control Register
                MOV LAT,        SP1       ;Save Latch
                MOV LSP,        SP2       ;Save bits 0-15
                MOV MSP1,       SP3       ;Save bits 16-31
                MOV #0140000,   MCONT     ;Set scale to 3
                MOV MSPX,       RO        ;Get bits 18-33
                BIC #1,         RO        ;Store bit 34
                JMP -P34,       +2        ;In the LSB using
                OR  #1,         RO        ;Condition code
                MOV RO,         SP4       ;Save bits 32-34
                MOV #04000, MCONT         ;Set unsigned MPY
To Restore State: MOV #0100000, LAT       ;LAT = 1/2
                MOV SP3,        MLAT      ;Restore MSP, but one bit too far
                                           right
                                          ;by multiplying by
                MOV #1,         MLATA     ;Shift it left one bit/MPY by 2
                                          ;to get in upper accumulator
                MOV #1,         LAT
                MOV SP2,        MLATA     ;Restore LSP with MPY by 1
                MOV SP1,        LAT       ;Restore Latch
                MOV SP0,        MCONT     ;Restore Control Register
                MOV SP4,        XTP       ;Restore Extended Product
```

4.12.2.6 Overflow Detector. The overflow detector circuit is used by the microprocessor to monitor the current state, or condition, of the booster computations. The condition testing is performed by addressing source addresses 210 through 214. See table 4-12. The upper accumulator bits are tested as shown in the table. In the case of the VMX source, the SCALE0-1 bits in MCONT are used to determine which bits are tested. In this case, the overflow bit in the MSPX source is tested. In the case of the P34 source, only the 34th bit is tested.

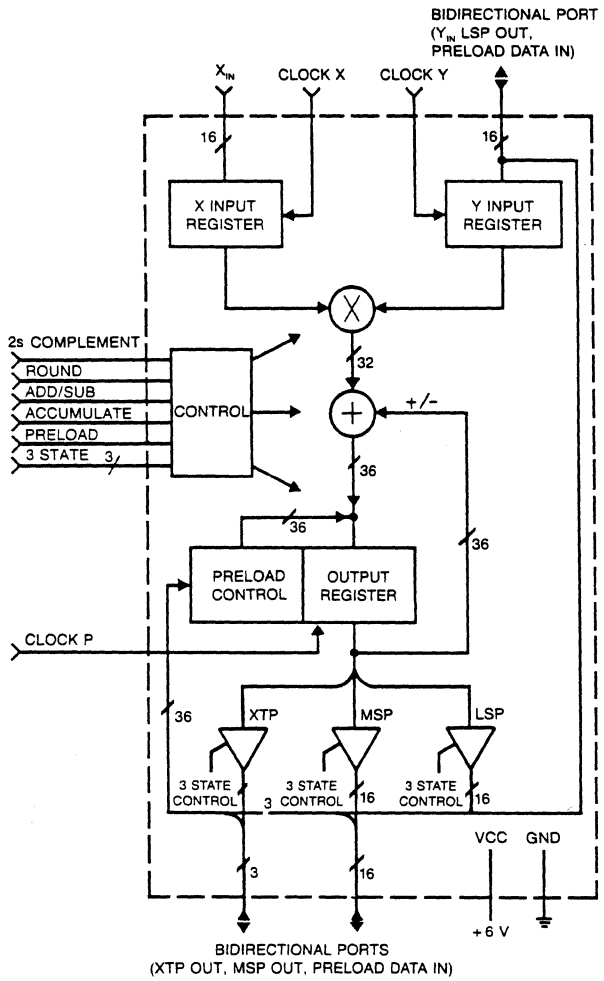


Figure 4-29. Multiplier-Accumulator Functional Diagram

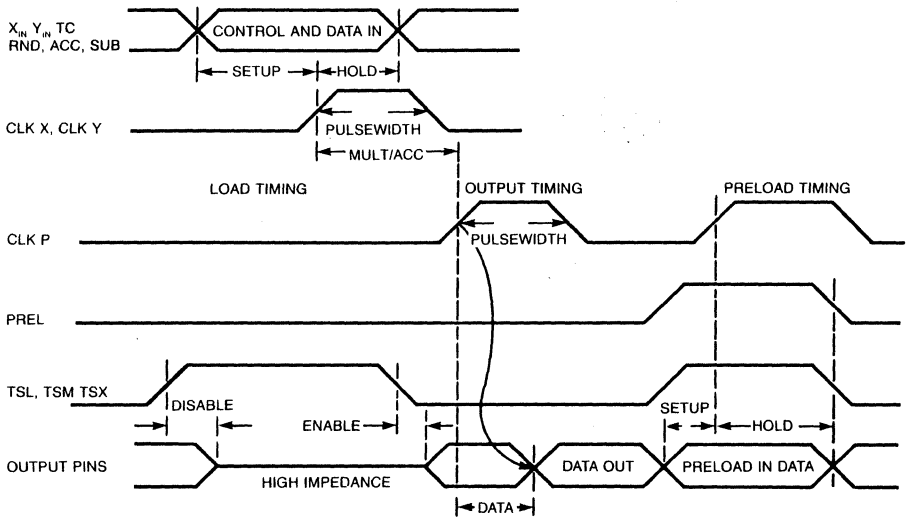


Figure 4-30. Multiplier-Accumulator Timing Diagram

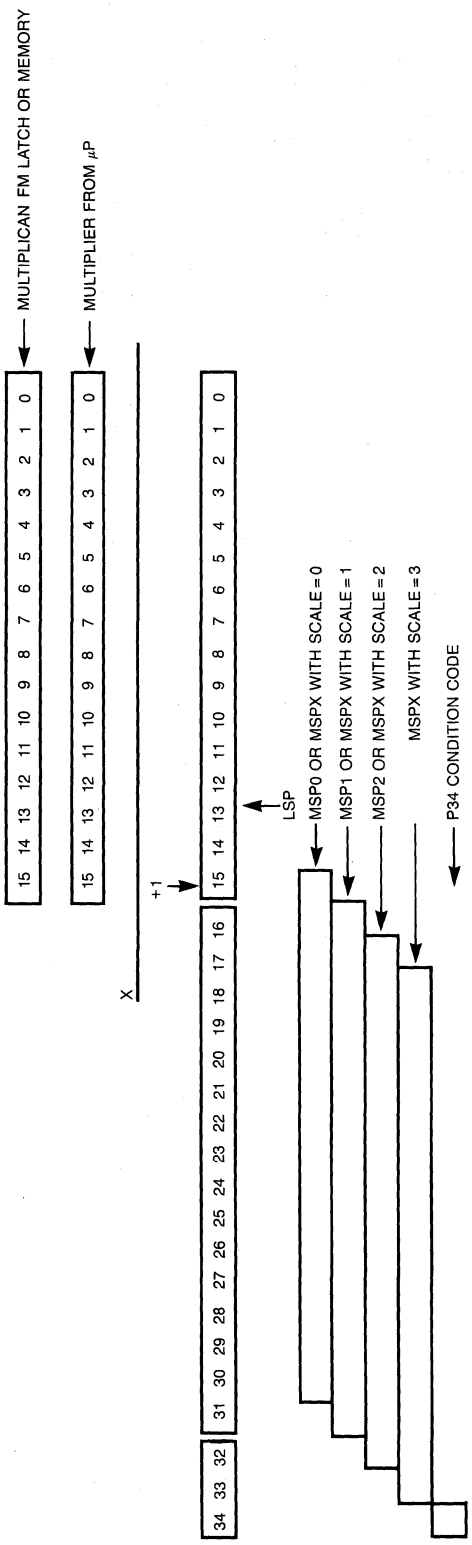


Figure 4-31. Multiplier-Accumulator Bit Configuration

4.12.3 Test Signal Generator

The test signal circuitry consists basically of one digital-to-analog converter (DAC) and two multiplying digital-to-analog converters (MDAC). The DAC and first MDAC can be loaded from any one of four signal sources off the tri-state bus. These signal sources are the sine wave generator, random noise generator, gain register, and FIFO storage register. Refer to Figure 4-32.

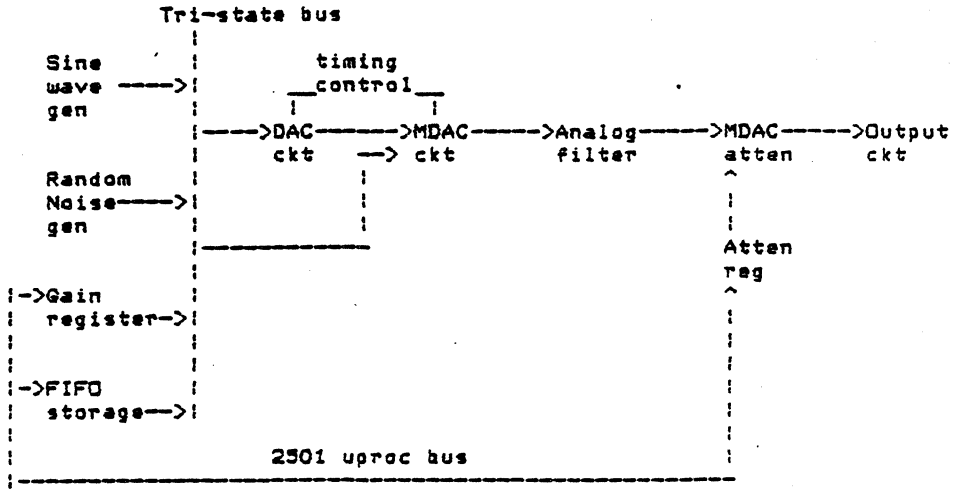
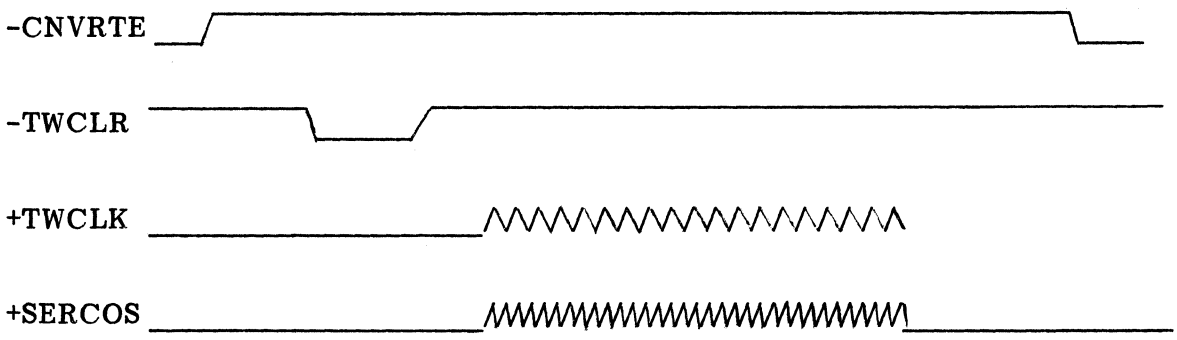


Figure 4-32. Simple Diagram, Test Signal Generator Circuits

4.12.3.1 Sine Wave Generator. The sine wave generator uses the serial data stream generated by the Channel Control board. This signal is used by the 25 kHz Channel board for the Zoom translation operation and is generated by the twister circuitry. The sine wave may be output by itself or may be multiplied with the random noise generator to output a translated (and bandlimited) random signal.

The sine wave signal is acquired in the same manner as it is by the Channel boards. The 16 bit serial stream +SERCOS is sampled by the +TWCLK signal from the Channel Control board. The two 8-bit shift registers are cleared by the -TWCLR signal also generated by the Channel Control board. Although 16 bits are sampled, only the last 12 bits are used. These are the last 12 bits shifted in to the registers.



4.12.3.2 Random Noise Generator. This 23-bit long shift register circuit generates pseudo-random noise. The single bit noise output is filtered by a ROM with taps into the shift register, serving as delays, to a one-bit input digital filter. The output is 8 bits wide. Four different filters may be selected which have different passband characteristics. The bandwidth may be modified by controlling the clock to the shift register. The output rate filter is fixed, and signals generated with the divided clock have more harmonic content due to the fact that the signal is "held" for a number of sample times. The random noise generator contains the following major components:

Pseudo-Random Noise Generator
Clock Divider
ROM Filter for Random Noise

a. Pseudo-Random Noise Generator

As stated before, the 23-bit long shift register circuit generates pseudo-random noise. Using feedback from stages 18 and 23, the sequence is unique and contain random frequency components in a one-bit signal output. The sequence repeats every 2**23 sample outputs. The sequence is considered pseudo-random because it repeats and does not have an infinite number of frequencies. The frequency resolution of this signal is the clock frequency/2 divided by the shift register length. For the output rate of 64 kHz, the frequency resolution is 15.6 millihertz. During the generation of the entire sequence, the one-bit signal changes to create all the

frequencies from DC to the clock frequency/2 with this frequency resolution. This sequence then repeats again. The repeat time of the signal is 128 seconds at the 64 kHz output rate.

b. Clock Divider

The clock divider circuit is used for random signals only and is simply a series of 4-bit counters. The rate to be used is selected by two 8 to 1 multiplexers. The selection is controlled by the SRCLK0-3 bits in the TSCTL control register.

The circuit allows the output rate of the pseudo-random signal to be slowed down by factors of 2. This allows the output rate to be modified to match the effective sample rate of the input data acquisition sample rate. The clock signal (-CNVRT) is the same clock that is used by the front end A/D circuitry.

The front end uses digital filtering to eliminate aliasing of frequencies during data acquisition. There are no such filters for output and therefore, the reduced bandwidth signals generated in this way will have harmonic frequency content due to the fact that the successive outputs at the D/A converter are held for one or more samples. The analog smoothing filter is set for the 64 kHz output rate (25 kHz bandwidth) so that this is the only rate at which harmonics do not occur.

c. ROM Filter for Random Noise

As noted above, there is a smoothing filter to remove harmonics at the D/A output. This filter will make the single bit output signal Gaussian in its amplitude distribution. This filter is not effective when the output rate is reduced as noted above. There are four digital filters which can be used in conjunction with the clock divider to make the signal output more Gaussian.

The single bit noise output is filtered by a ROM with taps into the shift register serving as delays to a one bit input digital filter. The output is 8 bits wide and is more Gaussian than the binary sequence. Four different filters may be selected which have different passband characteristics.

Filter #1 keeps the signal a two-valued (binary) signal except that the output is symetric about 0 volts.

Filter #2 has a cutoff point at the half-band frequency point.

Filter #3 has a cutoff point at the quarter-band frequency.

Filter #4 has a cutoff point at the one-eighth-band frequency.

4.12.3.3 Gain Register. The gain register is simply 12 bits which are used to determine output amplitude or DC level. It is interpreted as a signed 12-bit number when loaded into either the DAC or MDAC. The Gain register is loaded off the microprocessor bus. The loading of this register can be synchronized with the loading of the DAC/MDAC allowing the amplitudes of both a FIFO generated signal and a selected gain to be modified at the same time.

4.12.3.4 FIFO Storage. This is a 64-word FIFO which can be loaded off the microprocessor bus. The FIFO is unloaded with the sample clock and loaded into the DAC/MDAC as data.

Fifo storage circuitry consists of the following sub-sections:

FIFO Memory
FIFO Control and Status
FIFO Output Latch
FIFO Up/Down Counter
Interrupt Control Logic

a. FIFO Memory

The FIFO Memory is 12 bits of data 64 words deep. The FIFO is loaded directly by the microprocessor at destination address TSFIFO. This loading is asynchronous with the output from the FIFO which is controlled by the -CNVRT signal.

b. FIFO Control and Status

The timing of loading and unloading is controlled so that the last data value is held if the FIFO is emptied before the next point is loaded into the FIFO. The status bits are generated to indicate any error conditions that may have occurred.

If shifting data out of the FIFO is attempted when the FIFO is empty, the FOERR status is set.

If the microprocessor attempts to shift in a value when the FIFO is full, then the FIERR status is set.

The FIFO is considered EMPTY when the last point has been shifted out which is indicated by the FIFO chips output not being ready by the time +DACL occurs.

c. FIFO Output Latch

The FIFO output latch holds the last FIFO value shifted out in case the FIFO output error condition occurs. This holds the signal output at the last point to avoid a discontinuity. Loading of the latch is ANDed with the EMPTY condition.

d. FIFO Up/Down Counter

This counter allows the microprocessor to find out if the FIFO is less than 1/4 full. This can be done either by polling the status register FIFOST or using the SEQSTA interrupt condition.

The FIFO load and unload events are latched and the counter incremented or decremented respectively and synchronized with the microprocessor clock (-MC). If the FIFO has been both loaded and unloaded between clocks, then the up/down counter is not modified.

e. Interrupt Control Logic

All interrupts are associated with the FIFO status. FIFO error status is latched if the interrupt is enabled and if the FEINT bit is set in TSCTL register. The SEQSTA condition (<1/4 full) is also latched if the enable SEINT bit is set in TSCTL.

The request for interrupt at priority level 6 (-IR6) is then generated synchronized with the microprocessor clock at U86. When interrupt grant at level 6 (+IG61) returns and this board was the one requesting the interrupt, the vector is put on the vector bus by V108. If this was not the requesting device, the interrupt grant is passed on to the next board (+IG6O).

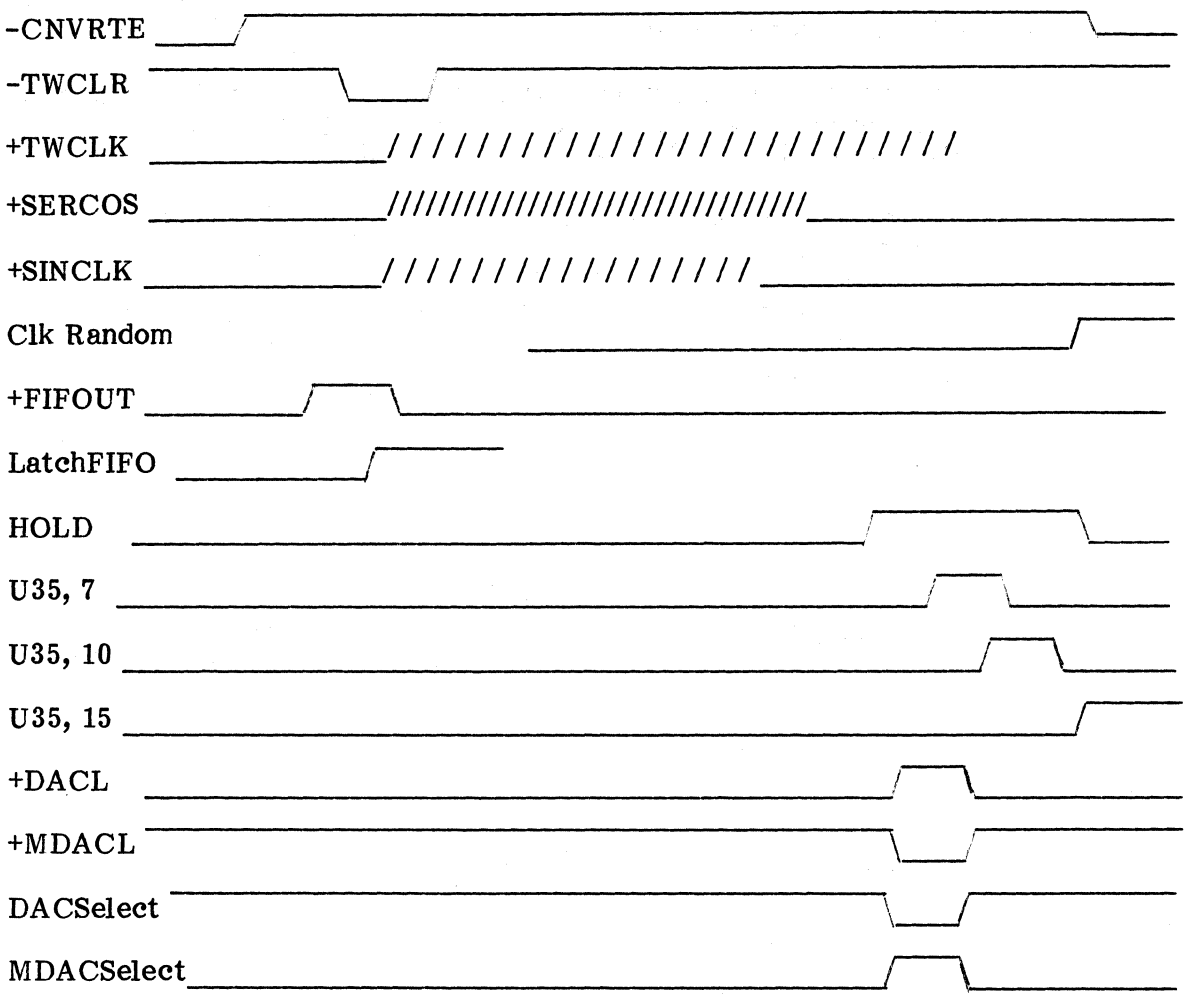
4.12.3.6 Timing Control Logic. The control logic section of the board generates the timing for the signal generators and timing for loading the DAC and MDAC circuits.

All the timing is controlled by and synchronized with signals from the Channel board in the digital front end. The -CNVRT signal occurs on every input sample and is therefore used to synchronize the change in the output signal. The remaining timing signals are those provided by the twister circuit on the Channel Control board. In particular, the TWCLK and TWCLR signals supply the remaining clock needs of the output signal generator.

U17 allows the -CNVRT clock from the Channel Control board to be divided down by factors of two. A change in this jumper selection (which is normally set for E11 - E12) requires a change in the interpolating filter to correspond to the output rate reduction. There are pads provided on the layout to allow this change of capacitor values which is not programmable. -CNVRTE is the resulting divided signal.

U37 counts up 16 TWCLK transitions. During this time the sine wave generator is enabled to shift in 16-bit data values with +SINCLK. After this time TWCLK clocks a Johnson counter which provides the additional timing for the DAC and MDAC selection and loading.

FIFO output is done during TWCLR and the output value is latched at the rising edge of -TWCLR. The random signal is clocked at the end of the cycle, falling edge of -CNVRT.



U57 provides the selection of the various devices to be loaded into the DAC and MDAC. The source to be loaded into the DAC is normally selected until the DAC has been latched on the rising edge of +DACL. The MDAC data source is then enabled on the tri-state bus and is latched on the rising edge of +MDACL.

4.12.3.7 DAC Circuit. The DAC latch is loaded on the rising edge of +DACL. The DAC circuit is configured for bipolar operation. The output span is set for ±5 volts to avoid overloading the interpolating filter. This loss in the signal amplitude will be recovered in the attenuator circuit.

4.12.3.8 MDAC Circuit. The latches which feed the MDAC are loaded on the rising edge of +MDACL. The circuit is configured for a four-quadrant multiplying DAC.

4.12.3.9 Sample and Hold. The sample and hold circuitry has been removed from the current board but the component layout on the schematic and PC board remains. This circuit was intended to hold the previous output value while the new value was

settling but has been removed. The simple R-C filter (R28, C71) was intended to limit the input bandwidth to the sample and hold. The timing signals for this circuit are still generated in the control logic.

4.12.3.10 Analog Filter. The analog filter section smooths the discrete values output by the DAC and MDAC and consists of the following two circuits:

Interpolating Filter
Compensation Filter

a. Interpolating Filter

The interpolating filter is a fixed cutoff filter used to smooth the discontinuities generated by the discrete nature of the D/A output. It is set for a 25 kHz cutoff to correspond with the 64 kHz output rate. In this respect, the signal generator is the mirror image of the Channel board data acquisition circuit. However, since there are no digital filters to allow different bandwidths the interpolation of different bandwidths will have to be provided by microcode algorithms. The filter itself is identical to the filter on the 25 kHz Channel board using the same components.

b. Compensation Filter

Following the Interpolating Filter there is a unity gain buffer and a second order filter (U72) which compensates for the loss in signal level near the 25 kHz bandwidth due to the fact that the signal is "held" at the same value for a finite period of time. A signal "held" in this way will have a characteristic "sinx/x" rolloff with the first null at half the output frequency. The amount of droop is approximately 2.3 dB at the 25 kHz cutoff point.

4.12.3.11 MDAC Attenuator & Attenuator Register. The attenuator register is loaded directly from the microprocessor. The output of the attenuator is then multiplied by 2 to correct for the reduction in gain to the ± 5 volt span at the DAC. The MDAC is configured as a unipolar (2-quadrant) multiplying DAC.

4.12.3.12 Output Circuit. The quasi-differential output circuit allows small DC offsets in external equipment connected to the output to be compensated for without having to connect the chassis grounds of the two pieces of equipment. Such a connection might easily result in increased noise in the inputs due to ground loops.

The output signal ground is measured and inverted. It is then summed with the output signal and the net effect is to cancel the DC offset present in the output signal ground return. If the offset is greater than approximately 0.5 volts

then the offset potential is clamped to ground through diodes CR3, 4. A potentiometer in the output section allows elimination of the overall internal DC offset voltage.

The current buffer amplifier (U106) allows higher drive current than would be available from the output op amp (approx. 75mA output current). If voltages are mistakenly applied to the signal output, a pair of clamping diodes (CR1, 2) will protect the circuitry. If the applied signal has too large a potential, the 51 ohm resistor in series with the output will serve as a fuse.

Relay K1 is normally open so that power up transients are not seen at the output BNC. The signal is only applied when the OUTENB bit in the TSAUX register is set.

4.12.3.13 Destination Address Registers.

a. TSCTL, Test Signal Control Register, Destination Address 220.

15	14	13	12	11	10	9	8
SRCLK3	SRCLK2	SRCLK1	SRCLK0	MDACSEL1	MDACSEL1	DACSEL1	DACSEL0

7	6	5	4	3	2	1	0
SHORT	ROM1	ROM0	SEQENB	SEINT	FEINT	FINIT	RST

RST Reset. Bit 0 is used to clear the latched input to the DAC and MDAC and disable the loading clock to the DAC and MDAC latches. This is a software reset which will cause the output to be set to 0.0 volts.

FINIT FIFO INITalize. Bit 1 is used to clear FIFO error interrupt and clear the FIFO error status bits. This is the only control bit which is active low and it must be returned to the ON condition for the error status bits and interrupts to work again.

FEINT FIFO Error INTerrupt Enable. Bit 2 enables a microprocessor interrupt when a FIFO error occurs.

SEINT SEquence INTerrupt enable. Bit 3 is the enable interrupt of the microprocessor when the output sequence FIFO goes from 1/4 full to less than 1/4 full.

SEQENB SEQuence ENaBle. Bit 4 allows the FIFO to load, unload, and enable the output sequence counter. When cleared this bit will clear and disable the sequence counter, reset the FIFO, and clear the FIFO interrupt.

ROM0-1 ROM filter select bits. Bits 5 and 6 select the ROM filter to be used to filter the pseudo random sequence. Selection is made as follows:

ROM0	ROM1	FILTER
0	0	No filter
0	1	1/2 band
1	0	1/4 band
1	1	1/8 band

SHORT Short random sequence enable, bit 7. This feature has been disabled but the control bit is still latched.

DACSEL0-1 DAC source SELect bits 8 and 9 choose the source of the data to be loaded into the DAC as follows:

DACSEL1	DACSEL0	SOURCE
0	0	Sinewave Generator
0	1	Random Sequence Generator
1	0	FIFO port
1	1	Gain Register

MDACSEL0-1 MDAC source SELect bits 10 and 11 are used to choose the source of the data to be loaded into the FIFO.

DACSEL1	DACSEL0	SOURCE
0	0	Sinewave Generator
0	1	Random Sequence Generator
1	0	FIFO port
1	1	Gain Register

SRCLK0-3 Sample Rate Clock control bits 12 through 15. These bits choose the clock rate to be used for the pseudo-random sequence output where the number loaded into these bits is the power of two factor by which the basic 64 kHz clock is divided. For example, a 0 in these bits will select the maximum bandwidth (25 kHz) and a 2 will select the sample rate divided by 4 (bandwidth = 5 kHz).

b. TSGN, Test Signal Gain Register, Destination Address 222

- not used-	11	10	9	8	7	6	5	4	3	2	1	0
	- Test Signal gain register -											

The gain value is the least significant 12 bits of the TSGN register. It is a 2's complement number which will be used as a signed fraction by which to multiply when selected for the MDAC and a 2's complement level when selected for the DAC.

c. TSFIFO, Test Signal FIFO Load Register, Destination Address 223

- not used-	11	10	9	8	7	6	5	4	3	2	1	0
	- Value to be loaded into FIFO -											

The 64 work FIFO is loaded through this register. The least significant 12 bits are stored.

d. TSATTN, Test Signal Attenuator, Destination Address 224

- not used-	11	10	9	8	7	6	5	4	3	2	1	0
	- Attenuator value +AT00-11 -											

This register is loaded with the value to be used in the output attenuator circuit which follows the analog smoothing filter. It is a twelve bit unsigned integer fraction by which the filter output will be multiplied.

e. TSAUX, Test Signal Auxiliary Register, Destination Address 226

- not used -	0
	OUTENB

OUTENB Output ENaBle. This bit enables the final output of the test signal generator. It controls the opening and closing of the relay on the output. The other bits are not used.

4.12.3.14 Source Address Registers.

a. FIFOST, FIFO Status Register, Source Address 220

15 - 4	3	2	1	0
Not Used	SEQSTA	EMPTY	FOERR	FIERR

This register is read back to allow the microprocessor to interrogate the status of the FIFO output circuitry. This register will typically be read when an interrupt occurs to determine the cause of the interrupt but can also be read at any time.

FIERR FIFO Input ERRor. Bit 0 is set if the microprocessor attempts to load data into the FIFO and the FIFO is not ready to accept it. The not ready condition could exist if the FIFO is full or if it is in the RESET state (FINIT is low).

FOERR FIFO Output ERRor. Bit 1 is set if the DAC or MDAC requires the FIFO data to be latched and the output of the FIFO is not ready. The FIFO not ready condition will exist when the last word in the FIFO has been shifted out and the next value (if one was entered) has not made it to the output stage of the FIFO yet.

In the case of a FIFO output error, the last value to come out of the FIFO will be held by the DAC/MDAC until a new value makes its way through the FIFO.

EMPTY FIFO EMPTY. Bit 2 indicates that the last value has been shifted out of the FIFO. An error condition does not exist in this case since there is still time to load another value into the FIFO. The amount of time is determined by the output clock rate which is expecting another point to come out of the FIFO in approximately 12 μ sec. (16 μ sec [time between outputs], 4 μ sec [overhead for current output]; FOERR is set if the next point is not loaded by this time.)

SEQSTA SEQuence STAtus. Bit 3 indicates that the FIFO has gone from 1/4 full or greater to less than 1/4 full. This status bit is typically used to generate an interrupt so that the FIFO interrupt may be serviced at a more leisurely pace than the EMPTY bit interrupt would allow. This bit is set approximately 280 μ sec before the FIFO output error condition occurs.

4.12.3.15 Test Signal Interrupts. There are two interrupts which may be generated by the test signal hardware. They are both related to the FIFO output circuit. These are interrupt vectors 220 and 221. Both interrupts have microprocessor bus priority 6.

a. Interrupt Vector 220 occurs when a FIFO error occurs or the FIFO EMPTY condition occurs. The interrupt will be generated if the FEINT bit in the TSCTL register is set and any one of FIERR, FOERR, or EMPTY bits is set in the FIFOST register.

The interrupt is cleared by lowering the interrupt enable bit FEINT in TSCTL. The error bits are cleared by lowering FINIT in TSCTL.

b. Interrupt Vector 221 occurs when the FIFO goes from 1/4 full or greater to less than 1/4 full. The interrupt is generated when the SEQSTA bit is set in FIFOST and the interrupt is enabled by SEINT in TSCTL. The interrupt is cleared by lowering the interrupt enable bit SEINT in TSCTL. The SEQSTA bit is cleared by lowering the SEQENB bit in TSCTL.

4.13 CHANNEL AND CONTROL BOARDS

The following subparagraphs describe the relationships, functions, and major circuitry of the channel control board, PN 2515-4701, and 25 kHz channel board, PN 2515-4700. The text references the below listed diagrams.

Functional Block Diagram, Channel and Control Board	Figure 4-33
Schematic Diagram, Channel Control Board	2515-4701-2D
Schematic Diagram, 25 kHz Channel Board	2515-4700-2D

The channel board(s) acquires analog signals and processes them with a combination of analog and digital filtering under the direction of the channel control board. The control board provides the necessary timing and data buffering for up to four 25 kilohertz channel boards for a maximum of 16 channels. The control board is interfaced to the CPU (2501 microprocessor) and the memory controller. Two of the several functions of the memory controller are to monitor the status of the FIFO holding data from the channel boards and to control direct memory access (DMA) transfers into the Unibus (PDP-11) memory. The 25 kHz channel boards are referred to in numerical order such as channel board #1 through #4, with #1 assumed to contain channels 1 through 4, board #2 to contain channels 5 through 8, and so on. The channel control board is referred to as the control board.

4.13.1 Channel and Control Board Relationship

The close relationship between the control board and channel boards requires that the boards be described within one major paragraph.

The control board and up to 4 channel boards (four 25 kHz) from a system in which up to sixteen 25 kHz BW baseband or 8 channels of zoomed data may be acquired and directly accessed (DMA'ed) to memory. The hardware involved with the filtering computations is sequenced by the control board and all channel boards perform their calculations in step under the direction of the control board. The results of the computations (data) are fed back to the control board. Selected data go to the FIFO to be output to the memory and, depending on conditions, may or may not be examined periodically by the trigger circuitry. Overload conditions are also sent to the control board (and from these to the memory controller) for access by the 2501 microprocessor (located on the CPU board).

The channel board setup parameters are passed to the selected boards via a serial interface on the control board. There are five control registers on each 25 kHz channel board that control parameters such as gain, coupling, bias source, and digital filter processing sequences. These control registers are write only devices and are discussed in more depth in paragraph 4.13.5.

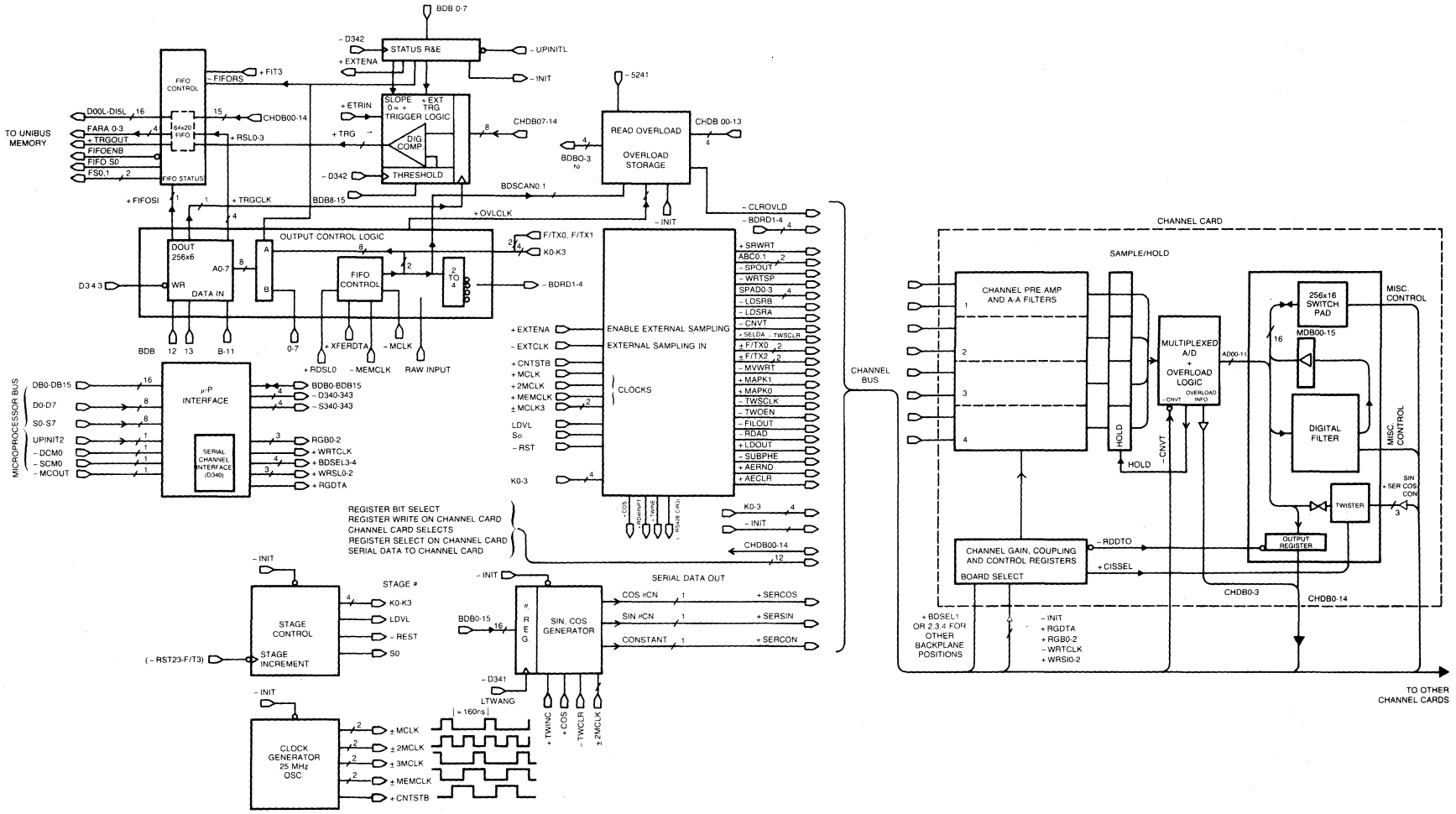


Figure 4-33. Functional Block Diagram, Channel and Control Boards

4.13.2 Channel Board Input Circuits

Each 25 kHz channel board contains four identical pre-amplifiers, A-A filters, and sample/hold circuits to process the analog input data before the analog to digital (A/D) conversion. See figure 4-34.

4.13.2.1 PreAmplifiers (U25-U28). The pre-amp is a differential type with the low side of the input returned to analog ground via a 10 ohm resistor. This aids in reducing the amplitude of the ground loop induced input noise. Full scale ranges of the pre-amp are from 8 volts to .0625 volts in a binary sequence of 8 steps. Each pre-amp is equipped with a 2 milliamp current source that is software controllable (on/off).

Each pre-amp can be programmed for four modes: AC coupled, DC coupled, CAL (zero), and PCB Bias On. The third mode, CAL, is used to measure the DC offset in the channel and subtract it out, via software, if desired. This technique eliminates the adjustments usually necessary to null DC offsets, therefore no alignment procedures are required. The subtraction of the offset is a fixed point computation and does not significantly impact any processing times.

4.13.2.2 Analog Anti-Aliasing Filter (HB 17, 20, 27, 32). The analog A-A filter is of the Caur (or eliptic) type and is most efficient, in terms of hardware, when frequency response is the prime objective. The filter is both equiripple in the pass and stop bands with a step response overshoot of approximately 20%.

The filter is implemented using a ladder structure with Frequency Dependent Negative Resistors (FDNRs) as the active elements in the filter. Phase match is within 1.5 (±.75) at the analog A-A filter cutoff. Gain matching is within .1 dB channel to channel.

4.13.2.3 Sample/Hold (U66-U68). The A-A filter outputs are fed to a group of four monolithic sample/hold circuits (U66-U68). These devices require 5 microseconds to acquire .01% (10-volt steps). All sample/hold (S/H) circuits go into HOLD simultaneously.

The outputs of the S/H circuits drive a 4-to-1 DMOS FET multiplexer. The multiplexer minimizes channel crosstalk and nonlinearities. Approximately 500 nanseconds are allowed for setting of the multiplexer after switching channels before starting the A/D conversion.

All four channels are sequentially converted every sampling interval and the results are stored into the first rank of register files. These converted values are transferred to the second set of register files near the end of the sampling interval for use by the digital filter and for transferring undigitally filtered data back to the control board. Five conversions take place during the sampling interval, however the fifth converted result is not used.

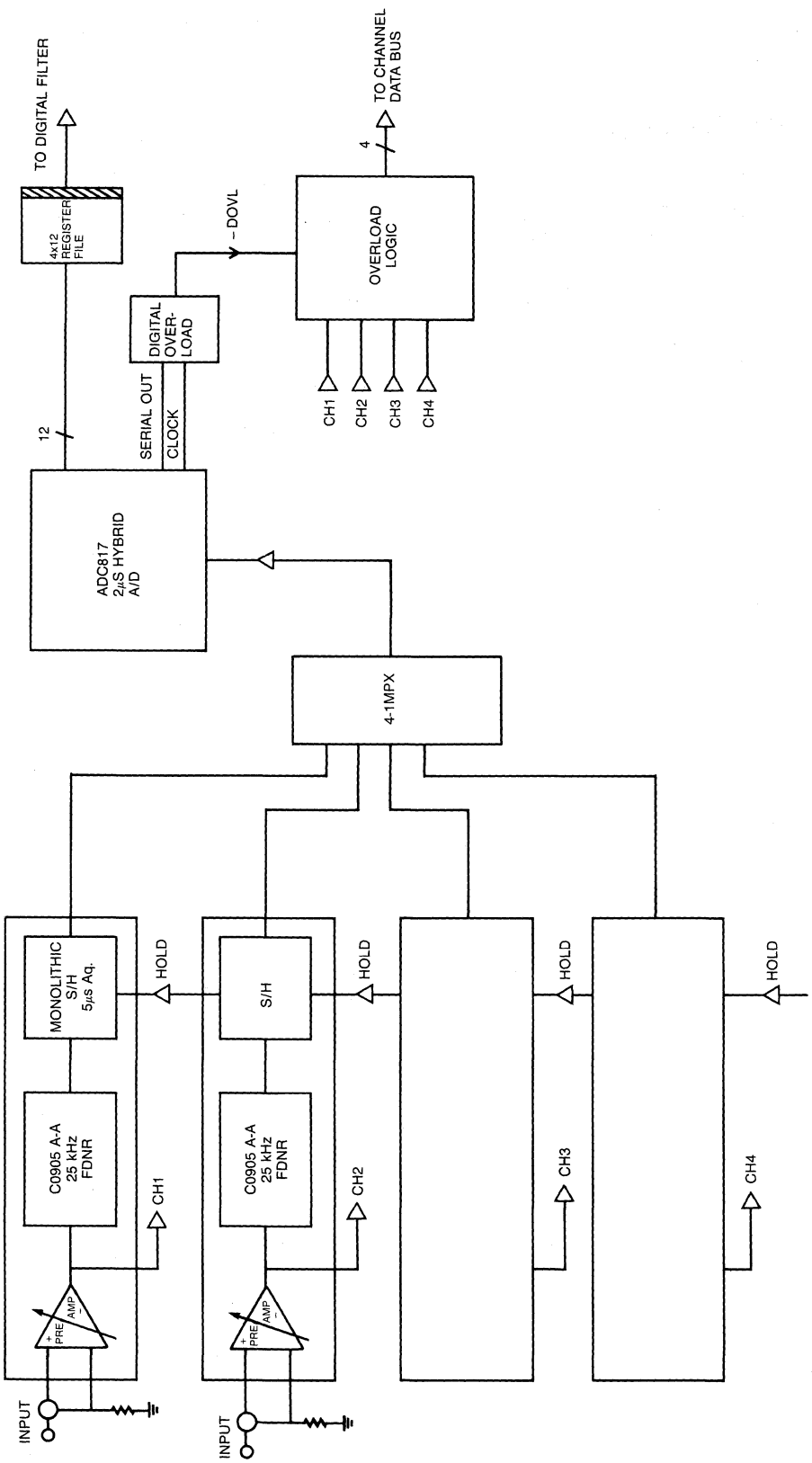


Figure 4-34. 25 kHz Input Circuitry

4.13.2.4 Overload Detection. Overload monitoring is performed at two places. The output of the pre-amp is monitored by comparators that are scanned between the channels in step with the A/D scanning. This insures that signals beyond the passband of the A-A filter do not overload it giving erroneous results. The serial digital output of the A/D converter is also monitored to detect F.S. conditions, which are flagged as overloads. Transient signals that are below the analog overload levels at the input to the filter can cause the filter output to be above the input range of the A/D and are detected here. The overload bits are read by the control board and stored there for access by the microprocessor. The overload conditions are cleared automatically upon completion of the microprocessor reading the four 4-bit words corresponding to the 4 boards and 4 channels. These conditions can also be cleared through a bit in the TRIG register.

4.13.3 Channel Board Signal Processing Sequence

The basic sampling rate of the 25 kHz system is 65.536 kHz which is a 14.25 μs interval. This interval is referred to as "T" (refer to figure 4-35. Each channel board is capable of processing 4 baseband channels or 2 Zoom channels during one 15.25 μs interval. The hardware is time shared, and consequently operates in an interval of T/4. On each channel board, there are three distinct processes happening concurrently under direction of the control board. These are listed below.

1) Sequential A/D conversion of 4 channels per board.
2) Multiplication of selected A/D results by a cissoid or constant.
3) Digitally filtering the output of the multiplier with a cascade of decimate by two filters.

This is a "pipelined" process where the results from process 1) are used by 2) and the results of 2) are used by 3).

It is assumed that the S/H circuits are in the sample or track mode previous to time 0T. At time 0T, the A/D cycle is started on all channel boards by placing all S/H circuits in HOLD mode upon receipt of the -CNVT (convert) signal from the control board. The A/D converters on all boards then sequentially convert the 4 channels on each board and temporarily store the results in the first rank of register files. The A/D converter timing is not controlled by the control board but by the converters' internal clock and by two one-shots with associated logic on the channel board. This allows the converters to have slightly different conversion speeds. At the end of the fourth conversion on each board the S/H circuits are placed back in sample (track) mode to acquire the signal for the next sequence of A/D conversions that commence at time 1T. Consequently, before the end of one sampling period all channels have been converted. Near the end of the sampling interval the four converted values are transferred to the second set of register files. This transfer occurs when the information in the second set of register files is no longer needed by the digital processing hardware.

The digital hardware processes these samples from the A/D by dividing the main sampling period T into four subperiods in which the signal processing hardware can process the converted input signals. Each of the subperiods is further divided into 24 states in which the required transfers and computations are carried out under the direction of the control board.

During the four subperiods all of the twister and digital filters on the associated channel boards are processing in the same manner but the data used can be different. The twisters function is to multiply a selected A/D results by one of three things:

1) A constant "K" for digitally filtered baseband
2) K*Cos (THETAC*N) where THETAC=zoom center frequency
   N = sample number
3) K*Si(THETAC*N)

For each of the four subperiods (0, 1, 2, 3), the twister will multiply the software selectable A/D data by either the constant or the cissoid. When one selects the constant this implies that the desired results will be a baseband sequence. When the cissoid is selected, in subperiods 0 and 2 the data is multiplied by the cosine function whereas in 1 and 3 it is the sine function. For instance, if the channel board is set to multiply by the cissoid in subperiod 0 the selected data from the A/D register file will be multiplied by the cosine function.

Time state 1T→2T serves as an example of baseband processing where during subperiod 0 the converted data on channels 1, 5, 9, 13 (assuming all channel boards are set up the same) is multipled by K. These results are automatically transferred to the input of stage 0 of the cascaded digital filter. Likewise in subperiods 1, 2, 3 the remaining twelve channels of data (assuming one had that many channel boards) would be multiplied by K and fed into the input of the digital filters.

Time state 2T→3T is an example of zoom and baseband processing. Considering just the first channel board, channel 1 is multiplied in subperiod 0 by K*Cos(THETAC*N), labeled cos in 4-35, and the same data is multiplied by K*Sin(THETAC*N) in subperiod 1. It would then be possible to multiply any other channel on this board by the cos, sine functions in subperiods 2, 3 to give two channel zoom. However, for this example the data from channel 1 is multiplied in subperiod 2 by the constant K and the data from channel two is also multiplied by K in subperiod three.

This demonstrates how the processing of data can be manipulated to provide all baseband, all zoom, or mixed zoom and baseband.

Each board can be set up independently of the others for this processing, however, A/D data generated on one channel board cannot be fed into the digital section of another channel board.

The results of the previous multiplies are fed into the input of stage 0 of the digital filter which is operating concurrently with the twister but utilizing the previous set of outputs from it. The digital filter is a decimate-by-2 design which means that for every two input points, one output is generated. This design is effectively cascaded so that if the first filter operation takes place on the data from the twister and is stage 0, the output of stage 0 is fed into stage 1 and so on. Through each successive filtering operation the sampling rate (data rate) is reduced by a factor of 2 and so is the bandwidth of the data.

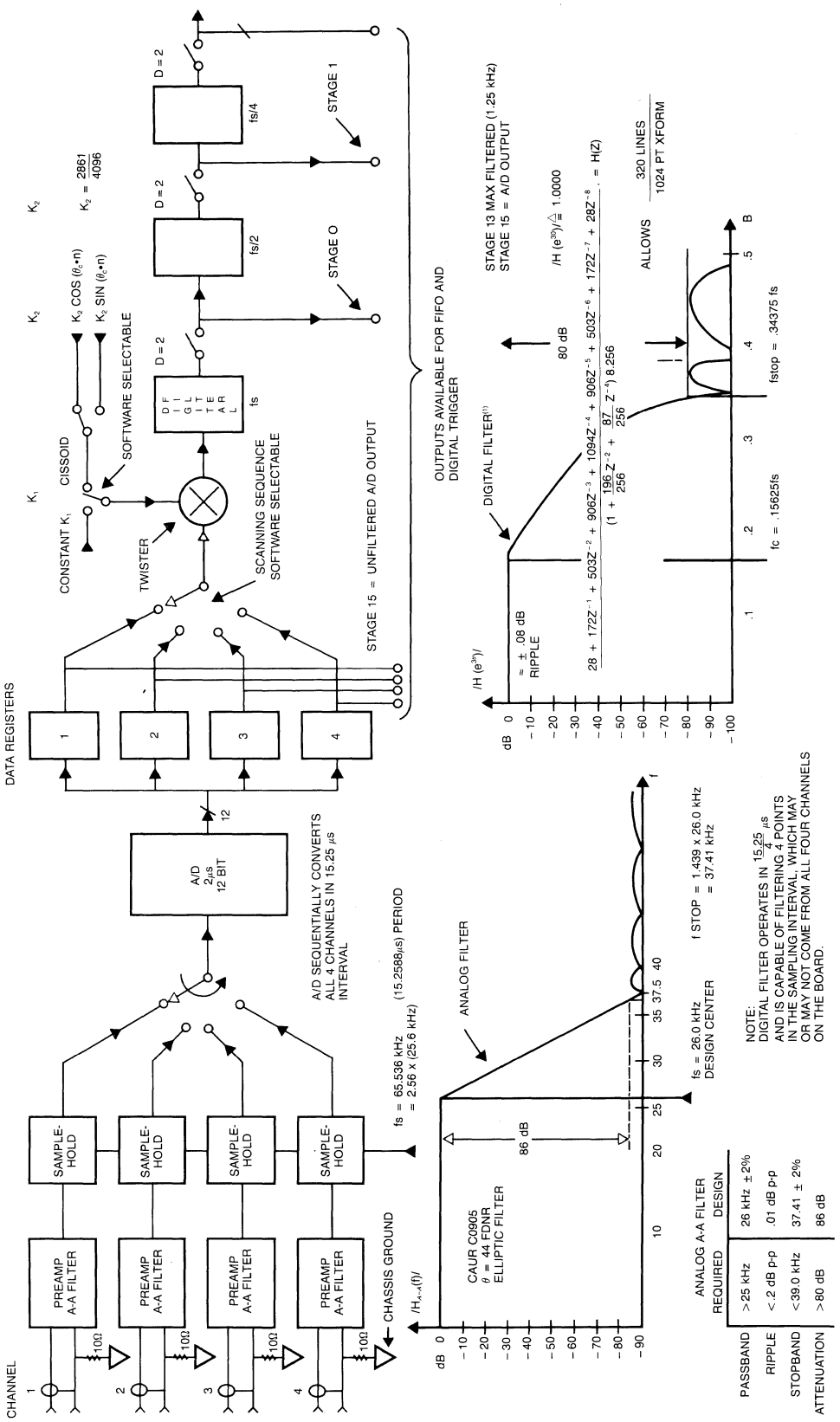


Figure 4-35. 25 kHz Analog-Digital Filtering

Since the filter takes two input points and produces one output point per computation, the effective computation rate of the 0th (first stage after twister) is Fs/2 where Fs=4/T, T being the main sampling interval. Remember, the digital filter must go 4 times as fast as the main sampling rate. The "1st" stage operates at a computation rate of Fs/4, the 2nd Fs/8, etc. The interval required computation rate is therfore Fs/2+Fs/4+Fs/8+Fs/16+Fs. As the number of stages of filtering is increased the total computation rate approaches Fs. The stages of digital filtering are time sequenced by the control board to provide the required filter computations when needed. From the previous analysis it is clear that the filter processsing hardware must operate at the rate Fs but that half of the time the filter is operating on stage 0 and the other half is spent on all subsequent stages. Therefore, not only is the filter hardware shared among the channels on the board but it is shared among all the stages in the overall filtering process.

4.13.4 Channel Board Digital Processing

The following text discusses the digital processing function and related hardware. Refer to figure 4-36.

4.13.4.1 Twister. A serial multiplier is used to multiply the data from the A/D channel selected by the software setup of the channel board by the appropriate multiplicand be it a constant, sine, or cosine. The multiplier chips, U135 and U129, are parallel loaded by the MDB (major Data Bus) with AD0-AD11 from U97-U99, the second set of A/D register files. The multiplicand input is one of three serial bit streams provided by the control board. These streams correspond to the constant K, K*Sin(THETAC*N) and K*Cos(THETAC*N).

The result emanates serially from the multiplier chips, U135 and U129, and is stored in shift register U123 and U121. The serial output forms the LSB first and is equal in length to the sum of 12 bits in the multiplier and 16 bits in the multiplicand. However, only the most significant 16 bits are retained and the magnitude of this number is always less than 5.

The multiply takes 29 clocks to compute. The clock takes only 80 nanoseconds to accomplish the multiply at the same time as the digital filter computes an output point. The result of the multiply is output onto the MDB and deposited into the scratchpad memory (U115, U120, U125, U110). This output is used by stage 0 of the digital filter.

All timing and control signals for the twister originate from the control board.

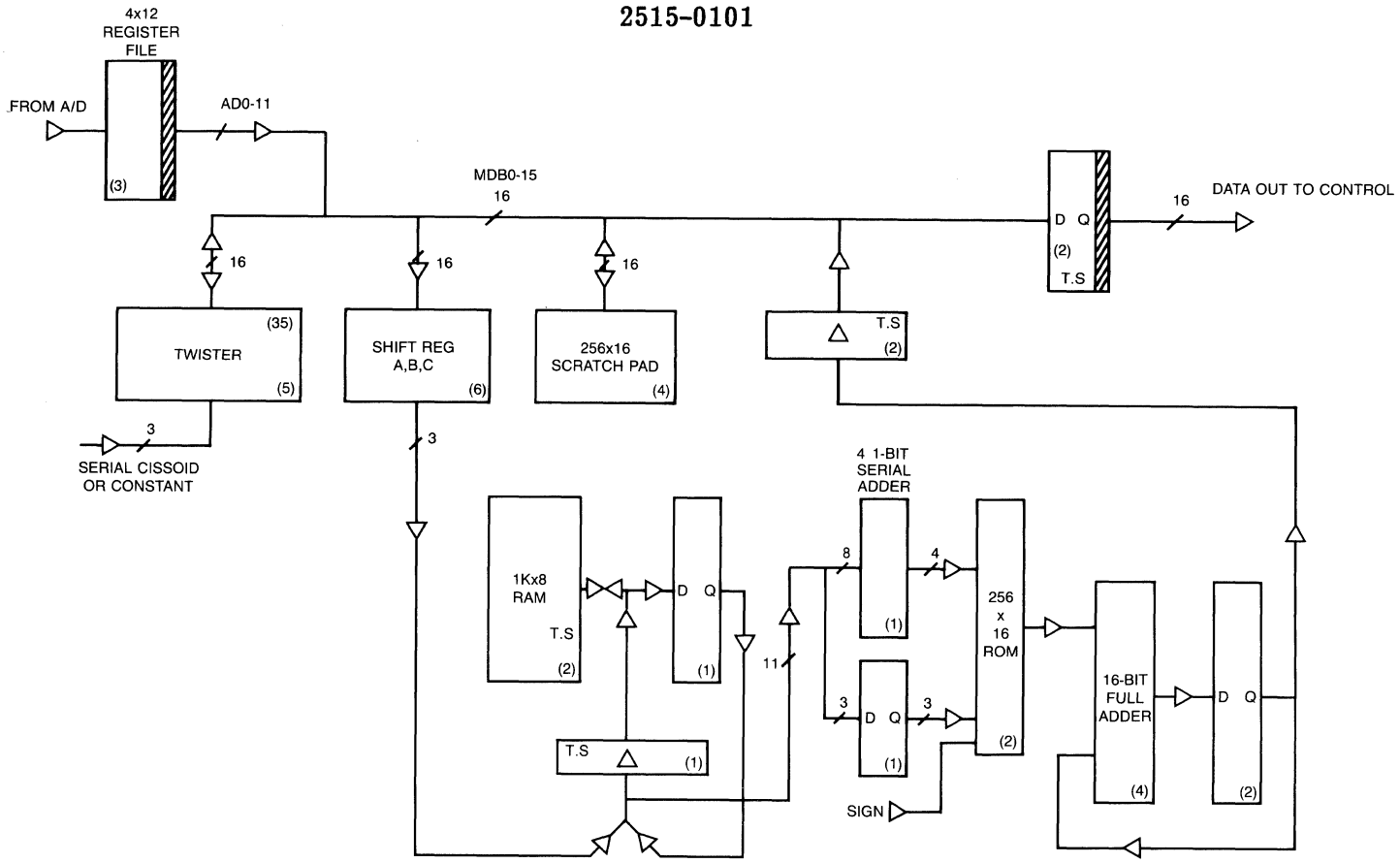


Figure 4-36. Digital Filtering

4.13.4.2 Digital Filter. The digital filtering structure needs no multipliers to compute an arithmetic statement basically composed of a sum of multiplied inputs and outputs. Each filter circuit goes through the same operations to compute outputs, however the data utilized are different. For example, the filter computation that occurs during subperiod 0 on channel board #1 could be data from channel 1 that was multiplied by a constant by the twister if stage 0 of the filter was being computed. Stage 1 of the filter always gets its inputs from stage 0, stage 2 gets inputs from stage 1, and so on. Consequently, when computing stage 1, one is processing the output of stage 0.

The filter is a decimating type designed to lower the bandwidth and sample rate factors of 2 through successive stages.

The filter decimation is accomplished by the data path organization of the shift registers holding the input data (X(n), X(n-1), X(n-2), etcetera),. The total length of the shift register is 9 x 16 bits. The filter requires two inputs, such as X(n) and X(n-1), to produce one output, Y(n). The first two shift registers in the SRA and SRB chain are parallel in/serial out registers which are loaded from the MDB (channel bus) with data from the scratch pad RAMs. The data contains the outputs from previous stages or the twister outputs. The previously computed filter outout is also loaded from the scratchpad RAM into shift register SRC, another 16-bit parallel in/serial out device.

The remaining shift registers are serial in/serial out registers implemented by a RAM and counter arrangement. The RAM stores the data for the other three channels and the other 15 stages (of which up to stage 13 are usable). This scratchpad RAM provides temporary data storage for all channels and stages on the board. It is necessary to pull three words out of the RAM memory and load them into three shift registers before the filter computation can begin.

Once set up, the filter requires 17 clocks to complete the computation. The data shifts by two during the computation leaving two inputs empty, so to speak, and ready to receive two new inputs.

The shift register serial outputs are sent. LSB first to the serial address to perform the required addtion. The resulting serial streams are fed into a PROM along with the remaining variables.

Upon completion of the computation, the output is written back to the RAM location corresponding to the output data and also the location corresponding to the input of the next stage. The information needed to indicate which pair of inputs initiate the next stage comes from the stage control circuitry on the control board.

The total number of clock cycles to completely acquire data (2 clocks), compute (17 clocks), and deposit data (3 clocks) is 22. Since there are 24 cycles to each subperiod, 2 clock cycles are left as spares.

4.13.4.3 Available Bandwidths. Refer to table 4-13 for data on bandwidths available for the 25 kHz input channel. These bandwidths are true for either translated (zoomed) or untranslated data. Some of the data are dependent on frame size. The numbers provided are based on 1K frames.

Various bandwidths are selected by choosing stage outputs from the digital filter. The 25 kHz band is obtained directly from the A/D converter and all lower frequency bands are from selected stages of the digital filter. More than one band may be selected on the digital filter at one time. Additionally, different channels may have different bandwidths at the same time.

4.13.4.4 Signal Processing Gain. If the 25 kHz bandwidth is selected the gain will be .8, that is, on the .25 volt range, .5 volts will generate 80% of full scale or (.8) (215) output of the A/D. If one selects the 10.24 kHz range and down, the A/D output is multiplied by K, and the signal is attenuated. As the bandwidth is progressively lowered by passing through more filter circuits the filter gain stays at unity with a ripple amplitude of .11 dB maximum.

When zooming, the data is multiplied by sine and cosine functions with ampitude K to prevent filter overloads. If no overload is detected the data is valid.

These gains must be considered when using the data and setting trigger levels as the trigger circuitry operates on digital data from the channel boards rather than analog data.

Table 4-13. 25 kHz Input Channel, Available Bandwidths

Band	Decimation	Stage No.	Usable Lines	Resolution	Frame Time	Zoom C.F. Resolution	Sample Frequency
24.96 kHz	None	15	390	64 Hz	15.62 ms	N/A	65.536 kHz
25.024 kHz	None	15	291	64 Hz	15.625 ms	N/A	65.536 kHz
10.24 kHz	2	0	320	32 Hz	31.25 ms	32 Hz	32.768 kHz
5.12 kHz	4	1	320	16 Hz	62.5 ms	16 Hz	16.384 kHz
2.6 kHz	8	2	320	8 Hz	125 ms	8 Hz	8.192 kHz
1.28 kHz	16	3	320	4 Hz	.25 sec	4 Hz	4.096 kHz
640 Hz	32	4	320	2 Hz	.5 sec	2 Hz	2.048 kHz
320 Hz	64	5	320	1 Hz	1 sec	1 Hz	1.024 kHz
160 Hz	128	6	320	.5 Hz	2 sec	1 Hz	5512.0 Hz
80 Hz	256	7	320	.25 Hz	4 sec	1 Hz	256.0 Hz
40 Hz	512	8	320	.125 Hz	8 sec	1 Hz	128.0 Hz
20 Hz	1024	9	320	62.5 MHz	16 sec	1 Hz	64.0 Hz
10 Hz	2408	10	320	31.25 MHz	32 sec	1 Hz	32.0 Hz
5 Hz	4096	11	320	15.625 MHz	64 sec	1 Hz	16.0 Hz
2.5 Hz	8192	12	320	2.8125 MHz	128 sec	1 Hz	8.0 Hz
1.25 Hz	16384	13	320	2.90625 MHz	256 sec	1 Hz	4.0 Hz

4.13.4.5 **Triggering.** Triggering occurs using all digital circuitry that can be sourced with data from the channel boards under the direction of the OUTSEL matrix. The trigger circuitry has hysterisis and slope. It has the ability to observe, or process, data that is not necessarily being put into memory and can observe more than one channel at a time. This is similar to an ORing function where any channel, or switching between channels, can cause a trigger.

The trigger signal, generated on the control board, is taken from the FIFO along with channel data and access initiator information by the memory controller. If the memory controller's trigger interrupt enable bit is set and a trigger signal is detected, an interrupt will occur. The memory controller will then cease loading FIFO data into memory on completion of depositing the data that was pulled concurrently with the trigger.

4.13.4.6 Output Data Format. The output data is a 15-bit, two's complement; left justified word that is DMA'ed into memory under the direction of the control board. Since 80% of the A/D ranges are used, this is equivalent to saying that on the 8 volt range and undigitally filter data, 8 volts correspond to -8/10*32768. The output can go to -32767 and still be valid if no overload is detected. On the filtered ranges, these numbers are scaled by 2861/4096. The full scale ranges are:

1) RAW DATA (no digital filtering) -32768 ^N<+32766
2) Processed Data -22888 <N<22888.

If, on the 8 volt range, a value of 16,000 is read out of memory and the data is from the unprocessed A/D, then it has a value of 4.8828 volts. If it is from the digital filter it has a value of 6.905 volts.

4.13.4.7 Data Rate and FIFO Depth. The observed data rate is a function of the active channels and the bandwidths and is 15.258 μs.

The control board FIFO depth is 64 words of which only 48 are usable. The other 16 words make up the FIFO buffers. The priority of the FIFO's access to memory is a variable. The FIFO will dominate the memory if it becomes more than 3/4 full.

An overload bit is read back (see paragraph 4.13.5.6) to detect if the FIFO has been overrun.

4.13.5 Register Descriptions

The following text and related illustrations describe register functions and circuitry of the channel and control boards. Because of the closely interrelated activities the boards will be described together rather than separately except where necessary.

Initialize, control data, and source/destination codes are sent from the CPU board to the control board via the microprocessor interface. The control board utilizes the data and codes to set up the hardware into the various required modes.

The following modes are set by data, three sources and four destination codes.

Destination Code:

D340: ADSET controls

1) Channel preamp gains, couplings, bias source, and input zeroing.
2) Multiplication of input data by cissoid for zooming.
3) Digital filter source data (channel multiplexing into filter).

D341: ZANGLE CONTROLS

1) Zoom center frequency setting.

D342: TRIG controls

1) Trigger threshold, slope.
2) External Sampling, triggering, FIFO reset, channel clear, twister clear, overload clear and system initialize.

D343: OUTSEL controls

1) Selection of points to be loaded in FIFO.
2) Memory controller acces initiator selection.
3) Selection of points for digital trigger.
4) A synchronization bit.

Source Code:

S340: ADSET reads

1) Most significant 7 bits back from ADSET plus a busy bit.

S341: OVLD reads channel overload data, and FIFO overload.

S342: TWTST

1) Provides a bit to aid in loading the Zoom center frequency while acquiring.

4.13.5.1 D340, ADSET.

15	14	13	12	11	10	9	8
BUSY	BDSEL4	BDSEL3	BDSEL2	BDSEL1	WRSL2	WRSL1	WRSL0
	— Selects Channel Card —				—Selects Register—		

Sixteen bits of data are sent to the control board from the CPU board map register circuit via the microprocessor interface. The most significant bits 8-15 are used for the D340 ADSET register (see sheet 7 of 2515-4701-2D).

Each channel board has five 8-bit registers (U14, U24, U29, U55, U69) used to set up its processing parameters (see sheet 4 of 2515-4700-2D). Register selection is controlled by bits 8, 9 and 10 (WRSL0,1,2) as indicated in the selection below.

Table 4-14. ADSET Register Selection

WRSL Bit No.			Register Selected
2	1	0	
0	0	0	U24
0	0	1	U29
0	1	0	U69
1	0	0	U14
0	1	1	U55

Board Selection is controlled by bits 11-14. The data to be loaded is in the 8 LSBs, bits 0-7, of the D340 word. Since four bits are available to select the board(s) to be loaded, one or more boards may be loaded with the same information at the same time.

The write to the selected channel board register is automatically initiated when D340 is loaded from the microprocessor. Source code S340 allows a readback of the 8 MSBs of the D340 word. MSB bit 15 is used to indicate that the serial interface is busy loading the data (LSBs 0-7) into the seleced boards and registers.

The serial interface takes approximately 1.2 microseconds to transfer the 8-bit word to the selected register.

Note: For ease of schematic referencing all 25 kHz channels will be referred to as channels 1-4 on each board. So if channel 6 is to be addressed, it will be referred to as channel 2 on board 2.

Register U24

7	6	5	4	3	2	1	0
+2AC	2AV2	2AV1	2AV0	+1AC	1AV2	1AV1	1AV0

Register U29

7	6	5	4	3	2	1	0
+2AC	2AV2	2AV1	2AV0	+1AC	1AV2	1AV1	1AV0

Volts Full Scale	Preamp AV2	Gain AV1	Selection AV0
8.	0	0	0
4.	0	0	1
2.	0	1	0
1.	0	1	1
.5	1	0	0
.25	1	0	1
.125	1	1	0
.0625	1	1	1

Registers U24 and U29 control the gain and coupling of channels 1-4 on a board. For example, bit 7, +2AC, controls the AC coupling on channel 2 of a selected board. If this bit is a logic 1 the channel is AC coupled. The AV bits (0-2, 4-6) select one of three available gains on the four channels. In this case, 3AV2, 3AV1, and 3AV0 control channel 3's gain setting again on the boards selected by bits 11 through 14 of the D340 word.

Register U14

7	6	5	4	3	2	1	0
ION4	ION3	ION2	ION1	ZERO4	ZERO3	ZERO2	ZERO1

Register U14 controls the 2MA current sources and the input zeroing modes. Bits0-3 enable the input zeroing feature of the hardware. This can be used to measure the dc offset of the channels. The value obtained can be used to cancel the existing dc bias.

Bits 4-7 enable the 2MA current sources which are connected to the channel inputs. These are used to provide the bias current to the accelerometers. When this mode is enabled the ac coupling to the channel must also be activated to prevent the large dc bias present from overloading the channel. The ION and zero bits are active true.

Register U69

7	6	5	4	3	2	1	0
Sin3	Cos2	Sin1	Cos0	not used	not used	not used	+DTHR

Register U55

7	6	5	4	3	2	1	0
MS3	LS3	MS2	LS2	MS1	LS1	MS0	LS0

Registers U69 and U55 are concerned with how the data from the A/D is processed by the digital filter and twister. As stated earlier in this paragraph, the channel board, or digital filter can process four baseband channels or two zoom channels, or one zoom channel and two baseband channels in step with the input data.

To allow this flexibility it is necessary to supply two pieces of information for each subperiod the filter operates in: 1) a decision to multiply the input data by cissoid (for zooming) or not, and 2) which channel to be processed during each subperiod. U69 and U55 define those decisions.

If SIN3 bit (bit 7 of register U69) is a 1 it means that during subperiod 3 (SP3) of the process the data will be multiplied by the K*Sin value from the lookup table. If the Cos0 bit is a 1 then the data fed into the twister during subperiod 0 will be multiplied by the K*Cos value. A choice can be made to multiply data by a sine or cosine function by selecting the appropriate bits.

When doing two zoom channels all 1's will be in the sine, cosine bits. When doing all baseband all zeros will be in the sine, cosine bits. When a zero is present, the data from the A/D register is multiplied by the constant K.

A dither source signal of approximately 40 dB below full sale (at kHz) tends to improve any sensed nonlinearity of the A/D converter. When the +DTHR bit is set, the signal goes to all four channels.

Register U55 controls what channel is processed during each subperiod SP 0-3. The four channels per input boards are by two bits, MS and LS. Channel #1 is 00, and channel #3 is 10, and so on. To use the register, insert the appropriate channel identification bits into the MS/LS0-3 and that channel will be processed by the digital filter during the subperiod dictated by the number following the MS/LS designators.

Note the following examples.

Example 1: Baseband on all four channels, normal processing order 1,2,3,4 in SP0,1,2,3

```
WRSL
011     11100100    LSB

010     0000xxxx
```

Example 2: Zoom on channels 2 and 3 processing 2 during SP0, 1 and 3 during SP2, 3

```
011     10100101
010     1111xxxx
```

Example 3: Zoom on channel 3, baseband on 1 and 3 process zoom data in SP2,3 and baseband channel 3 in SP0, and 1 in SP1

```
011     10100010
010     1100xxxx
```

4.13.5.2 D341, ZANGLE. The D341 ZANGLE register sets the center frequency for the zoom mode of processing. This register is double buffered so that processing need not stop in order to change the angle. This can be useful if one is using the sine table to feed the D/A on the utility board to generate swept sine functions. After the microprocessor loads D341 with the desired value it is transferred to the angle increment register of the sine generator synchronized by the sample clock.

Bit 15 in TWTST S342 can be tested as follows to eliminate the possibility of the microprocessor changing ZANGLE at a time that could cause a one sample glitch in the sine table output.

1) wait for bit to be a 1
2) immediately load ZANGL

15	14	13	12	11	10	9	8
15							
2							

7	6	5	4	3	2	1	0
				2	1	0	
				2	2	2	

This is directly calibrated in Hz for the 25 kHz system and for 100 kHz the LSB is 4 Hz.

4.13.5.3 D342, TRIG.

15	14	13	12	11	10	9	8
Sign							LSB
(- - -			Two's complement threshold				- - -)

0 = + slope, for trigger

7	6	5	4	3	2	1	0
	+	-	+		-	+	+
SLOPE	EXT SAMP	FIFO RESET	EXT TRIG	ANG CLR	OVL CLR	CHNL INIT	RUN (-INIT)

Bits 7-15. The TRIG register's main function is to supply threshold and slope information to the digital trigger circuitry along with control bits for the boards.

The trigger threshold is programmed by the upper 8 bits (bits 7-15) of the TRIG register. The full scale range for analog data is -102 to +102 and ± 71 for digital full scale range. To obtain a threshold for a percentage of full scale for RAW DATA (analog filtered data) compute P%*102 + V ofs. To compute the threshold for digitally filtered data, use P%*71 + V ofs. The outcome is loaded into the upper 8 bits of the TRIG register in twos complement format to provide the required threshold and slope.

Bit 6, +EXT SAMP. This bit enables the external sampling to the control board circuits. The circuits can be driven by an external sample clock when the sampling is to be synchronized to an external event. The sampling creates a short delay after a negative transition occurs on the -EXT input. The duration of the pulse is a minimum of 50 nanoseconds.

Bit 5, -FIFO RESET. This bit provides two functions. First, it allows processing of input data (if bit 0 is set to a logic 10 while not putting the results immediately into the FIFO. It holds off the loading of selected data while transients are settling out. The amount of time required for settling is a function of the minimum bandwidth utilized. For instance, if the minimum bandwidth selected is 320 Hz then the required time is 50 milliseconds.

The second function of this bit is to change control board multiplexers. This allows the user to change memory contents that determine which filter data get fed into the FIFO, and which are fed into the digital trigger circuitry. It is here that the assignments of which data streams get put into the various sections of memory are made. This aids in sorting the data from different stages (more than one bandwidth).

When the -FIFO Reset bit is set high, further changes to the OUSEL registers (paragraph 4.13.5.4) are disabled and loading of the FIFO will begin upon sensing a SYNC bit (bit 14 of the OUTSEL register). The addition of the SYNC bit allows loading the FIFO at a predetermined state and permits unscrambling data from multiple stages using a single access initiator.

Bit 4, EXT TRIG. This bit is a logic 1 and enables the external trigger input to the circuit. A TTL input, its negative edge is the trigger event.

Bit 3, -ANGCLR. The TWISTER Sin/Cos lookup address may be set to a known state, 0. This bit being low forces the outputs of the sin/cos generator to be set to K* Cos or K* Sin.

Bit2, -OVLCLR. This bit clears all overload registers on all channels. It is active low and also clears the bit used to detect a FIFO overload.

Bit 1, +CHINIT. This bit sets all channel board registers to 0 (bit 0 does not do this). The microprocessor INIT also sets all registers accessed through D340 ADSET to a zero state.

Bit 0, +RUN (-INIT). This bit initializes the digital control board circuits to a known state (a logic 0). In doing this, it stops the internal clock from operating therefore rendering the serial I/O to the channel board registers inoperative. It does not, however, change the state of the channel board registers nor does it change the states of any other registers.

A logic 1 must be loaded into the RUN bit in order to communicate any parameter changes to the channel board registers accessed by D340 ADSET. When the +RUN bit is set to a logic 1, the internal clock is started. If the circuit is not set to accept external sampling inputs, it begins sampling and processing data in the manner dictated by the setting of the control board registers. If the -FIFO RESET bit is a logic 0, the processed data is not passed to the FIFO; it is, instead, ignored. If the FIFO RESET bit is a logic 1 and the SYNC conditions (specified in D343 OUTSEL) are met, the data is put into the FIFO and triggered under the direction of the OUTSEL matrix.

The advantage of simultaneously starting the FIFO along with the processing is that if the FIFO is being loaded with outputs from multiple stages, and does not use a different access initiator for each bandwidth, the data can be decoded since the circuitry starts in a totally known state. Once initialized, the FIFO RESET bit controls the data stream from the filter, allowing modifications to the setup of the processing parameters.

4.13.5.4 D343, OUTSEL.

15	14	13	12	11	10	9	8
	+	+	+				
RAW INPUT	SYNC	FIFO POINT	TRIGGER POINT	RSL 3	RSL 2	RSL 1	RSL 0

— Memory Controller access initiator selection —

Send points to trigger if a 1

Send points to FIFO if a 1

Enable FIFO filling

7	6	5	4	3	2	1	0
				FLT	FLT	BD	BD
K3	K2	K1	K0	X1	X2	SCAN1	SCAN0

— Stage selection 0-13 for filtered, 15 for RAW input —

— Filter Subperiod Select —

— Channel Board Select

The OUTSEL register is a selection matrix that is setup by software to control which data form the channel boards gets put into the FIFO along with an appropriate access initiator tag and whether or not the data gets fed into the trigger circuitry.

It is a sieve that filters out desired sources of data from the 15 to 16 possible choices is defined by bits 0-7 of the register. This register is a RAM that can be written to only when the FIFO RESET bit (bit 5 of D342 TRIG) is low.

This prevents inadvertent changes in the middle of loading the FIFO. The upper eight bits should be zeroed upon initialization. The upper eight bits, of which only 7 are used, are fed to the RAM to enable outputs to the trigger and FIFO. The lower eight bits are the RAM address. When the FIFO RESET bit is low all transfers to the FIFO cease, and the FIFO data is purged. The address bits of the RAM are connected via a multiplexer to the lower 8 bits of D342 while the data into the RAM is connected to bits 8-13.

Each source of data to the trigger and/or FIFO is activated by writing the appropriate information to D343.

Which channels go into the FIFO and/or trigger are determined by bits 0-3. The selection of which channel board(s) is controlled is done by bits 0 and 1 (BDSCAN0 and BDSCAN1). Once board selection has been established, the next two bits, 2 and 3 (FLT X1 and FLT X2) select which filter cycle (subperiod) is to be output. The channels that are output are a function of how the registers on each channel board are configured. These registers control the data processed during the four sub-periods into the twister/digital filter.

Bits 4-7, K0-3 specify from which stage (bandwdith) of filtering the data to the FIFO/trigger is taken.

OUTPUT OF STAGE	7 K3	6 K2	5 K1	4 K0	DECIMATION FACTOR	
0	0	0	0	0	2	
1	0	0	0	1	4	
2	0	0	1	0	8	
:		:			:	
:		:			:	
:		:			:	
13	1	1	0	1	16,384	
14		ILLEGAL			*	
15	1	1	1	1	1	UNdigitally filtered data

Bits 8-11, RSL0-3, specify which access initiator is to be used with the data. This feature allows outputs from multiple stages to use separate AIs which will not automatically sort the data into appropriate buffers. Separate AIs are used to sort various channels into separate buffers. The data from a common stage and different channels is interleaved and is handled in that format.

Remember that there are 4 subperiods in which computations take place on all channel boards simultaneously.

Upon completion of the computations during subperiod 0, all four channel boards (if present) may or may not have data to be picked up by the FIFO simultaneously. If directed by the OUTSEL matrix these four pieces of data are loaded into the FIFO in rapid succession. So, if the channel boards were set up to sequence through four channels on each board for baseband processing, the order of the channels in the FIFO would be 1, 5, 9, 13, channel 1 being first.

The overall order is dependent on the order of channel board processing. That is, the first channel of each board during subperiod 0, the second channel of each board during subperiod 1, etc. This is referred to as the baseband normal order (BNO).

For 16 channels in BNO the data in memory would appear as

1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16, 1 5 9 13, etcetera.

Deleting channels from the sequence does not change the order of the remaining channels. For instance, doing 8 channels in BNO would appear as

1 5 2 6 3 7 4 8, 1 5 2 6 3 7 4 8, ... etcetera.

Bits 12 and 13 dictate if the source of data (channel, stage) is to be put into the FIFO or trigger circuitry. If bit 12 (+ TRIGGER POINT) is a logic, the data is passed into the trigger circuitry. If bit 13 is a logic 1, the data is put into the FIFO with a flag on it corresponding to the access initiator selected by bits 8-11.

Bit 14 is a SYNC bit that was added facilitate sorting data that is from multiple stages using single access initiator. Data going into the FIFO and trigger are held off until the occurance of the SYNC condition.

If separate access initiators are always used for separate stages the SYNC bit is set permanently to a logic 1; that is, for each selection in the OUTSEL matrix, also set the SYNC bit.

It is recommended that to use the SYNC bit on stages 0-13 the bit should be set to the desired stage K, subperiod 0 (FLTX 1, 2=0) and board 0. For stage 15, RAW INPUT, synchronize on stage 0.

Refer to the OUTSEL matrix in table 4-15.

Table 4-15. OUTSEL Matrix

	+													
	T													
	R													
	+ I													
+	F	G												
S	I	G										B	B	
Y	F	E										D	D	
N	O	R										S	S	
C	P	P	R	R	R	R			F	F	C	C		
	O	O	S	S	S	S			L	L	A	A		
	I	I	L	L	L	L			T	T	N	N		
	N	N					K	K	K	K	X	X		
	T	T	3	2	1	0	3	2	1	0	1	0	1	0

0	0														No trigger, no FIFO
0	1														Point to trigger not to FIFO
1	0														Point to FIFO not to trigger
1	1														Point to FIFO and trigger
		0	0	0	0										AI0
		0	0	0	1										AI1
				*											
				*											
				*											
						0	0	0	0						Output from stage 0 D=2
						0	0	0	1						Output from stage 1 D=4
						1	1	0	1						Output from stage 13 D=16, 384
						1	1	1	1						Output unfiltered A/D path
										0	0				Subperiod zero
										0	1				Subperiod one
										1	0				Subperiod two
										1	1				Subperiod three
												0	0		Board 1 (channels 1-4)
												0	1		2 (channels 5-8)
												1	0		3 (channels 9-12)
												1	1		4 (channels 13-16)
1	X	X	X	X	X	X	-	-	-	-	0	0	0	0	Sync on specified stage
															Xs don't care.

4.13.5.5 S340, ADSET. This source code, S340, reads back the upper seven bits, 8-14, of the ADSET register. The main use, however, is the BUSY bit 15 that can be tested to ascertain that the serial interface has completed its transfer to the channel boards.

4.13.5.6 S341, OVLD.

4	3	2	1	0
+FIFO OVLD	OVL4	OVL3	OVL2	OVL1

This register can be read back to determine the condition of the input circuits and if the FIFO capacity has been exceeded on all four channel boards. When first read back, the register contents represent the overload conditions on channel board 1 (channels 1-4). The OVL0 is the channel 1 overload bit. When read back successively, it (on the second reading) gives the overload conditons on channel board 2, channels 5-8. After the fourth reading, the register automatically clears all overload registers and resets back to channel board 1. The FIFO OVLD bit is valid for all readings since it is channel independent. It is also cleared following the fourth reading.

Bits 0-3, OVL1-4, are read back as logic 1s if no channel board is present. All missing boards will be represented as all channels overloaded during a read back. If the boards are set up for zeroing and the overload bits are then read back, the quantity of channels present in the system can be determined.

4.13.5.7 S342, TWTST. The only valid bit in this code is bit 15. This bit is used to determine when to load the ZANGLE register. Refer to paragraph 4.13.5.2.

SECTION V

DIAGNOSTICS

5.1 INTRODUCTION

This section provides descriptions of the diagnostic programs, instructions for their operation, and troubleshooting hints to aid in detection of problems and isolation so far as possible to a particular unit or circuit board. Also included is a description of the built in octal debugging tool (ODT) functions that can be used with the diagnostics. Maintenance is based on replacement of defective boards or units. The diagnostic programs exercise and test both GenRad and vendor units in the system. The diagnostic programs have been divided into the following groups:

- Microcoded Diagnostics,
- RT-11 Diagnostics.
- XXDP+ Diagnostics,

The microcoded programs (described in paragraph 5.4) are self contained in the first 2K region in PROM, and are used to test the CPU board, PROM board, Memory Controller board, Display board and keyboard.

The RT-11 based programs (described in paragraphs 5.5 through 5.7) are used to test the analog to digital front end, floppy disk drive, winchester drive, front panel, and the utility board. The RT-11 based programs are written in either C or assembly language, both of which are callable from the RT-11 operating system. The RT-11 programs are contained on one dual density diskette, part number 2515-0691.

The XXDP+ programs (described in paragraphs 5.8 through 5.12) are used to test the LSI-11/23, Unibus memory and DLV11 Interface and other devices on the LSI-11 Q-bus. The XXDP+ programs presented in this manual are a modified subset of the XXDP+ diagnostic package supplied by the Digital Equipment Corporation (DEC). The XXDP+ programs are contained on one double density diskette, part number 2515-0690. For a description of the complete XXDP+ package refer to the DEC XXDP+ system users manual (product name: AC-F348E-MC).

5.2 REQUIRED TEST EQUIPMENT

The following equipment or its equivalent is required:

Nomenclature	Manufacturer	Model
Precision dc Voltage Source	Dial-A-Source	DAS 46A
Digital Voltmeter	Dana	4300
Oscilloscope	Philip	PM 3260E
Alignment Tool		
Circuit Board Extender.		

5.3 SOFTWARE REQUIRED

Diganostic diskettes #2515-0691 (RT-11), and #2515-0690 (XXDP+) and a Dysan alignment diskette #802020 (which must be ordered directly from Dysan, Santa Clara, Calif.) if the floppies are to be aligned.

5.4 MICROCODE DIAGNOSTICS

There are two types of microcode test programs contained within the first 2K of PROM. First there is the all fast test sequence that executes automatically on power up. Then, there is the user selectable test program. A description on both types is provided below.

5.4.1 All Fast Test Sequence

The all fast test sequence is executed automatically upon power up and whenever the system is initialized. If the all fast test sequence should fail, the system can not be booted. The all fast test sequentially executes the following tests:

- CPU (2501) Test
- Scratchpad RAM Test
- Display Test Fast
- Memory Controller (2901 port only) Test

For a description of these individual tests, refer to paragraph 5.4.2.3. Should any of these tests fail, the all fast test sequence halts and attempts to display an error code message on the CRT. The display format of the error code message is six octal digits (sixteen bits binary) which provide the following information:

- Test Number
- Pass number
- Expected Data
- Actual Data
- Test Specific Data
- Error Code

The error code display format is the same for all of the tests in both the all fast test sequence and the user selectable test program. In addition to displaying the error code message on the CRT, the program displays a binary error code in the LED indicators excluding the on line and off line LED's on the keyboard. Note the error codes displayed on the keyboard are in binary and that the error codes in table 5-1 along with the error codes displayed on the CRT are in octal. The error code interpretations for all of the microcode diagnostics are listed in table 5-1.

Table 5-1. Microcode Error Code Interpretation

Error Codes Octal	Definition
	CPU TEST
1	Jump Instruction Failure
2	Clear Instruction, Zero Status Bit Failure
3	Clear Instruction Failure
4	Add Instruction, Sign Bit Failure
5	Sign Bit Toggle Failure
6	Subtract Instruction, Sign Bit Failure
7	Increment and Decrement Instruction Failure
10	Complement and Negate Instruction Failure
11	Arithmetic Right Shift Instruction Failure
12	Arithmetic Left Shift Instruction Failure
13	Rotate Right Instruction Failure
14	Rotate Left Instruction Failure
15	Logical Right Shift Instruction Failure
16	Bit Clear Instruction Failure
17	Bit Set Instruction Failure
20	Logical or Instruction Failure
21	Logical and Instruction Failure
22	Exclusive OR Instruction Failure
23	Exclusive NOR Instruction Failure
24	Move Instruction Failure
	SCRATCH PAD RAM TEST
30	Page Zero Fixed Data Read-Write Compare Failure
31	Page Zero Sequential Data Read-Write Compare Failure
32	Fixed Data Compare Failure on Pages 1 - 15
33	Sequential Data Compare Failure in Pages 1 - 15
	MEMORY CONTROL TEST
40	Memory Address Register Read-Write Failure
41	Memory Address Register Increment Failure
42	Memory Read-Write Compare Failure
43	Memory Sequential-Register Data Compare Failure
44	Memory Sequential-Stack Data Compare Failure
45	Memory Sequential-Scratchpad Data Compare Failure
46	Memory Sequential-Random Data Compare Failure

Table 5-1. Error Code Interpretation (Continued)

Error Codes Octal	Definition
	KEYBOARD TEST
50	Sequential Keystrike Test Failure
51	Shifting Zero Keystrike Test Failure
52	Shifting One Keystrike Test Failure
	DISPLAY TEST
60	Keyboard Printer Control Bit Read-Write Test Failure
61	Display control Register Bit Read-Write Failure (Bits 0 - 4)
62	Display control Register Bit Read-Write Failure (Bits 7 - 12)
63	Data Compare Error in Pixel Read-Write Test (Part of Screen)
64	Data Compare Error in Pixel Read-Write Test (Entire Screen)
65	Previous State Error in Pixel Complement Test
70-73	User Defined

5.4.2 User Selectable Test Program

The user selectable test program is only accessible after the system has been booted. To gain access to the user selectable test program press the SET-UP key. Once the SET-UP key is pressed, a menu similar to the one shown below appears across the bottom of the CRT:

1) Line on	3) Color Normal	5) Click off/on	7) Scroll Fast	0) Reset
2) Mode GR	4) Cursor Block	6) Rate 60 Hz	8) Maintenance	

To start the user selectable test program, disconnect the printer from the system and press key number 8.

Note: Once the user selectable test program is started, keystrokes are not echoed on the CRT nor do any menus or prompts appear on the CRT.

5.4.2.1 Command Mode. The command mode is automatically entered whenever the user selectable test program is started. The valid commands are as follows:

M - Memory available
O - Enter option select mode (refer to paragraph 5.4.2.2)
R - Reset
S - Current status display
T - Test select and execute mode (refer to paragraph 5.4.2.3)
X - Exit

To execute any of the commands listed above, press the key associated with the command. Note that the command is executed immediately after the key is pressed. No carriage return (CR) is required and no other characters are accepted. Whenever the command mode is entered, an error code message is displayed on the CRT, reflecting the parameters of the last test run.

a. **Memory Command.** When the memory command is selected the total amount of memory available is displayed on the CRT. The highest address displayed has the least significant word on the third line and the most significant word on the fourth line. The two lines above and below are zero. The time to size memory depends on the amount of memory present but in general should be less than 1 second for 256K words.

b. **Reset command.** The reset option is used to reset all of the test options (i.e., clears all of the test options except halt on error) and the current status display (current status display equals all zero's) to their default values.

c. **Current Status Display Command.** When this option is selected, it displays the status of the last test successfully completed on the CRT. The format of the display is as follows:

Six octal digits (sixteen bits binary)

- All zeros (indicates monitor running)
- Total error count
- Bits "stuck" high
- Bits "stuck" low
- Current test options
- All zeros (indicates no errors)

5.4.2.2 Option Select Mode. The option select mode allows the user to set or clear specific options that occur when a specific condition is encountered. The valid commands are as follows:

B - Beep on keystroke, error and test completion

D - Diagnostic mode, reserved for manufacturing test purposes only

H - Disable halt on error

L - Loop forever (Restart test after completion)

M - A special option bit has been designated to allow specialized testing for manufacturing. When this bit is set the test sequence (All Tests) is selected, and the standard powerup test is run followed by memory tests 3, 4, and 5, the fast and slow display board tests. All existing test options are supported when this option is selected (i.e., the sequence may be run continuously if the Loop forever bit is set). Keyboard tests are a special case and will never be run in a test sequence or in the Loop forever mode. This bit is toggled ON by pressing the M key and the RETURN key. To toggle the bit OFF press the M key and the DELETE key.

P - Pass count display (Allows the error code message to be displayed on the CRT as a test is being run)

R - Reset test after error (Loop on error)

X - Exit to command mode (The only way to exit from the option select mode).

Note: The loop forever and disable halt on error options cannot be used in conjunction with any of the keyboard tests.

To select an option, press the key associated with the option and the carriage RETURN key. To select multiple options, repeat the procedure described above for each option.

To clear an option, press the key associated with the option and the DELETE key. To clear multiple options, repeat the procedure described above for each option. To exit from the Option Select Mode, press the X key.

5.4.2.3 Test Select Mode. The test select mode allows the user to select and run any of the following tests. The valid commands are as follows:

A - All Fast Test sequence Test
C - CPU (2501) Test
D1 - Display Test Fast
D2 - Display Test Slow
M1 - Memory Address Register Read/Write Test
M2 - Memory Address Register Increment Test
M3 - Memory Read/Write Compare Test
M4 - Memory/CPU Interaction Test
M5 - Fast Memory Read/Write Test
K1 - Sequential Keystrike Test
K2 - Shifting Zero Keystrike Test
K3 - Shifting One's Keystrike Test
S - Scratch Pad RAM Test
X - Exit Test Select Mode (return to command mode)

To execute one of the tests listed above, press the key(s) which correspond to the test and it executes immediately except for the keyboard tests which wait until eight keys have been pressed in the order described below.

Note: If a test is selected during the test select mode which requires two keys to be pressed and only the first key is pressed, the system does not accept any other commands until an acceptable second key is pressed.

The program returns to the command mode upon the successful completion of each test. If an error is detected during a test, the test halts (if neither the disable halt on error or reset test after error options were selected during the option select mode) and display an error code (refer to table 5-1) message on the CRT. Once an error is detected, and a test is halted, the only commands that the program accepts are the abort and universal exit commands. To execute the abort command, press the A key. To execute the universal exit command press the X key. When either of these commands are executed the program returns to the command mode.

To stop a test in progress, press the S key. Once the test has been stopped, the user has the options of continuing the test from where it was stopped by pressing the C key, aborting the test by pressing the A key which returns the program to the command mode or the user can return to the command by pressing the universal exit key (X).

a. **All Fast Test Sequence (A).** The A command repeats the all fast test sequence described above which consists of the CPU test, scratch pad RAM test, fast display test, and fast memory test.

b. **CPU (2501) Test (C).** The CPU test has been designed to check the most commonly used operations and addressing modes of the (2501) CPU board and the scratchpad RAM section of the PROM board. The CPU test consists of a series of subtests that are performed sequentially, as soon as the C key is pressed. No other action by the user is required.

   If an error is detected during any of the subtests, the CPU test halts and displays a (binary) error code (refer to table 5-1) in the LED's on the keyboard. Note that an error code message will not be displayed on the CRT if this test should fail. If the CPU test should fail, try repeating the test several times to verify that it is a hard error. If this test should fail the probable cause is either a faulty CPU or PROM board.

c. **Fast Display Test (D1).** The fast display test operates basically in the same manner as the slow display test. The difference being that in the slow display test, all of the pixels are read after each subtest while in the fast test, all of the pixels are read after the last subtest. If this test should fail, the probable cause is either a faulty display board or CRT.

d. **Slow Display Test (D2).** The slow display test has been designed to check the medium resolution display board (all but the keyboard interfaces) and its ability to execute, function, generate appropriate status conditions, and read/write to all locations in the internal memory. Some of the subtests result in a momentary visual display if the CRT is functional. The slow display test consists of the following subtests which are executed sequentially:

Subtest 0 - Checks the basic read/write capability of the display control register.

Subtest 1 - Checks some of the basic functions and verifies that the appropriate status bits are set.

Subtest 2 - Write all pixels so they are 'set', read all pixels and verify their state.

Subtest 3 - Complement every other pixel (to clear). Read all pixels and verify alternating one's and zero's pattern.

Subtest 4 - Reset (clear) every other pixel and read all pixels verifying that all data is cleared.

Subtest 5 - Complement each pixel and verify its previous state as indicated by the pixel data bit. Then read all pixels and verify all are set.

Subtest 6 - Checks the vector function by drawing vectors in such a sequence that expected grid patterns will be seen on the screen.

Subtest 7 - Growing block using finite erase block function, a small block appears in the center of the screen and continues to 'grow' while alternating ON and OFF.

If an error is detected during any of the subtests, the slow display test halts and displays a (binary) error code (refer to Table 5-1) in the LED's on keyboard. An attempt is also made to display an error code message on the CRT. If this test should fail, the probable cause is either a faulty Display board or CRT.

e. **Memory Address Register Write/Read Test (M1).** This test checks the write/read capability of the address registers on the 2901 bus port by writing and reading a series of pseudo-random numbers for a total of 200 passes. If this test should fail, the probable cause is a faulty Memory Controller board.

f. **Memory Address Increment Test (M2).** This test checks the auto-increment feature of the address registers on the memory controller board by loading a known data value into the register, incrementing the register and comparing the register to the expected result. If this test should fail, the probable cause is a faulty Memory Controller board.

g. **Memory Read/Write Compare Test (M3).** This test is actually a Unibus memory test and is executed directly by the (2501) CPU through the Memory Controller board. It first determines the amount of available memory by writing all zero's followed by all one's starting at location zero. The first location that exhibits an error is assumed to be one location beyond the end of address space. This address is saved and further testing does not go beyond the high limit. The test then fills the entire memory space with a series of pseudo-random integers, reads the data back and checks for discrepencies. This takes 45 seconds to complete. If this test should fail, the probable cause is either a faulty Memory Controller board or a faulty Unibus memory board.

h. **Memory/CPU Interaction Test (M4).** This test checks the timing between Unibus memory and the CPU by writing and reading a series of five words into memory via the registers on the Memory Controller board and the scratchpad RAM on the PROM board. If this test should fail, the probable cause is either a faulty CPU board, PROM board or Memory Controller board.

i. **Fast Memory Read/Write (M5).** The memory controller fast write/read test is very similiar to the memory controller slow write/read test described. The difference being that during the slow test a new seed is fed into the random number generator for each random number output; during the fast test a single seed is fed into the random number generator which produces a single random number that is then continually incremented by one. If this test should fail, the probable cause is a faulty Memory Controller board.

If any of the memory controller tests should fail, a binary error code is displayed in the LED's on the keyboard (refer to Table 5-1) and an attempt is made at displaying an error code message on the CRT.

j. **Sequential Keystrike Test (K1).** After the sequential keystrike test is selected, the test waits until eight keys have struck. The following seven keys to be pressed shall be such that the ASCII codes (refer to Table 5-2) are in an ascending order and differ by a factor of one. The ASCII code for the first key must be less than 171 octal. If an error is detected, the test should be repeated at least twice to verify that correct keys were pressed in the proper order. If this test should fail, the probable cause is either a faulty Display board (keyboard interface) or keyboard.

Table 5-2. 7-Bit ASCII Code

Octal Code	Char	Octal Code	Char	Octal Code	Char	Octal Code	Char
000	NUL	040	SP	100	@	140	`
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	_	177	DEL

k. **Alternating (Shifting) Zero Test (K2).** After the alternating zero test is selected, the following eight keys must be pressed in the order listed:

Key	ASCII Code Character
Delete key	DEL
~ Key	~
Right Parenthesis	}
Left Parenthesis	{
Lower case 'w'	w
Lower case 'o'	o
Underline	_
Question Mark	?

If an error is detected, the test should be repeated at least twice to verify that the correct keys were pressed in the proper order. If this test should fail, the probable cause is either a faulty Display board or keyboard.

l. **Alternating (Shifting) Ones Test (K3).** After the alternating ones test has been selected, the following eight keys must be pressed in the order listed:

Key	ASCII Code Character
CTRL - @	NUL
CTRL - A	SOH
CTRL - B	STX
CTRL - D	EOT
CTRL - H	BS
CTRL - P	DLE
Space Bar	SP
At Sign	@

If an error is detected during the keyboard test, the test should be repeated at least twice to verify that the correct keys were pressed in the proper order.

If this test should fail, the probable cause is either a faulty keyboard or a faulty Display board.

If any of the keyboard tests should fail, an attempt will be made to display a (binary) error code in the LED's on the keyboard (refer to table 5-1) as well as an error code message on the CRT.

m. **Scratchpad RAM Test (S).** The scratchpad RAM memory test is essentially a memory test which uses four data patterns to identify the most common types of memory failures. Data is written to the entire scratchpad RAM bank and then read back. If this test should fail, a binary error code will be displayed in the LED's on the keyboard and an attempt will be made to display an error code message on the CRT. Should this test fail the probable cause is either a faulty CPU or PROM board.

5.5 RT-11 BASED DIAGNOSTICS

The following RT-11 based routines are used to test the analog to digital front end, floppy disk drive, front panel and utility board.

5.5.1 Loading Instruction

Step 1. Mount the diagnostic diskette #2515-6091 on drive DY0.

Step 2. Power up and boot the system as described in the 2515 System Description and Operation Manual (2515-0100).

5.5.2 Winchester and Floppy (WINFLP) Routines

Some queries require a specific response such as "Y"(es) or "N"(o) and waits until one of the responses is typed. All other queries use a default response of "NO" if a carriage return, space bar or other character is typed.

All numeric inputs and outputs are in octal.
Insert all diskettes (all must be of the same density).

NOTE

Remove the diagnostic floppy and mount a formatted scratch floppy.

NOTE

Before running the WINFLP diagnostic be sure that floppy disks are present in all drives to be tested. The program starts by determining all devices present and setting them up in an internal device table. A floppy drive is not recognized unless a diskette is mounted. No writing takes place until the device is explicitly enabled for testing by the user. Winchester drives must be up to speed although they need not be formatted.

In response to the prompting dot (.) enter **Run WINFLP (CR).** After entering the run command the following text is displayed on the screen and the Setup Configuration mode is entered.

```
GENRAD FLOPPY-WINCHESTER DIAGNOSTIC

CTRL-C RETURNS TO RT-11

CTRL-R ABORTS FUNCTION AND RETURNS TO COMMAND MODE

ALL NUMERIC INPUTS/OUTPUTS ARE IN OCTAL

REMOVE SYSTEM DISKETTE IF PRESENT!

INSERT ALL DISKETTES (ALL MUST BE OF SAME DENSITY)

 WINFLP V5F
 ENTER SYSTEM TYPE  (0-2515, 1-2510, 2-2511/2514) ____ (CR)


          *0 HELP TEXT
          *1 SETUP CONFIGURATION
          *2 ACCEPTANCE TEST
          *3 SPECIFIC TEST
          *4 UTILITY FUNCTIONS

       Enter Desired Selection 0-4 _____
```

5.5.3 Command Mode Menu

```
          *0 HELP TEXT
          *1 SETUP CONFIGURATION
          *2 ACCEPTANCE TEST
          *3 SPECIFIC TEST
          *4 UTILITY FUNCTIONS

             Enter Desired Selection (0-4) ___
```

To select one of the routines listed in the command mode menu shown above, press the key associated with the desired routine. After one of the keys (0-4) is pressed, the following query will be printed on the screen:

```
             Are You Sure (Y, N)? ___
```

A description of each routine contained in the command mode menu is provided in the following paragraphs.

Note: For a description of the error messages associated with the WINFLP program, refer to table 5-3.

5.5.3.1 Help Text. When the help text routine is selected, the system prints the following general information on running the floppy and Winchester routines which should be reviewed before proceeding.

- Before any of the commands in the command mode menu may be executed, except the help text command, the user must enter the setup mode and specify which devices are to be test enabled (TE).
- Write protect all units which have data to be saved, **the WINFLP program is data destructive.**
- Insert pre-formatted, write enabled floppy diskettes in all drives.
- Reset all status counters before running the acceptance tests. Acceptance tests run continuously on all drives until stopped by the user. Note that track limits are not used when running acceptance tests.
- Individual tests run only a single pass on each drive.
- The line feed key (LF) can be used to examine the current track and sector status while a test is in progress.
- Restart addresses are 1104 or 1114 octal.
- CTRL C-Allows return to RT-11 if available. If in ODT, return control to the program.
- CTRL D - Causes transfer to ODT.
- CTRL O - Aborts output to screen for duration of current command.
- CTRL R - Aborts current function and returns control to command mode.
- CTRL S - Stops output to screen, press any other key to continue.
- If 200 errors occur on the same logical unit then the unit is purged from testing. This means that the device is test disabled and the next available device selected for testing. If no more devices are available the test is aborted. (A specific message is displayed and the user must type a character to continue.)

Enable additional help text during testing (Y/N)? ___

A Yes response causes a short description of each test to be displayed before each test is run.

5.5.3.2 Setup Configuration. The setup configuration command enables the user to specify which drives are to be accessed by the various test functions. The default status is all devices test disabled.

NOTE

The setup information should be reviewed before starting any test.

This command also allows the user to set the test parameters as described below. Note that Winchester drives may contain more than one logical device number.

```
*0 HELP TEXT
*1 SETUP CONFIGURATION
*2 ACCEPTANCE TEST
*3 SPECIFIC TEST
*4 UTILITY FUNCTIONS

Enter Desired Selection (0-4) _
Are You Sure (Y,N)? _
```

The address for control and status registers and interrupt vectors are:

DEVICE	CSR ADDRESS	INTR VECTOR ADDRESS
RL01/02	174400	160
RL01/02	175000	120
RX02	177170	264
RX02	177150	270

```
Winchester drives may contain more than one logical unit number.
The track range may be set for each unit individually.
Devices are excluded from testing unless explicitly specific.
Units with data to be saved should be write protected.
Enabling interrupts causes an interrupt to be generated with each disk operation.
Review setup information before starting testing.
Type a character when you are ready to proceed ___.

CHANGE TEST CHARACTERISTICS (Y,N)? ____
```

A Yes response results in the halt on error, screen page mode, specific test pass count and device purge error count queries (description follows) being asked. A No response results in the system using either the default value associated with each query or the value last specified for each query.

Enable Halt on Error (Y,N)? _

If the halt on error is enabled, the program stops after printing an error message and waits for either a control character to abort the test or a "C" (continue) to return to the test which was being run. This feature allows time for user comprehension of the error message. Another valuable technique is to enter ODT at this point and attempt the same function manually with slightly different parameters. A single control C brings control back to the halt on error handler where the test may be continued or aborted. If halt on error is not enabled the program prints error messages as fast as they occur and immediately returns to the test in progress.

Enable Screen Page Mode (Y, N)? _

Enabling screen page mode allows only 40 lines of text to be displayed before the program "waits". To view the next 40 lines of text requires that a single character be typed. This feature is implemented in software and is independent of hardware scroll or page mode. Disabling screen page mode causes messages to be displayed as fast as they occur and without regard to user comprehension. This should be the case when the test is run unattended since all error messages are saved in an error buffer for future user review. Before the error buffer is examined page mode should be enabled as a convenience to the user.

Specific Test Pass Count (Y/N)? _

Specific tests are normally used as a "quick" check of a specific function or device (i.e., scanning a diskette for CRC errors). This test allows a function to be checked more extensively over a longer period of time. The maximum test pass value is 40,000 octal or 32767 decimal. The test continues until this test pass count is reached or until the device accumulates a critical number of errors (see purge count below).

Purge Error Count (Y,N)? _

When several logical units are being tested it is not desirable to allow one bad unit to kill the entire test. For this reason the test 'purge' the 'bad' unit (i.e., disable it from further testing) when a certain number of errors have been accumulated (if the reply to the Purge Error Count query is Yes). After this, the test continues to the next available device or stops the test if no further devices are available. The maximum permissable purge count is 40,000 octal or 32767 decimal. This purge count applies to all devices and cannot be set to different values for individual units.

The address for control and status registers and interrupt vectors are:

DEVICE	CSR ADDRESS	INTR VECTOR ADDRESS
RL01/02	174400	160
RL01/02	175000	120
RX02	177170	264
RX02	177150	270

Winchester drives may contain more than one logical unit number.
The track range may be set for each unit individually.
Devices are excluded from testing unless explicitly specific.
Units with data to be saved should be write protected.
Enabling interrupts causes an interrupt to be generated with each disk operation.
Review setup information before starting testing.
Type a character when you are ready to proceed ___.

The current device states and status counters for errors and operations are shown.
After the display you may clear all status counters by typing 'Y' to the query.

LG UNIT	UNIT	DEVICE	FLAGS	#ERRORS	#RD/WT	#XFERS	#DATA ERRORS
1	0	W WE	TD ID	0	0	0	0
2	1	W WE	TD ID	5	36	36	5
3	0	F WE	TD ID	0	470	235	0

NOTE

Device flag meanings are as follows:

Column 1	(W or F)	Specifies Winchester or floppy
Column 2	(WE or WP)	Specifies write enabled or protected
Column 3	(TD or TE)	Specifies test disabled or enabled
Column 4	(ID or IE)	Interrupts disabled or enabled

Reset Status Counters (Y,N)? ___

A yes will reset the values of the #ERRORS, #RD/WT, #XFERS, and the #DATA ERRORS to zero.

Change Device Status (Y, N)? ___

If the device status is to be changed, the user must enter the logical unit number for the particular device. This number is obtained in the first column of the status display. If a new unit is to be entered then the logical unit number must be the first available number. For example if 2 logical units are entered in the table then the new device would be assigned logical unit number 3. Any number which is higher would not be accepted.

Device (1-3)? ___ **(CR)**
CSR: 177170

The control and status register address is extremely important. Only valid possibilities are accepted after being checked against a table of possible values and checking that the address actually exists. Invalid responses are never accepted and the query will be repeated until a valid entry is made. To change the control status register (CSR) address type in the new address followed by a carriage return.

NOTE

If the control status register address is changed, the following jumper configurations on the AED controller must also be made.

RL01/02 (Winchester)	Standard	Alternate
PIO address	774400	775000
Jumper	E4-E5	E5-E6
RX02 (5¼" floppy)	**Standard**	**Alternate**
PIO address	777170	777150
Jumper	E7-E8	E8-E9

Physical Unit Number (0,1)? ___

The physical unit number is independent of the logical unit number and reflects the number which is recognized by the controller. There may be more than one physical unit per Winchester drive but only one unit per floppy drive. Be careful if this number is changed.

Interrupt Vector Address: ___

The interrupt vector address is setup automatically during program startup and should never be changed. If a new logical unit is entered the vector address must be valid for the particular control and status address. This information may be obtained in the disk controller manual. Standard interrupt vectors that correspond to CSR addresses are:

Device	CSR Address	Interrupt Vector
RL01/02	174400	160
RL01/02	175000	120
RX02	177150	264
RX02	177150	270

To change the interrupt vector address, type the new value followed by carriage return. To accept the current value, press either the space bar or carriage return key.

Lower Track Limit: ___

To change the value of the lower track limit, type the new value followed by a carriage return. The minimum track limit for any device is zero. To accept the current value, press the space bar.

Upper Track Limit: ___

To change the value of the upper track limit, type in the new value followed by a carriage return. To accept the current value, press the space bar or carriage return key. The maximum allowable track limits are as follows:

Floppy - 114 (octal)
RL01 - 377 (octal)
RL02 - 777 (octal)

The entire track range for each device is used. Install scratch disks in all floppy drives. Setup mode is entered when a character is typed.

* 0 Floppy and Winchester
* 1 Floppy Only
* 2 Winchester Only

Enter Desired Selection (0-2) ___
Are you sure (Y,N)? ___

Device	CSR Address	Interrupt Vector Address
RL01/02	174400	160
RL01/02	175000	120
RX02	177150	264
RX02	177150	270

Winchester drives may contain more than one logical unit number. The track ranges may be set for each unit individually. All devices are excluded from testing unless explicitly specified. Units with data to be saved should be write protected. Enabling interrupts causes an interrupt to be generated with each disk operation.

Review setup information before starting the test. Type a character when you are ready to proceed. The current device states are status counters for error operations shown. After the display you may clear all status counters by typing Y to the following queries:

Reset Status Counters (Y,N)? ___
Change Device Status (Y,N)? ___

LG UNIT	UNIT	DEVICE	FLAGS	#ERRORS	#RD/WT	#XFERS	#DATA ERRORS
1	0	W WE	TE ID	0	0	0	0
2	1	W WE	TE ID	5	36	36	5
3	0	F WE	TE ID	0	470	235	0

Floppy and/or Winchester acceptance test now starting.

NOTE

The track range must be set for each unit individually.

Enable Device Testing (Y, N)? ___

A Yes will cause TE (test enabled) to appear in the current status table shown above. A No causes TD (test disabled) to appear. If the carriage return key or the space bar is pressed, it causes the device to be test disabled (TD).

Write Protect Device (Y, N)? __

All units with data to be saved must be write protected (WP). If either the carriage return key or the space bar is pressed, it causes the device to be write enabled (WE).

Enable Interrupts (Y, N)? ___

A Yes will cause an interrupt to be generated (IE) with each disk operation. If either the carriage return key or the space bar is pressed, it causes the interrupt to be disabled (ID). The current device states are status counters for errors. Operations are shown after the display. To clear all status counters type 'Y' to the query.

LG UNIT	UNIT	DEVICE	FLAGS	#ERRORS	#RD/WT	#XFERS	#DATA ERRORS
1	0	W WE	TE ID	0	0	0	0
2	1	W WE	TE ID	5	36	36	5
3	0	F WE	TE ID	0	470	235	0

To return to the command mode menu, answer No to the next two queries.

Reset Status Register (Y,N)? ___
Change Device Status (Y,N)? ___

5.5.3.3 Acceptance Test. The acceptance test performs the following sequence of test once on each device that is test enabled.

- Interface Test
- Interrupt Test
- Scan Test
- Sequential Write/Read Test
- Random Write/Read Test
- Seek Test

Each device is tested individually once; at the completion of each test an asterisk (*) is printed on the screen. This test continues until the user types CTRL R. This test does one short acceptance test on the first eight tracks and then resets the limit variables back to the default values. It then initiates the continuous test.

NOTE

If this test should fail, an error message(s) will be printed on the screen. For a description of the error messages and appropriate corrective action refer to Table 5-3.

When the acceptance test is selected, the following text is printed on the screen:

A series of all individual tests is run on each drive selected for testing.
The entire track range for each device is used.
Install scratch disks in all floppy drives.
Setup mode is entered when a character is typed.

Description of Status and Error Displays. WINFLP prints out error and status information under a wide variety of circumstances. All error messages printed on the screen are also sent to a circular buffer in memory as well. The buffer size is determined by available memory. The circular buffer is useful if a hard copy is not being run off and the error printouts are longer than can be displayed on the screen. The dump error buffer command in the utility menu is used to examine the error messages in the circular buffer. The status variables that might appear on the screen are explained below:

RXCS:	Shows the contents of the floppy command and status register.
RLCS:	Shows the contents of the winchester command and status register.
#BAD:	This variable indicates the number of status errors detected.
#RD/WRT:	This variable indicates the number of read and write operations performed error free.
RXDB:	Shows the contents of the floppy data buffers.
RLDA:	Winchester disk address register.
RLBA:	Winchester bus address.

Each error message includes a test mnemonic (a two-character code) which identifies the specific test being run at the time of failure. Valid test mnemonics are as follows:

SC:	SCAN TEST
IF:	INTERFACE TEST
IT:	INTERRUPT TEST
RW:	RANDOM WRITE-READ TEST
RR:	RANDOM READ TEST
SW:	SEQUENTIAL WRITE-READ TEST
SR:	SEQUENTIAL READ TEST
SK:	SEEK TEST
FM:	FORMAT UTILITY

The test mnemonic is always located on the second line of the error message just before the error identifier (bounded by asterisks).

Logical unit 3 unit # 0 CSR: 177170 at track 0 sector 1

SW * WRITE PROTECT ERROR *

RXCS	RXDB	ERR	CODE
104446	0	100	

The "SW" indicates that the sequential write/read test was being run on a floppy drive which was write protected.

Table 5-3. WINFLP Error Messages and Meanings

*** Buffer Unchanged by Read ***

This error message is printed if an initial data pattern is still present in memory following an empty buffer operation. This indicates that no data transfer took place. (RX only.)

Possible Causes	Corrective Action
• Controller bad	Replace the controller and repeat the test.

*** Bus Reset Error ***

ADDRESS Expected	Actual Data
17XXXX	XXXX

This error message is printed when a bus initialization does not clear the appropriate bits in the interface register. Note that certain emulations may not support this function and may not be truly 'critical' unless other errors are also seen. (RL and RX.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** CRDY Time out Error ***

RLCS

XXXX

This error message is printed when the controller ready bit does not set within a reasonable period of time indicating that the function did not complete. (RL only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle the main power and repeat the test. If the error persists, examine the contents of the RLCS.
• Controller bad	Replace the controller and repeat the test.

*** Data Address Mark Not Found ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0170

This error message is printed when the first portion of a diskette header field is located but the start of the data filed could not be located in two revolutions of the media. (RX only.)

Table 5-3. WINFLP Error Messages and Meanings (Continued)

***Data Buffer Overread ***

Actual	Expected	Word #
XXXX	XXXX	XXXX

This error message is printed if during an empty buffer operation one or more extra words were transferred to main memory. Check the word count through a read status operation and see if more than a single sector of data was specified. (RX only.)

Possible Causes	Corrective Action
• Controller bad	Replace the controller and repeat the test.

*** Data Compare Error ***

WORD #	EXP	ACT DATA
XXXX	XXXX	XXXX

This error message is printed when data read from a specific disk sector does not compare to the expected value. (RX or RL.) Data expected for these words have the following meaning:

RL

Word 1 = Track Address
Word 2 = Sector Number
Word 3 = Side (0 or 1)
Word 4 = Unit Number (0, 400, 1,000 or 1,400)

RX

Byte 2 = Sector Number
Byte 3 = Track Number

Table 5-3. WINFLP Error Messages and Meanings (Continued)

Possible Causes	Corrective Action
• Invalid track limits	Enter setup mode and verify that the specified track limits are between 0 and 114 octal (floppy); 0 and 377 octal (RL01); 0 and 777 (RL02). If not, enter valid track limits and repeat the test.
• Data was not initialized by a sequential write/read test before this test was attempted.	Run the sequential write/read test and repeat the test.
• Floppy bad	Replace the floppy and repeat the test.
• Winchester bad	Format the drive, retry the test, and if error persist, run a bad block utility* (do not repeat the test).

*** Data CRC Error ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when a cyclic redundancy check on a sector data field fails. (RL only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Data not initialized by a previous sequential write/read test.	Run the sequential write/read test and repeat the test.
• Winchester not formatted properly.	Reformat the Winchester and repeat the test.
• Bad blocks	Run a bad block utility and repeat the test.

* RT-11 DUP, INIT/BAD
XXDP+ CZRLMD0

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Data CRC Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0200

This error message is printed when a cyclic redundancy check on the diskette data fields fails. (RX only.)

Possible Causes	Corrective Action
• Bad spot on the disk	Format the diskette and repeat the test. If the problem persists replace the diskette.
• Drive bad	Replace the drive and repeat the test.

*** Data Error ***

Actual Data	Expected Data	Word #
XXXX	XXXX	XXXX

This error message is printed when the data read after a floppy operation does not compare to the expected data. Word number two contains byte 2 (sector) and byte 3 (track) information. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Diskette bad	Replace the diskette and repeat the test. If the error persists on the same track or sector, replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Data Error - Empty Buffer ***

Actual Data	Expected Data	Word #
XXXX	XXXX	XXXX

This error message is printed when the data checked after a floppy empty buffer operation does not compare to the expected data. Word number two contains byte 2 (sector) and byte 3 (track) information. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Diskette bad	Replace the diskette and repeat the test. If the error persists on the same track or sector, replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Density Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0240

This error code message is printed when the density specified in the RXCS density bit (400 octal) does not match the density of the media. Note that certain 5¼" floppy emulations may report this error since they do not support "single density" emulations. (RX only.)

Possible Causes	Corrective Action
• Incorrect media density	Read drive status through ODT and verify the actual density of the media. If the media is single density, reformat it to double density and repeat the test.
• Controller bad	Replace the controller and rerun the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Done Time Out Error ***

RXDB	RXCS
XXXX	XXXX

This error message is printed when the operation done bit (40) in the RXCS does not set within a reasonable period of time. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Loose cabling	Reseat all cables.
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Transfer Request Time Out Error ***

RXDB	RXCS
XXXX	XXXX

This error message is printed when the transfer request bit (RXCS bit 200) does not set within a reasonable period of time on a multiple parameter operation. (RX only.)

Possible Causes	Corrective Action
• Timing conflicts - controller bad	Initialize the system, cycle main power and repeat the test. If the same error occures repeatedly, replace the controllor.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** DRDY Time Out Error ***

RLCS

XXXX

This error message is printed when the drive ready bit (1) does not set within a reasonable period of time indicating that either the drive is bad or has not yet completed an operation. (RL only.)

Possible Causes	Corrective Action
• Loose cabling	Check that all of the cables between the system and drive are connected.
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Drive Home Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0050

This error message is printed when resetting the drive to track zero and the track zero indicator does not come true. (RX only.)

Possible Causes	Corrective Action
• Obstruction of the heads	Remove any obstacles that may obstruct the free movement of the heads and repeat the test.
• Loose cables	Verify that all cables are connected and properly seated and repeat the test.
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Drive Select Error ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when an attempt is made to access a drive unit which does not exist. (RL only.)

Possible Causes	Corrective Action
• Physical unit number not entered in setup utility	Enter setup mode and verify that a valid physical unit number has been entered for the unit under test. If not, enter a valid number and repeat the test.
• Loose cables	Seat all loose cables.
• Unit select jumpers on the drive not set correctly	Set the jumper on the device under test as described in paragraph 5.5.3.2 and repeat the test.

*** Drive 0 Initialize Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0010

This error message is printed when either an initialization operation or a power up initialization cycle results in problems being detected on RX unit 0. (RX only.)

Possible Causes	Corrective Action
• Head obstructed	Remove any obstructions from the heads and repeat the test.
• Media not present	Mount a diskette in the empty drive and repeat the test.
• Cables not appropriately connected	Check that all cables are connected correctly and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Drive 1 Initialize Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0020

This error message is printed when either an initialization operation or a power up initialization cycle results in problems being detected on RX unit 1. (RX only.)

Possible Causes	Corrective Action
• Head obstructed	Remove any obstructions from the heads and repeat the test.
• Media not present	Mount a diskette in the empty drive and repeat the test.
• Cables not appropriately connected	Check that all cables are connected correctly and repeat the test.

*** Extra Transfer Request ***

RXDB	RXCS
XXXX	XXXX

This error message is printed when the transfer request bit (0200) in RXCS sets, indicating that the controller is waiting for an additional parameter during a floppy operation. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initilize the system, cycle main power and repeat the test. If the error persists, retry the same function through ODT and verify that all of the parameters are being accepted by the controller.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Floppy Initialize Done Error ***

RXCS	RXDB
XXXX	XXXX

This error message is printed when the floppy initialize operation does not complete and thus did not set the initialize done bit (4) in RXDB. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Floppy Initialize Time Out Error ***

RXCS
XXXX

This error message is printed when the initialization operation on the floppy does not complete in a reasonable period of time. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Floppy Status Error ***

RXCS RXDB ERROR CODE

XXXX XXXX XXXX

This error message is printed when any error is detected which does not fall into any of the previous categories. (RX only.)

Possible Causes	Corrective Action
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Data CRC Error ***

RLCS RLBA RLDA STATUS

XXXX XXXX XXXX XXXX

This error message is printed when a cyclic redundancy check on the sector header is incorrect. (RL only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Winchester improperly formatted	Reformat the Winchester and repeat the test. If the error persists, replace the Winchester drive.
• Bad blocks	Run a bad block utility*.

* RT-11 DUP, INIT/BAD
XXDP+ CZRLMD0

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Data CRC Error ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when a cyclic redundancy check on the sector data field fails. (RL only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Winchester improperly formatted	Reformat the Winchester and repeat the test.
• Bad blocks	Run a bad block utility*.

*** Header Not Found ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when a target disk address is not located on the media. (RL only.)

Possible Causes	Corrective Action
• Invalid disk address	Execute a read header operation through ODT to verify the actual disk address against the target. Sectors may range from 0 to 47 (octal) and tracks range from 0 to 377 (octal) for RL01 or 777 (octal) for RL02.

* RT-11 DUP, INIT/BAD
XXDP+ CZRLMD0

Table 5-3. WINFLP Error Messages and Meanings (Continued)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Incorrect format	Reformat the Winchester and repeat the test.
• Bad blocks	Run a bad block utility*.

*** ID Address Mark Not Found ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0160

This error message is printed when the identification address mark in a diskette sector header is not located in two revolutions of the media. (RX only.)

Possible Causes	Corrective Action
• Diskette bad	Reformat media and repeat the test.

* RT-11 DUP, INIT/BAD
XXDP+ CZRLMD0

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** ID Mark Not Found ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0130

This error message is printed when the sector preamble is located on the diskette but the identification mark is not located. (RX only.)

Possible Causes	Corrective Action
• Media improperly formatted	Format the media and repeat the test.
• Media bad	Replace the media and repeat the test.
• Drive bad	Replace the drive and repeat the test.

*** Interface Test Error ***

Address	EXP	Act Data
17XXXX	XXXX	XXXX

This error message is printed when one or more of the read/write bits in the interface registers can not be set or cleared. (RL and RX.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Initialize Done - Power Fail ? ***

RXCS RXDB

XXXX XXXX

This error message is printed when the initialize done bit and the ac power low bits are set unexpectedly. (RX only.)

Possible Causes	Corrective Action
• Drive bad	Replace the drive and repeat the test.
• Power supply bad	Replace the power supply and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Interrupt Error ***

CSR VECTOR #INTR

XXXX XXXX XXXX

This error message is printed when the interrupt enable bit (100) is set and an extra interrupt occurs or no interrupt occurs. (RX and RL.)

Possible Causes	Corrective Action
• CPU priority not equal to zero.	Use ODT to check the priority level of the CPU. If the priority level is not equal to zero, reassign it to level zero and repeat the test. If the error persists, replace the controller.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** No Bus Response ***

ADDRESS

17XXXX

This error message is printed when the controller has died or the CSR address is not correct. (RX or RL.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system, cycle main power and repeat the test.
• CSR address is not correct	Enter setup mode and verify that a valid CSR address has been specified. If not, enter a valid address and repeat the test. If the address is correct, check that the jumper configuration on the controller matches the specified address. If they match it indicates a faulty controller.
• Controller bad	Replace the controller and repeat the test.

*** No DMA During Status Operation ***

RXCS RXDB

XXXX XXXX

This error message is printed during a read extended status operation when the data transfer does not take place. (RX only.)

Possible Causes	Corrective Action
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Nonexistent Memory Errors ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when an attempt is made during a read or write operation to transfer data (via DMA) to a location in memory which does not exist (Trap to 4). The bus address register RLBA will contain the address of this location. (RL only.)

Possible Causes	Corrective Action
• System memory consists of less than 16K.	Check the memory space through the memory map utility and look for small "holes" in the first 32K memory. If any are found, replace the memory board and repeat the test.

*** Preamble Not Found ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0120

This error code is printed when the sector preamble could not be found in two revolutions of the media. (RX only.)

Possible Causes	Corrective Action
• Media improperly formatted	Format the media and repeat the test.
• Media bad	Replace the media and repeat the test.
• Drive bad	Replace the drive and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Power Fail ***

RXCS	RXDB
XXXX	XXXX

This error message is printed when the ac low bit in RXDB sets indicating power may be low. (RX only.)

Possible Causes	Corrective Action
• Interface cable is not correctly installed	Repeat the interface cable as required and repeat the test.
• Low ac power	Measure customer ac power line.
• Power supply bad	Replace the power supply and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Read-Write Circuit Failure ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0230

This error message is printed when an error is detected in the internal read-write circuitry by the controller. (RX only.)

Possible Causes	Corrective Action
• Controller bad	Cycle main power and repeat the test. If problems persist replace the controller and repeat the test.
• Diskette improperly formatted	Reformat the diskette and repeat the test.
• Drive bad	Replace the drive and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Sector Not Found ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0070

This error message is printed when the controller can not find the requested sectors is two revolutions of the media. The allowable floppy sector range is 1 to 26 (decimal). (RX only.)

Possible Causes	Corrective Action
• Floppy not formated properly	Format the floppy and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Seek Time Out Error ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when a seek operation on the Winchester does not complete within a reasonable period of time. (RL only.)

Possible Causes	Corrective Action
• Headlock still in place	Remove the headlock and repeat the test.
• Controller hung	Initialize the system, cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** SEP Clock Time Out Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0110

This error message is printed when a internal controller circuit failure is detected. (RX only.)

Possible Causes	Corrective Action
• Loose cabling	Reseat all the cables and repeat the test.
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Set Media Key Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0250

This error message is printed when a set media operation is attempted with an incorrect keyword is specified. The normal keyword is 111 octal. (RX only.)

Possible Causes	Corrective Action
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Spin Error ***

RLBA RLDA STATUS

XXXX XXXX XXXX

This error message is printed when an operation is attempted on an RL which is not up to speed or has not completed an initialization sequence. (RL only.)

Possible Causes	Corrective Action
• Controller hung	Cycle main power and allow a minute for the drive to reach full speed then repeat the test.
• Spindle locks still on drive	Remove the spindle locks and repeat the test.

*** Status Error ***

RLBA RLDA STATUS

XXXX XXXX XXXX

This error message is printed when an RL error has occurred that can not be decoded. (RL only.)

Possible Causes	Corrective Action
• Controller bad • LSI-11/23 bad	Check the function bits and the specific error bits in the CSR, and all bits in the status word register. Refer to the Digital Equipment Corporation RL and peripherals handbooks as well as the Advanced Electronics Design, Inc. WINC-05 Operations and Maintenance manual (Publication number 990016).

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Stuck Error Bit ***

RXCS	RXDB
XXXX	XXXX

This error message is printed when the composite error bit in RXCS has not cleared after several attempts to initialize the controller. (RX only.)

Possible Causes	Corrective Action
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Track Address Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0150

This error message is printed when the target disk address is not valid for the particular drive. (RX only.)

Possible Causes	Corrective Action
• Invalid track address being specified during setup mode (>114 octal).	Enter setup mode and change the track address if required (refer to paragraph 5.5.3.2). Initialize the system, cycle main power and repeat the test.
• Single density diskette	Mount a known good double density diskette and repeat the test.
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Track Limit Access Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0040

This error message is printed when an attempt has been made to access a track beyond the media range. The allowable track range is between 0 and 114 octal. (RX only.)

Possible Causes	Corrective Action
• A invalid track range was specified in the setup mode menu.	Enter setup mode and verify that the specified track range is between 0 and 114 octal. If not, enter a valid track range (between 0 and 114 octal) and repeat the test.
• Drive not initialized	Initialize the drive and repeat the test.
• Drive bad	Replace the drive and repeat the test.
• Controller bad	Replace the controller and repeat the test.

*** Unexpected Done ***

RXDB	RXCS
XXXX	XXXX

This error message is printed if during a multiple parameter operation on the floppy, the done bit (40) in RXCS is set which terminates the function before all of the parameters could be loaded to the controller. (RX only.)

Possible Causes	Corrective Action
• Controller hung	Initialize the system , cycle main power and repeat the test.
• Controller bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Word Count Overflow ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0230

This error message is printed if a read or write operation is attempted with a word count greater than one sector.

Possible Causes	Corrective Action
• Density discrepency between the drive and the media	Mount a new diskette that matches the density the drive is looking for and repeat the test.
• Software is not compatable with the system	Return unit and software to the factory.

*** Write Check Error ***

RLCS	RLBA	RLDA	STATUS
XXXX	XXXX	XXXX	XXXX

This error message is printed when the data on the RL does not compare to data written in a previous operation. (RL only.) Words 1-4 for the RL have the following meaning:

Word 1 = Track Address
Word 2 = Sector Number
Word 3 = Side (0 or 1)
Word 4 = Unit Number (0, 400, 1000 or 1400)

Table 5-3. WINFLP Error Messages and Meanings (Continued)

Possible Causes	Corrective Action
• Data not initialized by a previous sequential write/read test.	Run the sequential write/read test and repeat the test.
• Winchester not formatted properly	Reformat the Winchester and repeat the test.
• Winchester bad	Replace the Winchester and repeat the test.

*** Write Protect Error ***

RXDB	RXCS	Error Code (octal)
XXXX	XXXX	0100

This error message results when an attempted write operation to the floppy is unsuccessful. (RX only.)

Possible Causes	Corrective Action
• Write protect label is still on the floppy	Remove the write protect label from floppy and repeat the test.
• Floppy bad	Replace the floppy and repeat the test.
• Write protect sensor on drive is bad	Replace the drive and repeat the test.
• Controller is bad	Replace the controller and repeat the test.

Table 5-3. WINFLP Error Messages and Meanings (Continued)

*** Write Protect Error ***

CSR Device Flags

17XXXX XXXX

This error message is printed when a write operation is attempted and the drive media is write protected by either hardware or software. (RL only.)

Possible Causes	Corrective Action
• Device is not write enabled	Enter setup mode and select write enable for the drive to be tested and repeat the test.
• Write protect tape is on the floppy	Remove the write protect tape from the floppy and repeat the test.
• Write protect switches on Winchester are on (if any exist)	Turn all write protect switches off and repeat the test.
• Controller bad	Replace the controller and repeat the test.

5.5.3.4 Specific Testing. After the specific testing command has been selected from the command mode menu, the following menu will be printed on the screen.

```
*0 SCAN FOR MEDIA DEFECTS
*1 INTERFACE TEST
*2 INTERRUPT TEST
*3 SEQUENTIAL WRITE/READ
*4 SEQUENTIAL READ
*5 RANDOM WRITE/READ
*6 RANDOM READ
*7 SEEK TEST

Enter Desired Selection (0-7) __
```

To enter a selection, press the numeric key associated with desired test and answer the following query:

```
Are You Sure (Y, N)? __
```

NOTE

After each of the following tests have been successfully completed, the program returns to the command mode menu. If any of these tests should fail an error message is printed on the screen. For a description of the error messages and appropriate corrective action, refer to Table 5-3.

A description of each test is provided below.

a. **Scan For Media Defects.** The scan test reads all sectors on all test enabled drives sequentially, and checks for CRC errors. No direct data checking takes place in this test; only status is checked. After all units are scanned once, the command prompt is displayed on the screen.

b. **Interface Test.** The interface test checks the response of all interface registers and issues a response error if a bus time out occurs. All read/write bits in each register and the floppy data buffer are verified to be individually set and cleared without affecting other bits. A maintenance -op code is checked along with a bus reset. Non-media functions are also checked through various CPU operations.

c. **Interrupt Test.** All RL or RX-op codes (except write) are executed with CPU interrupts enabled. If an interrupt does not occur, an interrupt error message is displayed.

d. **Sequential Write/Read.** The sequential write/read test writes a psuedo-random data pattern sequentially on all selected tracks. The test then reads and checks all data patterns.

e. **Sequential Read.** This test assumes that the specified media range has been written to by a prior sequence write test. This test reads the data on all of the selected drives sequentially, and compares the data pattern against what was written.

f. **Random Write/Read.** This test assumes that the specified media range has been written to by a prior sequence write test. This test selects a random sector of a selected drive, then reads or writes it and checks the data when appropriate. This test checks only one sector per operation.

g. **Random Read.** This test assumes that the specified media range has been written to by a prior sequence write test. This test reads randomly from a fixed number of sectors in the specified media range and checks the data after each read. This test checks only one sector per read operation.

h. **Seek Test.** The seek test performs all possible seeks within the user specified track and seek length boundaries. Thus, it is a worst case test of the drive stepper motor and head settling time.

5.5.3.5 Utility Functions. After this function has been selected in the command mode menu, the following menu will be printed on the screen:

```
*0 STATUS DISPLAY
*1 DUMP ERROR BUFFER
*2 MEMORY MAP
*3 FORMAT MEDIA
*4 FLOPPY ALIGNMENT CHECK
*5 FLOPPY WRITE TEST
```

NOTE

The floppy alignment check utility requires its use of a Dyson alignment diskette, #802020, which is not supplied with the system. The diskette must be ordered from the Dyson Corp. - Santa Clara, Calif.

Press the key associated with the test you wish to run and answer the following query:

Are You Sure (Y,N)? ___

a. **Status Display.** This option allows the user to review all device parameters and statistics of the systems by examining the current status register. When this option is selected, the current status counters are displayed and the following query is asked:

Reset Status Counters (Y,N) ___

A Yes response, resets all status counters to zero and then returns to the command mode menu. A No will only return to the command mode menu.

b. **Dump Error Buffer.** When the dump error buffer option is selected, it prints all of the previously displayed error messages stored in the circular buffer on the screen. The buffer can be cleared after it is displayed by this command.

c. **Memory Map.** When this option is selected, it prints all of the memory addresses contained in the first 64K of memory and the I/O page on the screen.

d. **Format Media.** This option allows the user to format the media. The floppy may be formatted only in double density. Floppies may also be formatted through the RT-11 Format Utility. The Winchester however, may not be formatted within RT-11. To format the Winchester you must use this utility or ODT.

CAUTION

Formatting is data destructive and should be used with great care.

When this option is selected, the following status routine display and query is printed on the screen.

LG UNIT	UNIT	DEVICE	FLAGS	#ERRORS	#RD/WT	#XFERS	#DATA ERRORS
1	0	W WE	TD ID	0	0	0	0
2	1	W WE	TD ID	5	36	36	5
3	0	F WE	TD ID	0	470	235	0

Reset Status Counters (Y,N)? ___

Enter the logical unit number of the device to be formatted (0-2) ___

Enter the value and press either the carriage return key or the space bar.

Format Media Are You Sure (Y,N)? ___

A No response aborts the routine and returns the user to the command mode menu. A Yes response checks to see if the device is write protected. If the device is write protected, the following message is printed on the screen:

* Device Is Write Protected *

Type any character to proceed.

When a character is typed, the user is returned to the command menu. If the device is not write protected and the device is a floppy drive, the following query is printed:

Set Density To (S,D) ___

NOTE

Single density is not supported on the 5¼" drive. If this option is selected the test issues an error message and returns to the command mode menu.

The formatting process starts as soon as the key is pressed. Approximately one minute later the message "Format Complete" is printed on the screen. Type a character when ready to proceed.

If the device is not write protected and the device is a Winchester, the following query is printed on the screen.

Format Media Are You Sure (Y/N)? ___

A NO response aborts the routine and returns the user to the command mode menu. A YES response results in the following query:

Interweave Factors 2 (Y,N)? ___

A NO response results in an interweave factor of 3. Note that pressing either the carriage return key or space bar also results in an interweave factor of 3. Formatting takes approximately 1½ minutes.

When formatting is completed the following messages are printed on the screen:

Format Complete
Type any character when ready to proceed.

When a character is typed, the user is returned to the command mode menu.

e. **Floppy Alignment Check.** A floppy alignment utility is supplied which supports a special "alignment diskette" part number 802020. Note that this diskette is not supplied with the system and must be purchased from the Dysan Corp., Santa Clara, Calif. This utility should be used if extensive testing on the floppy yields errors attributable to drive alignment.

CAUTION

The alignment disk is for read only. Extreme caution should be used to assure this diskette is not written on.

The alignment diskette has several tracks written with a special data pattern which can be used to check certain drive parameters. Before the floppy alignment check is performed, run the WINFLP diagnostic, remove the WINFLP diskette and mount the alignment disk. To check the alignment of the floppy perform the following steps.

Step 1. Step through WINFLP "Setup Configuration" to enable floppy for testing as shown below. Then go to "Utility Functions" and select (4) Floppy Alignment Check, select drive to be tested and select sub-test (2) Track Alignment. Type 'S' to start, as instructed.

Step 2. Connect Channel 1 probe to TP1 on PWA (printed wiring assembly). Connect Channel 2 probe to TP2 on PWA.

Step 3. Connect the external sync probe to Index, J1-08 on PWA or U23 pin 6 on PWA.

Step 4. Set up scope as follows:

Channel 1 Scale:	.1 V/div
Channel 2 Scale:	.1 V/div - Inverted
Channel 1 Coupling:	AC
Channel 2 Coupling:	AC
Vertical Mode:	Add
Trigger Mode:	Normal
Trigger Source:	External
Trigger Coupling:	Low Frequency
Trigger Slope:	Negative
Timebase:	20 ms/div

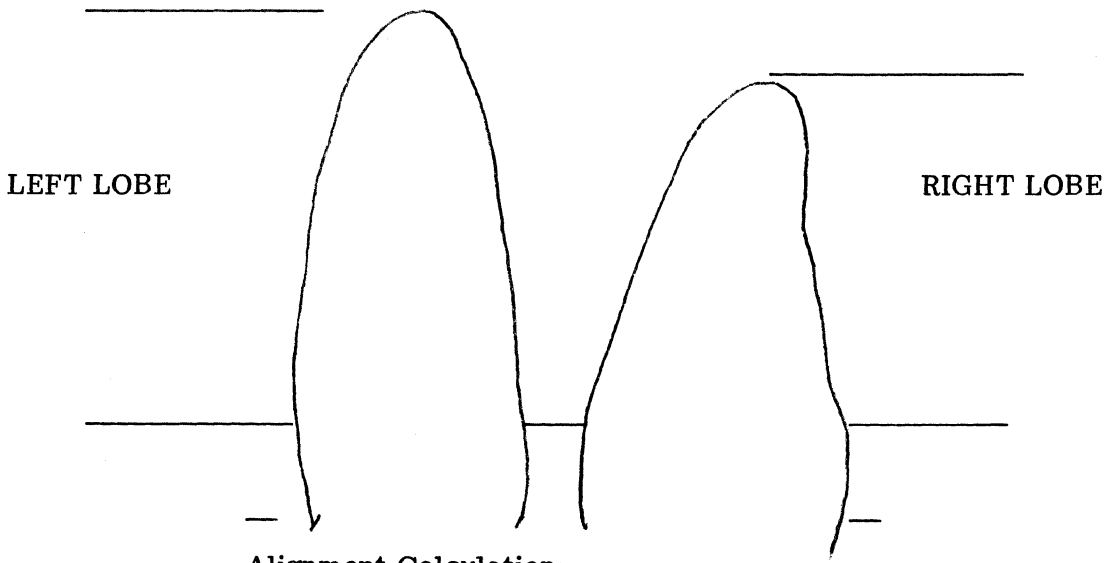
NOTE

Set ground level at one division up from lowest graticule. Adjust the channel Scales so that the top half of the cat's eye lobe (larger one in Figure 5-1) divisions from the zero point.

Before adjusting anything on the alignment, check the speed of the drive.

The outer ring of strobe marks on the spindle pulley should appear to stand still. If not, adjust pot R63 next to the power transister (Q1) until the marks appear to stand still.

With the drive flat on the bench (drive pulley down) check the alignment. If the cats eye lobes (refer to Figure 5-1) are within 5% of one another go to step 5.



Alignment Calculation:

$\frac{\text{L LOBE}}{\text{R LOBE}}$ 1 100% = (105% - 90%) Positioning Accuracy

Figure 5-1. Cat's Eye Pattern

Step 5. If the drive is more than 5% out of alignment, tilt the drive up (door up, back end down), loosen the two hold down nuts on the stepper motor and carefully rotate the motor until the alignment is within 5%. The setting may have to be off a small amount so that when the nuts are tightened and the drive set back down on the table the alignment settles back to <5%.

Step 6. Reseek the head to the alignment track by typing → (right arrow), L. This seeks the drive to track 0 and back to the alignment track. Verify that the drive is still in alignment. If not, go back to step 5. If okay, seek the other direction by typing ←(left arrow), L. The head should still be within 5%. If not, go back to step 5.

Step 7. To stop the alignment test, press T on keyboard and go back to Utility Functions. Select Floppy Alignment again and then select Index Burst Check.

Index to Burst Check and Adjustment. Select utility function then floppy alignment (all within WINFLP). Select subtest, "Index Burst Check". When 'S' is typed on keyboard, drive seeks to appropriate track (68). To perform the check and adjustment, perform the following steps.

Step 1. Set scope as follows:

Channel 1 on TPI on the PWA
Channel 2 to Index, J1-8 of PWA or U23 pin 6 on PWA

Channel 1 Scale:	.1 V/div
Channel 2 Scale:	2V/div
Channel 1 Coupling:	AC
Channel 2 Coupling:	AC
Vertical Mode:	Chop
Trigger Mode:	Normal
Trigger Source:	Channel 1
Trigger Coupling:	Low Frequency
Trigger Slope:	Negative
Timebase:	50 µs/div

Step 2. Verify that the leading edge of the index pulse (low going) to the burst pulse measures 200 ± 100 μs. If time is 200 ± 100 μs, it is acceptable then check is concluded. If time is not 200 ± 100 μs, proceed to step 3.

Step 3. Remove PWA without disconnecting cables or harnesses. Electrically isolate PWA from the frame of the casting.

Adjust the time from the leading edge of the index pulse to the burst pulse until it measures 200 ± 40 μs. To adjust the index to burst time, loosen the ½ inch hex head screw holding the photo transister located on top of the clamshell. Using a flat black non-magnetic screwdriver, engage the notch in the phototransister housing and move the photo transister until the specification is met. Tighten the screw while observing the scope signal. Verify that the adjustment still meets the ± 40 μs tolerance. While observing the signal on the scope, remove and reinsert the diskette three times.

Verify that the difference between the minimum and maximum index to burst time measured after each insertion is less than 100 μs. If the change is greater than 100 μs the cone and spindle are defective and must be replaced. See Figure 5-2.

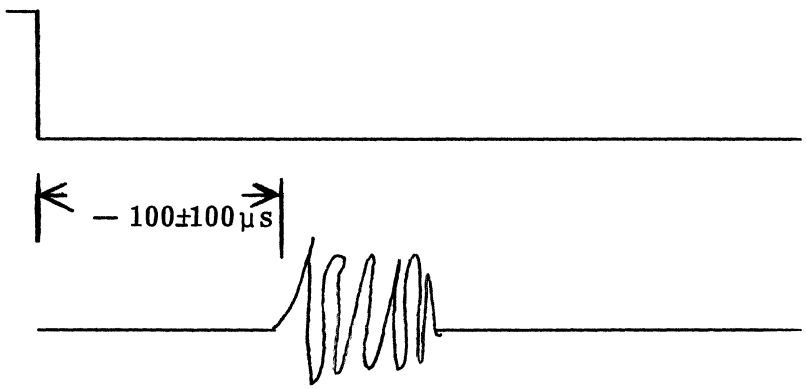


Figure 5-2. Index to Burst

Read Amplitude Test. Select utility function then floppy alignment test. Then select subtest 'Read Amplitude Test'. To perform this test perform the following steps.

Step 1. Connect channel 1 probe on TP1 to Drive PWB. Connect Channel 2 probe on TP2. Connect external size probe to Index J1.08 on PWA or U23 pin L on PWA.

Step 2. Set scope as follows:

Channel 1 Scale:	.2 V/div
Channel 2 Scale:	.2V/div - Inverted
Channel 1 Coupling:	AC
Channel 2 Coupling:	AC
Vertical Mode:	Add - ground level on center graticule.
Trigger Mode:	Normal
Trigger Source:	External
Trigger Coupling:	Low Frequency
Trigger Slope:	Negative
Timebase:	40 μs/div

Step 3. Insert Read Amplitude Reference Disk and type any character to start test.

Step 4. Verify that read amplitude (on scope) is within track 0 limits (700 - 1300 mv) and that variation is less than 20%.

Step 5. Type C to step to track 50 and repeat step 4 above with limit of 600 - 1300 mV.

Step 6. Type C to step to track 114 and repeat step 4 above with limit of 600 - 1200 mV.

Step 7. Type C to end test and go back to floppy alignment menu.

Exit. Returns control to the RT-11 operating system.

f. **Floppy Write Protect Test.** This test requires the use of a write enabled diskette and a write protected diskette. When this test is selected, the following text, status routine display and queries are printed on the screen.

The write protect switch on the floppy drive is checked by first writing to a write enabled diskette and then by attempting a write on a write protected diskette. Prompts are given for required operator actions.

Floppy Write Protect Test Starting

LG Unit	Unit	Device Flags	# Errors	# RD/WT	# XFERS	# Data Errors
1	0	W WE TD ID	0	0	0	0
2	1	F WE TE IE	0	0	0	0

Reset status counters (Y,N)? Y
Enter the logical unit number of the desired device 2
Verify selected device - Are you sure (Y,N)? Y
Place write enabled diskette in selected drive.
Type a character when ready to proceed
Remove write enabled diskette and install a write protected diskette.
Type a character when ready to proceed
Diskette is write protected or switch is bad - Aborting Test
Non-write protect error - check drive and media-Aborting Test
Error code = 0
Type a character when ready to proceed (Return to main menu)

Repeat the floppy write protect test as shown below.

Floppy Write Protect Test Starting

LG Unit	Unit	Device Flags	# Errors	# RD/WT	# XFERS	# Data Errors
1	0	W WE TD ID	0	0	0	0
2	1	W WE TD ID	0	0	0	0
3	0	F WE TE IE	0	0	0	0

Reset status counters (Y,N)? Y
Enter the logical unit number of the desired device 2
Verify selected device - Are you sure (Y,N)? Y
Place write enabled diskette in selected drive.
Type a character when ready to proceed
Remove write enabled diskette and install a write protected diskette.
Type a character when ready to proceed
Remove write protected diskette - test is complete
Type a character when ready to proceed

5.6 FRONT PANEL TEST

The front panel test provides a manual check on all front panel switches (buttons), LED indicators and a small portion of the DLV11-J serial interface to which the front panel is connected. Note that this test does not utilize any error counters or automated tests.

5.6.1 Loading and Running Instructions

Step 1. Mount the diagnostic diskette #2515-0691 on drive DY0:.

Step 2. Power up and boot the system as described in the 2515 system description and operation manual (2515-0100).

Step 3. In response to the prompting dot (.) enter **Run FPLTST (CR).** After you have entered the run command the following information and main menu are printed on the screen:

```
GenRad BTA front panel test program

CTRL-C returns to RT-11
CTRL-R aborts function and returns to command mode

All numeric inputs/outputs are in octal

Type a character when ready to proceed

FPLTST V2A - Internal Audit Version

        * 0     Help Text
        * 1     Setup Utility
        * 2     Key Test 1
        * 3     Key Test 2
        * 4     LED Test 1
        * 5     LED Test 2

  Enter help to enable additional text during testing
  Enter setup mode to verify panel interface channel
  Key Test 1 is a random key check
  Key Test 2 prompts for specific keys
  LED Test 1 checks all LEDs simultaneously
  LED Test 2 checks individual LEDs

  Enter desired selection (0-5) ___
```

5.6.2 Main Menu

```
* 0     Help Text
* 1     Setup Utility
* 2     Key Test 1
* 3     Key Test 2
* 4     LED Test 1
* 5     LED Test 2

  ENTER desired selection (0-5) ___
```

To select one of the routines listed in the main menu shown above, press the key associated with the desired routine. After one of the keys (0-5) is pressed, the following query is printed on the screen:

Are you sure (Y,N) ___

A description of each routine contained in the main menu is provided in the following paragraphs.

5.6.2.1 Help Text. When the help text routine is selected, the system prints the following general information on running the FPLTST test which should be reviewed before proceeding.

- Restart addresses are 1104 or 1114 octal
- CTRL C allows return to RT-11 if available
- CRTL D transfers control to ODT, single CTRL C returns
- CTRL 0 aborts output to screen for duration of current command
- CTRL R aborts current test and returns control to command mode
- CTRL S stops output to terminal - type any other character to resume typing
  Good Luck!

Enable additional help text during testing (Y,N) ? ___

A Yes response causes a short description of each test to be displayed before each test is run.

5.6.2.2 Setup Utility. The front panel is normally connected to a DLV11-J serial interface on channel zero with a base address of 77650. This utility should be used before any other testing to verify that the front panel is connected to the DLV11-J serial interface. The user is prompted to push any button on the panel when a possible channel number has been found.

Example:

Push a front panel button to verify channel ___
Panel connected to channel: 0
At CSR base address: 176500

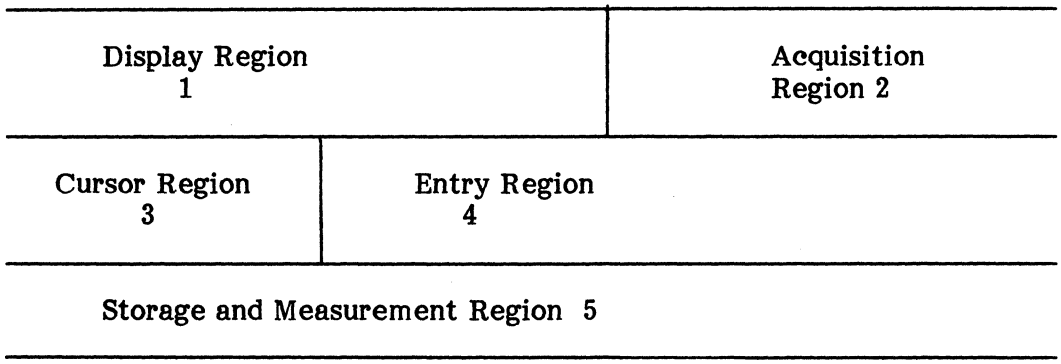
Type any character when ready to proceed (RETURN to main menu) ___

At this point the program is aware of this address and uses it during testing. An error message is printed if the channel cannot be identified which may indicate that no connection is made. Check all cables and jumpers on the interface if this occurs.

5.6.2.3 Key Test #1. This key test allows the user to verify that the keys on the front panel are operational. The following information and diagram are printed on the screen as soon as this test is started.

Select one of the following regions:

1 Display Region
2 Acquisition
3 Cursor Region
4 Entry Region
5 Storage and Measurement Region



Select one of the five regions on the panel for testing. Buttons in this region may be pushed and the program identifies which code was received and prints the associated button label (refer to Table 5-4). To terminate the test, type a T on the main keyboard. Additional regions may be selected at a future time.

Enter the region number (1-5) ___

Once a region is selected only buttons in this region will be accepted. Any buttons pushed in other regions are identified as "invalid" and the octal code displayed. After a region is selected the test prints the following message indicating it is starting:

Front Panel Test 1 Starting

At this point the user begins pushing buttons in the specified panel region. After each button is pushed there is a message printed on the screen indicating the button pushed (the label above the button). Press only one button at a time and check that the proper label is displayed.

Example:

OCTAL CODE : 170 BUTTON LABEL : REAL TIME
OCTAL CODE : 171 BUTTON LABEL : AVG
OCTAL CODE : 172 BUTTON LABEL : MEM
OCTAL CODE : 173 BUTTON LABEL : NEXT CHAN
OCTAL CODE : 163 BUTTON LABEL : EXCHNG
OCTAL CODE : 162 BUTTON LABEL : OVLY
OCTAL CODE : 161 BUTTON LABEL : DUAL
OCTAL CODE : 160 BUTTON LABEL : FULL

If an error should occur during this test, record the error information and refer to paragraph 5.6.3 for troubleshooting.

Table 5-4. Button Labels and Octal Codes

BUTTON LABEL	OCTAL CODE
Region 1 - Display	
Y Scale	24
Block Math	44
Down Arrow	154
Channel	156
Units	157
Dual	160
Ovly	162
Exchange	163
Next	164
Real Time	170
AVG	171
MEM	172
Next Channel	173
Up Arrow	174
Function	176
X Scale	177

Table 5-4. Button Labels and Octal Codes (Continued)

BUTTON LABEL	OCTAL CODE
Region 2 - Analysis	
CONT	26
Run	27
Arm	46
Hold	47
Region 3 - CURSOR	
Sideband	120
Inward Arrows	122
Outward Arrows	123
Harmonic	130
Store	132
Recall	133
Main	140
REF Set	142
ABS/REL	143
Left Arrow	150
Right Arrow	152
Region 4 - CURSOR	
- E	55
SPC F	56
SHIFT	57
J	65
R DOWN X	66
X L	67
X P	75
+/- Q	76
Y R	77
/ V	105
= W	106
Z :	107
O S	114
. T	115
, U	116

Table 5-4. Button Labels and Octal Codes (Continued)

BUTTON LABEL	OCTAL CODE
Region 4 - Analysis	
ENTER	117
1 M	124
2 N	125
3 O	126
DEL	127
4 G	134
5 H	135
6 I	136
CLR	137
7 A	144
8 B	145
9 C	146
EE D	147
Region 5 - Storage and Mesaurement	
Output Level	0
Up Arrow	1
Output Signal	2
CRT BRT	20
State	30
FREQ	32
Level	40
TRIG	42
Window	50
CHANS	52
Average	60
Mode	62
Recall	70
Down Arrow	73
Store	100
Next	103
TRIG	110
DIR	111
Up Arrow	113

5.6.2.4 Key Test #2. This test allows the user to verify that the keys on the front panel are operational. The following information and diagram are printed on the screen as soon as the test starts.

Select one of the following regions:

1 Display Region
2 Acquisition Region
3 Cursor Region
4 Entry Region
5 Storage and Measurement Region

Display Region 1	Acquisition Region 2
Cursor Region 3	Entry Region 4
Storage and Measurement Region 5	

Enter the region number (1-5) ___

After a region is selected the test prints a message indicating it is starting. At this point the program prompts the user to push a specific button in the selected panel region. Pushing any other button results in an error message indicating an invalid key was pushed. The test loops through all of the buttons in the specified region and prints a message indicating the test is complete.

Example:

```
FRONT PANEL TEST 2 STARTING

PUSH THE 'Y SCALE' BUTTON
PUSH THE 'BLOCK MATH' BUTTON
PUSH THE 'DOWN ARROW' BUTTON
PUSH THE 'CHANNEL' BUTTON
PUSH THE 'UNITS' BUTTON
PUSH THE 'SINGLE' BUTTON
PUSH THE "DUAL' BUTTON
PUSH THE 'OVLY' BUTTON
PUSH THE 'EXCHANGE' BUTTON
PUSH THE 'NEXT' BUTTON
PUSH THE 'REAL TIME' BUTTON
PUSH THE 'AVG' BUTTON
PUSH THE "MEM' BUTTON
PUSH THE 'NEXT CHAN' BUTTON
PUSH THE 'UP ARROW' BUTTON
PUSH THE 'FUNCTION' BUTTON
PUSH THE 'X SCALE' BUTTON

* TEST COMPLETE *
TYPE A CHARACTER WHEN READY TO PROCEED
```

Type any character at this point to return to the main menu. To terminate the test before all buttons in the region have been checked, press T on the main keyboard. If an error occurs during this test, record the error information and refer to paragraph 5.6.3 for troubleshooting.

5.6.2.5 LED Test #1. Test #1 provides a visual check on all 14 LED's on the front panel. When started, the program clears all of the LED's on the front panel and prompts for user verification of this.

Example:

LED TEST NOW STARTING

VERIFY THAT ALL LEDS ARE CLEARED

TYPE A CHARACTER (on the main keyboard only) WHEN READY TO PROCEED

When a key is pressed the program sets (lights) all 14 LED's on the front panel and prompts the user for verification of this. Make note of any LED's that are not on and type a character (on the main keyboard only) to clear all of the LED's on the front panel and exit to the main menu. If an error(s) occurs during this test, record the error information and refer to paragraph 5.6.3 for troubleshooting.

5.6.2.6 LED Test #2. Test #2 checks each LED on the front panel individually and runs until the user types T to terminate the test. When started the program prompts the user to push a button nearest an LED to toggle its state. The buttons to toggle are as follows:

Display	Acquisition	Cursor	Entry	Measurement
REAL TIME	RUN	MAIN	SHIFT	CRT BRT
AVG	ARM	HARMONIC		
MEN	HOLD	SIDEBAND		
SINGLE				
DUAL				
OVLY				

Each press of a button changes the LED from off to on or vice versa. Holding the button down causes an autorepeat mode which makes the LED flash at a fast rate. This also causes overrun errors on the interface and periodic INVALID KEY messages are printed. If an error(s) should occur during this test, record the error information and refer to paragraph 5.6.3 for troubleshooting.

5.6.3 Troubleshooting

A list of symptoms and corrective actions is provided below:

Symptoms	Corrective Action
Setup utility yields "check DLV11-J" message	Check all jumpers on the DLV11-J interface board (refer to paragraph 2.5.2), the cable to front panel, and intermediate connections. Replace DLV11-J and interface cable. If the problems persist, replace front panel. Be certain that at least 12 different buttons have been tried.
Invalid button message with a code having 140000 bits set	These 2 error bits indicate either an overrun error or that some button is being held down too long. This may indicate that the edge of the button hole is not releasing the button once it is pressed. Make note of the low 7 bits of the key code and refer to Table 5-4 to find which button may be suspect. Adjust the front panel so that all buttons may move freely. Repeat the test and make adjustments until this error is not reported.
Small number invalid button messages (1-6 total)	This indicates that some portion of the panel including the buttons may be faulty. Check every button on the panel and make a list of errors to include the button pressed, expected key code, and bits stuck high or low to isolate individual bit paths.
Large number of invalid button messages (greater than 6 total)	Make note of faulty buttons or "working" ones if this is easier. Look for consistantly stuck bits in actual and expected codes to isolate individual bit paths. Check physical connections to panel including power and ground. Check panel circuit starting at interface side and proceeding to panel side.
Several LED's do not turn ON	Run LED test #1 to turn all LED's ON and verify a voltage at leads. If voltage is present, then replace LED's. If voltage is not present, then trace back to where the problem originates.

None of the LED's turn ON	Start by checking voltage at each LED and trace back to where problem originates. Run the LED tests and verify that codes are being sent by the interface and received by the front panel. Replace components if they seem to be faulty.

5.7 FRONT-END DIAGNOSTIC

The front-end diagnostics are used to test the filter board, channel board, one part of the memory controller board and the analog output circuitry on the utility board.

5.7.1 Loading and Running Instructions

Step 1. Mount the diagnostic diskette #2515-0691 on drive DY0:.

Step 2. Power up and boot the system to DY0: as described in the 2515 Computer-Aided Test System Operating Manual (2515-0100).

Step 3. In response to the prompting dot (.) enter

RUN ANLTST (CR)

After the run command is entered the main menu is displayed.

5.7.2 Main Menu

```
=>  0.  TEST SETUP (for Production test use only)
    1.  HELP INSTRUCTIONS
    2.  AUTOMATIC TEST - Not implemented at this time.
    3.  CHANNEL TEST
    4.  FILTER CUTOFF TEST
    5.  TRIGGER TEST
    6.  DIGITAL FILTER TEST
    7.  DITHER TEST
    8.  MANUAL MODE (for Production test use only)
    9.  EXIT TO RT-11
```

To select one of the routines listed in the main menu shown above, position the pointer (⇒) next to the desired routine using either the up (↑) and down (↓) arrow keys or by pressing the numeric key associated with the routine on the keyboard. Once the pointer is positioned next to the desired routine, the routine is started by pressing the right arrow (→) key. To exit from any of the routines (except the manual mode routine) and return to the main menu, press the ESC key. A description of each routine contained in the main menu is provided in the following paragraphs.

5.7.2.1 Test Setup. This routine is reserved for manufacturing test purposes only.

5.7.2.2 Help Instructions. When the help instruction routine is selected, the system displays the following general information on the routines listed in the main menu:

Test Setup Routine	The test setup routine is used for setting up special switches for inhouse testing and debug of this program. Entry into this routine can cause numerous problems for the unwary!
Automatic Test	The automatic test will automatically run all of the other tests and check the data, giving a pass or fail indication at the end of the test. This test is not functional at this time.
Channel Test	The channel test is used to test the analog portion of the front end including gain settings, ac or dc coupling, bias current, zeroing and dc removel.
Frequency Cutoff Test	The frequency cutoff test checks the cutoff frequency of the digital filter by setting the input frequency, first above the cutoff frequency and verifying that the signal is cut off, and then setting the input frequency below the cutoff frequency and verifying that the input signal is not attenuated.
Trigger Test	The trigger test exercises the trigger circuitry by triggering the front end at several levels and at several frame delays. The frame delay is the number of points from the trigger point to the beginning of the data/display buffer, (- = pre-trigger, + = post-trigger). The trigger level range for ±100% on the display is 4 divisions each direction from the center line. Since the trigger is digital and common to all input channels on the same board, it is necessary to test only one channel.
Digital Filter Test	The digital filter test is a general health test on the digital filter. By turning on the dc bias with nothing connected to the inputs and the input coupling set to dc, a dc overload condition is generated. A zoom sample is then taken on overload. The zero frequency point is at the center of display. If there is any peak at this point above 80 dB, (2 divisions), there is a problem in the digital filter.
Dither Test	The dither test checks the dither circuit on the 25 kHz board. The inputs are left open and a sample is taken with the filter set to 0 decimation and the dither circuit disable (for reference). Then the dither circuit is enabled and a second sample is taken. A high frequency random noise type signal should be visible on the display.

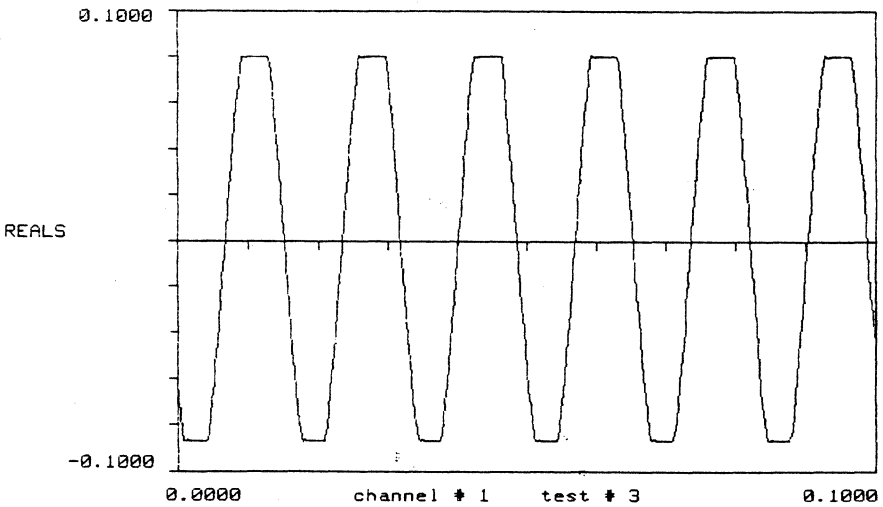
Manual Mode for Front — This set of manual routines is for use by those familiar with the operation of the front end. It is necessary to set up the data stream first, then the channel(s), and then any extra options before selecting the go check option.

Return to RT-11

End Diagnostic

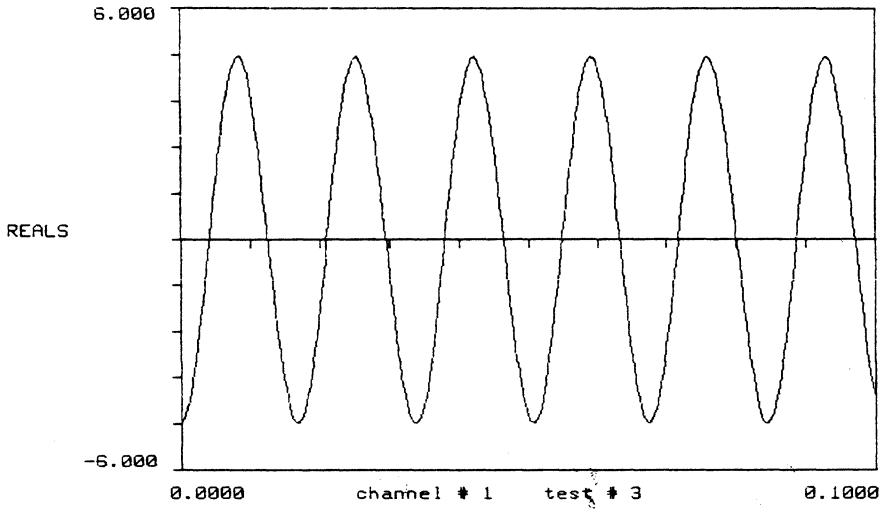
5.7.2.3 Automatic Test. The automatic test runs all of the tests in the main menu once. If one of the tests should fail, the acceptance test continues through all tests and displays the results at the end of the test. (This test is not functional at this time.)

5.7.2.4 Channel Test. Before the channel test is executed, you must have a sine wave source with the oscilloscope tied in to monitor the signal level input to each channel. Starting with channel 1, the input level should be set as directed. The results should be a clipped (squared off) sine wave for the first six levels and a sine wave for the last two levels. Then the test goes to the next input channel. See example of good displays below.



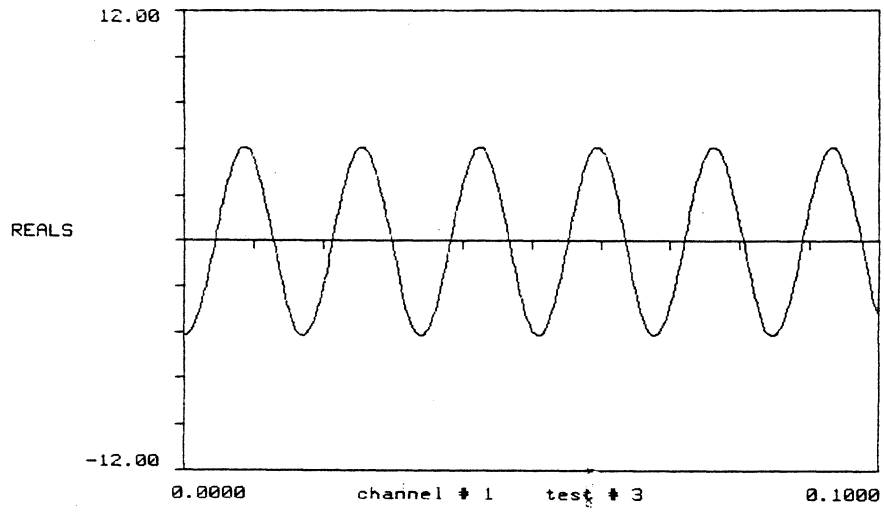
there should be a clipped signal of approximately
8 divisions (peak -> peak).

ESC returns to main menu-: press space bar to continue



there should be a signal of approximately
8 divisions (peak -> peak).

ESC returns to main menu-: press space bar to continue



there should be a signal of approximately
4 divisions (peak -> peak).

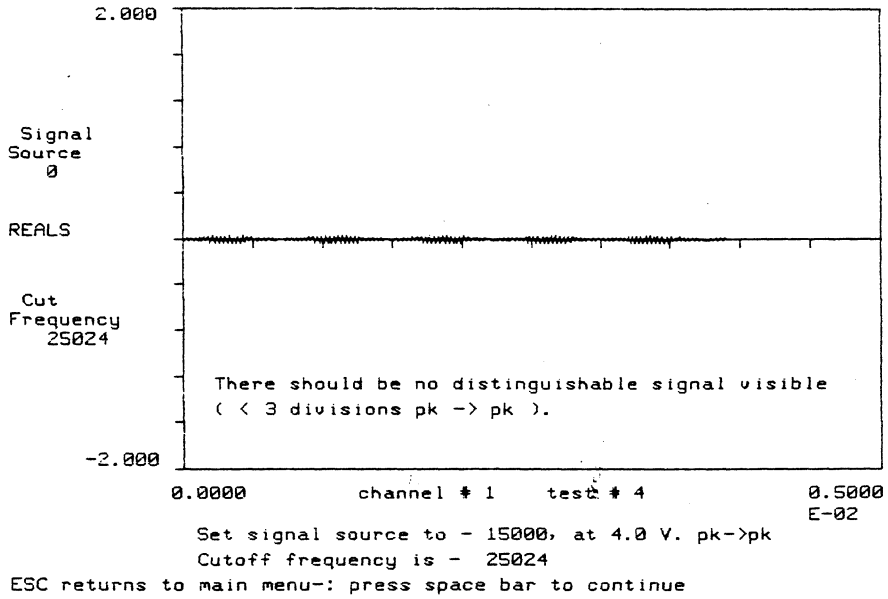
ESC returns to main menu-: press space bar to continue

5.7.2.5 Filter Cutoff Test. Before the filter cutoff test is executed, you must connect an oscilloscope to INPUT CHANNEL 1 on the front panel with a BNC-to-BNC patch cable. Then you are prompted to set the input signal to 4.0 volts peak to peak and the following frequencies:

Input Signal	Cutoff Frequency	Time
32,000Hz	25,024Hz	11 sec.
15,000Hz	25,024Hz	11 sec.
15,000Hz	10,240Hz	11 sec.
7,500Hz	10,240Hz	11 sec.
7,500Hz	5,120Hz	11 sec.
4,000Hz	5,120Hz	11 sec.
4,000Hz	2,560Hz	11 sec.
2,000Hz	2,560Hz	11 sec.
2,000Hz	1,280Hz	12 sec.
1,000Hz	1,280Hz	12 sec.
1,000Hz	640Hz	12 sec.
600Hz	640Hz	12 sec.
600Hz	320Hz	12 sec.
300Hz	320Hz	12 sec.
300Hz	160Hz	12 sec.
150Hz	160Hz	12 sec.
150Hz	80Hz	12 sec.
75Hz	80Hz	13 sec.
75Hz	40Hz	13 sec.
36Hz	40Hz	15 sec.
36Hz	20Hz	15 sec.
18Hz	20Hz	20 sec.
18Hz	10Hz	20 sec.
9Hz	10Hz	28 sec.
9Hz	5Hz	28 sec.
4Hz	5Hz	47 sec.
4Hz	2.5Hz	45 sec.
2Hz	2.5Hz	80 sec.

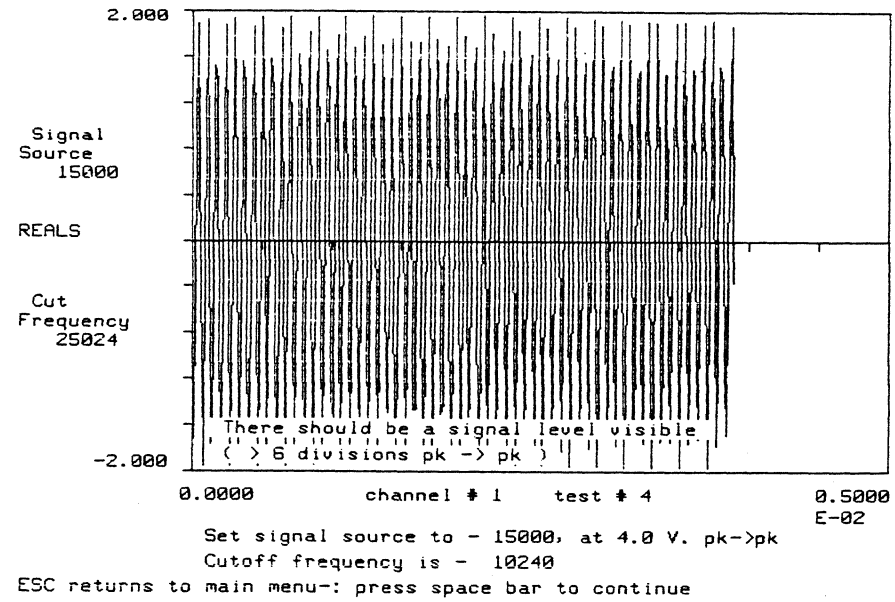
Note that if the system contains more than four input channels, this test must be repeated on the first channel (5, 9 or 13) on each additional 25 kHz board.

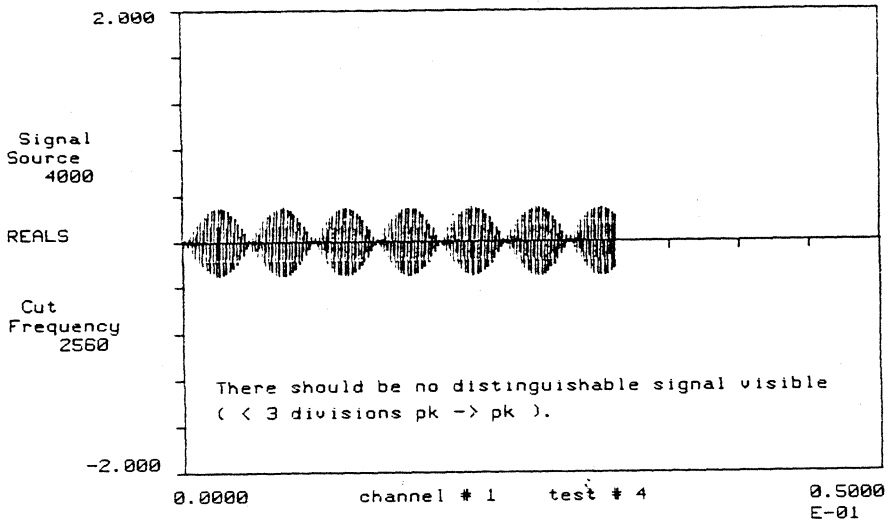
The filter cutoff test checks the cutoff frequencies of the digital filter by setting the cutoff frequency. It then prompts the user to set the input signal above the cutoff frequency and prints a display on the screen similar to the one shown below:



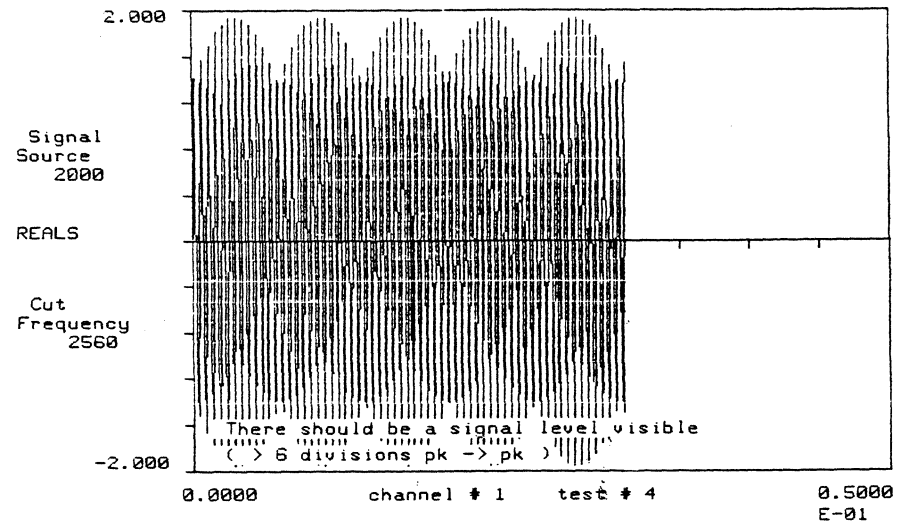
Typical Good Digital Filter Test

The user is prompted to hit the space bar to proceed. When the space bar is hit, the user is prompted to set the input frequency below the cutoff frequency and a display similar to the ones shown below is printed on the screen:





Set signal source to - 2000, at 4.0 V. pk->pk
Cutoff frequency is - 2560
ESC returns to main menu-: press space bar to continue



Set signal source to - 2000, at 4.0 V. pk->pk
Cutoff frequency is - 1280
ESC returns to main menu-: press space bar to continue

This process is repeated for each of the input signals and cutoff frequencies listed above.

5.7.2.5 Trigger Test. Before the trigger test is executed, you must connect an oscilloscope set to output a 60 Hz signal to the front panel INPUT CHANNEL 1 with a BNC-TO-BNC patch cable. The trigger test exercises the trigger circuitry by triggering the front end at several levels and at several frame delays. The trigger level is presented as a percentage of the full scale value (± 4 volts). Delay is the number of points acquired in a frame before triggering occurs. Each frame consists of 512 points.

Example:

If a delay of 0 points is utilized, triggering occurs at the first point of the frame. If a delay of +511 points is utilized, triggering occurs at the last point in the frame.

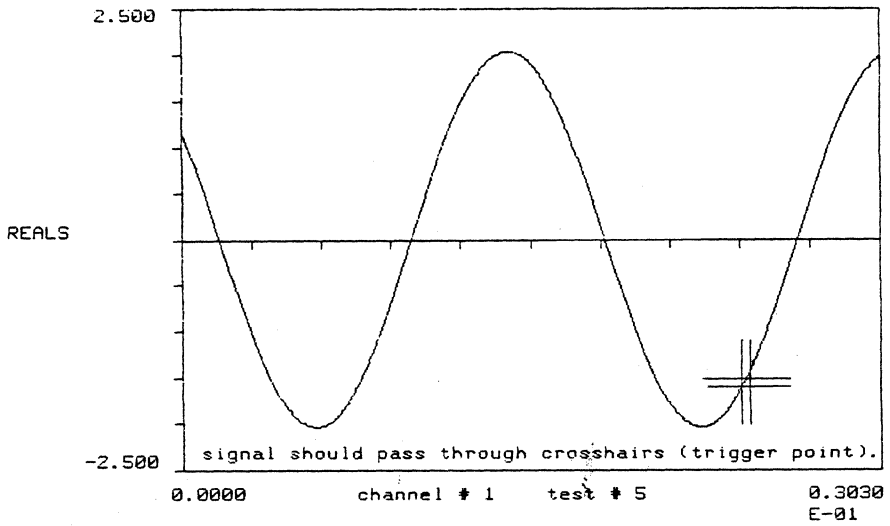
Since the trigger is digital and common to all input channels on the same 25 kHz board, it is necessary to test only one channel per 25 kHz board. Only the first channel on each 25 kHz board (channels 1, 5, 9 or 13) may be used in conjunction with the trigger test.

Triggering will occur twelve separate times at various trigger level percentages, delays and slopes during the trigger test. Each time triggering occurs, a display is printed on the screen. A crosshatch (+) is contained within each display with the triggering point within the crosshatch. The user must compare each display printed on the screen to the examples shown below. If any of the displays shown on the screen do not resemble the examples shown below, refer to the troubleshooting procedures at the end of this paragraph.

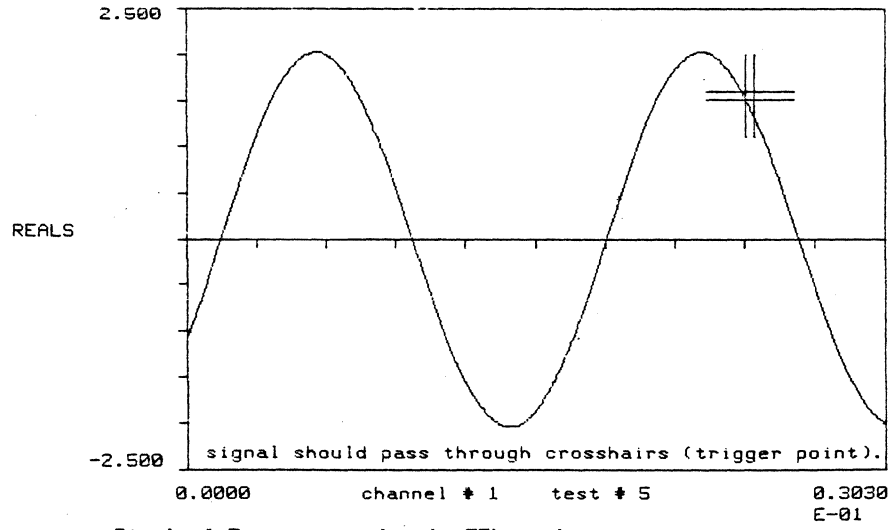
When the trigger is selected from the main menu, the following information and prompt are printed on the screen:

TRIGGER TEST

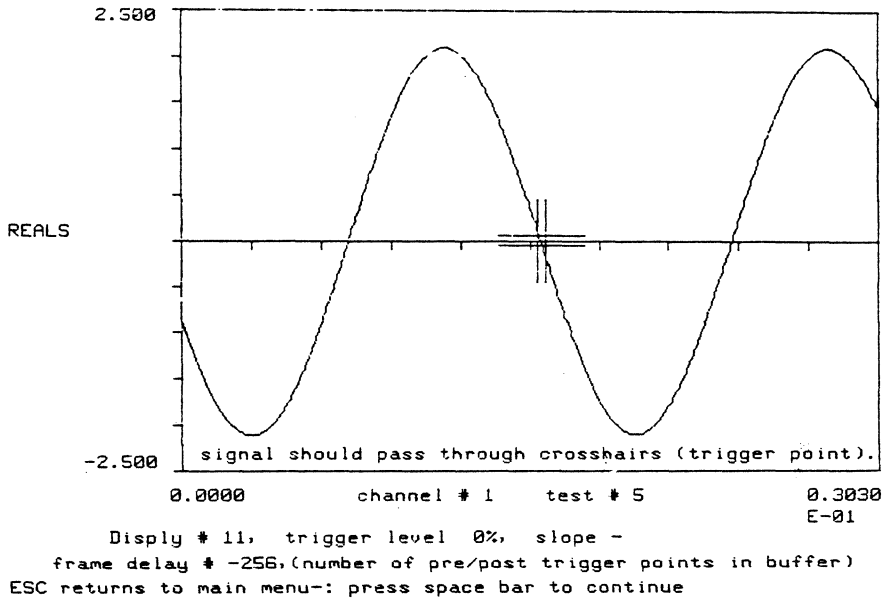
There are 12 displays for the trigger test. Several typical examples are shown below.



Disply # 1, trigger level -75%, slope +
frame delay # -400,(number of pre/post trigger points in buffer)
ESC returns to main menu-: press space bar to continue



Disply # 7, trigger level 75%, slope -
frame delay # -400,(number of pre/post trigger points in buffer)
ESC returns to main menu-: press space bar to continue



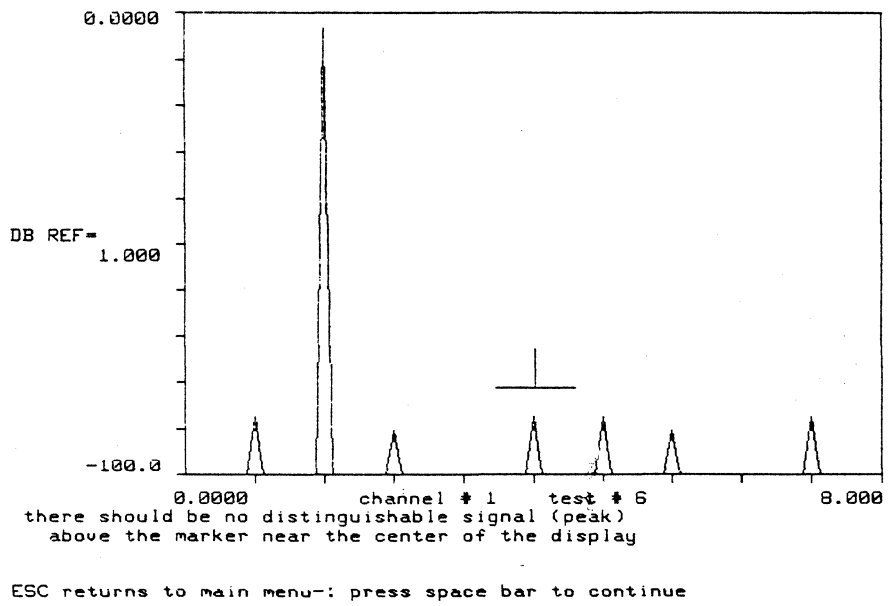
5.7.2.6 **Digital Filter Test.** The digital filter test performs a general health test for the digital filter circuitry. When the digital filter test is started a dc overload condition is generated by turning on the dc bias current with the coupling on dc and the inputs open. A zoom sample is then taken on the overload condition and displayed on the screen. The zoom point is at the center of the display. If there is any peak at this point above 80 dB (2 division marks), there is a problem in the digital filter circuitry. This test must be run on all four channels to test all of the digital filter circuitry. When the digital filter test is selected from the main menu, the following prompt is displayed on the screen:

Disconnect All Inputs, -:

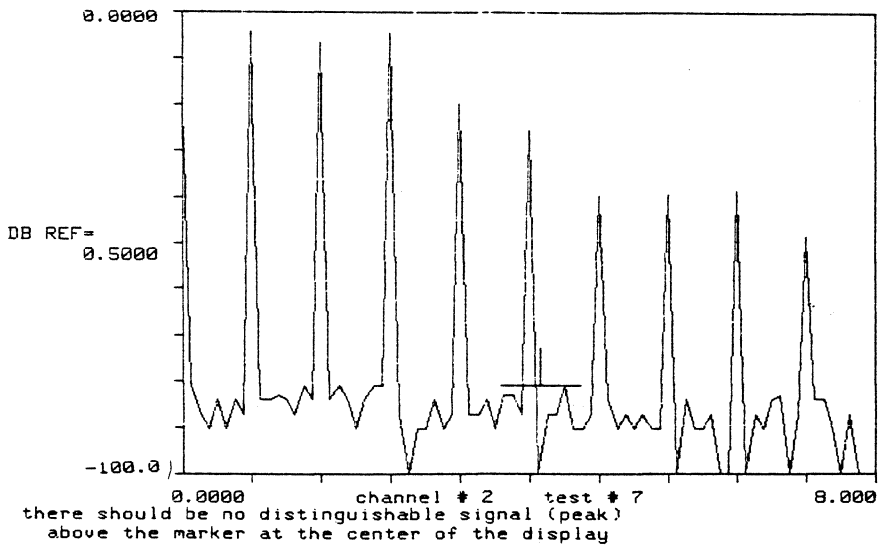
press the space bar to continue.

The message "WORKING" will appear in the upper left-hand corner of the screen for approximately 30 seconds after the space bar and then be replaced by a display similar to the typical good display shown above. Repeat the above process for each channel in the system. When the last channel is tested the main menu is displayed on the screen.

A typical good display is shown below:



Typical Good Display

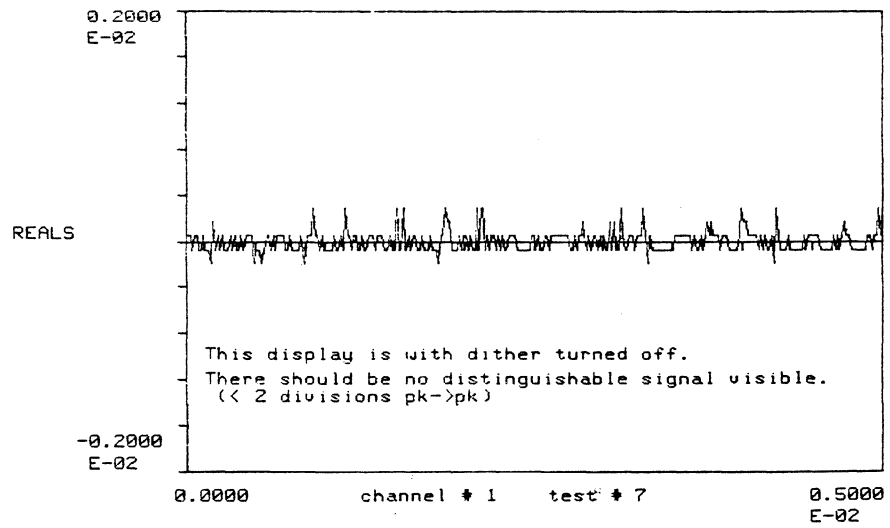


Typical Bad Display

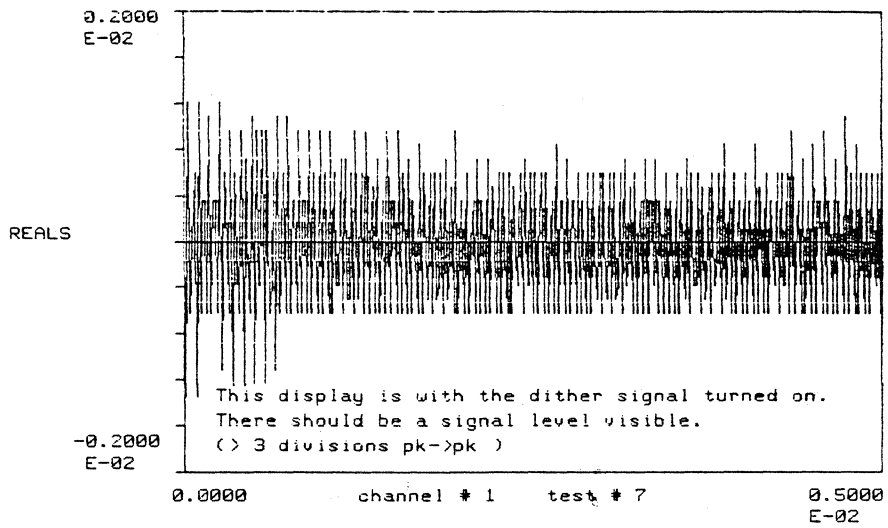
Troubleshooting.

Symptom	Corrective Action
Only one of the four channels tested fails.	Replace the digital filter board. If the error persists replace the memory controller board.
All of the channels tested fail.	Replace the digital filter board.

5.7.2.7 Dither Test. The dither test checks the dither circuit on the 25 kHz board. The inputs are left open and a sample is taken with the filter set to 0 decimation and the dither circuit disabled (for reference). Then the dither circuit is enabled and a second sample is taken. A high frequency random noise type signal should be visible on the display.



: press space bar to continue



: press space bar to continue

LEAVE INPUTS OPEN.

ESC returns to main menu-: press space bar to continue

5.7.2.8 **Manual Mode Test.** Reserved for manufacturing test purposes only.

5.7.2.9 **Exit.** Returns to RT-11.

5.8 INTRODUCTION TO XXDP+

The XXDP+ package is a multi-purpose group of tests designed by DEC to check the LSI family computer and options. The package has been modified for GenRad systems.

5.8.1 XXDP+ Operation

Note: The rotary switch labeled NORMAL/TEST on the back panel of the 2515 must be set to the TEST position in order to run XXDP+ tests. When the NORMAL/TEST switch is in the TEST position only XXDP+ tests will run, therefore, at the conclusion of the XXDP+ test(s) the NORMAL/TEST switch must be set to NORMAL in order to run RT-11 programs.

Start the execution of the XXDP+ Monitor by using the appropriate bootstrap program. The monitor prints a message identifying itself and requesting that the current date be entered. An example of this message is given below for the DKDP+ monitor.

```
CHMDYC0 XXDP+ DY MONITOR
BOOTED VIA UNIT 0
28K
DOES THIS SYSTEM HAVE A UNIBUS? (Y/N CR=Y) N
NON-UNIBUS SYSTEM

ENTER DATE (DD-MMM-YY):

RESTART ADDR: 152010
THIS IS XXDP+. TYPE "H" OR "H/L" FOR HELP.
```

After the date has been accepted by the monitor, the restart address of the monitor is printed.

When running any XXDP+ diagnostic on the 2515, most of the printouts are not legible because the way DEC diagnostics use resets in the programs. Error printouts are listed correctly and the diagnostics halt on error.

After the chain file has been completed, the following information is printed on the screen:

A good complete pass of the chain file is printed out as:

```
;"*****THIS COMPLETES THE 2515 CHAIN DIAGNOSTICS*****".
;
;TOTAL TEST TIME IS APPROXIMATELY 12 MINUTES.
;
$
TYPE ^X TO CONTINUE
```

```
;
*EXIT
;
;    2515 CHAIN DIAGNOSTICS
;    11/23 CPU DIAGNOSTICS (MD-11-CJKDB)
R  JKDB??/5
```

5.8.2 Obtaining a Directory

To obtain a directory of the system device, type one of the following:

D(CR)	To obtain directory on console terminal.
D/F(CR)	To obtain shortened or "fast" directory on console terminal.
C DIRECT	To obtain cross reference directory on console terminal.

The directory contains the following information:

Filnam, Ext	File name and extension assigned.
Length	Number of blocks used (decimal).
Start	Starting block number (octal).
Date	Date when file was put on medium.

The following is a sample directory of the XXDP+ diskette:

ENTRY#	FILNAM.EXT	DATE	LENGTH	START
1	HSAAD0.SYS	8-MAR-83	24	000067
2	HSADB0.SYS	8-MAR-83	25	000117
3	HUDIB0.SYS	8-MAR-83	5	000150
4	HMDYC0.SYS	8-MAR-83	12	000155
5	HDDYB0.SYS	8-MAR-83	3	000171
6	HDDLC0.SYS	8-MAR-83	4	000174
7	HMDLD0.SYS	8-MAR-83	13	000200
8	HDDKB0.SYS	8-MAR-83	2	000215
9	HMDKC0.SYS	8-MAR-83	12	000217
10	HDDXB0.SYS	8-MAR-83	3	000233
11	HMDXC0.SYS	8-MAR-83	12	000236
12	HELP1 .TXT	8-MAR-83	14	000252
13	HELP .TXT	21-OCT-83	3	000270
14	SETUP .TXT	21-DEC-83	2	000273
15	DIR .TXT	20-MAY-84	3	000275
16	SETUP1 .TXT	21-MAY-84	2	000300
17	XTECO .BIN	8-MAR-83	16	000302
18	UPD2 .BIC	8-MAR-83	25	000322
19	JKDAD1 .BIC	25-FEB-83	28	000353
20	JKDBD0 .BIC	25-FEB-83	52	000407
21	JKDCB0 .BIC	25-FEB-83	52	000473
22	JKDDB0 .BIC	25-FEB-83	32	000557
23	VMSAA0 .BIC	22-JUL-83	24	000617
24	ZRXEA2 .BIC	12-MAY-81	17	000647
25	VDLAB0 .BIC	22-JUL-83	16	000670
26	VKAFD1 .BIC	22-JUL-83	6	000710
27	DLVTST .BIC	19-DEC-83	16	000716

ENTRY#	FILNAM.EXT	DATE	LENGTH	START
28	DLV15 .BIC	29-DEC-83	16	000736
29	HELP1 .CCC	18-MAY-81	1	000756
30	HELP .CCC	21-OCT-83	1	000757
31	DIRECT .CCC	14-MAY-81	1	000760
32	2515 .CCC	28-DEC-83	2	000761
33	2515P .CCC	29-DEC-83	2	000763
34	EQBTST .CCC	21-MAY-84	1	000765

5.8.3 Loading and Running Programs

Type R and the program name. Do not type the extension. Note that the actual file must have a BIN or BIC extension. This will load and run the program. To just load the program, type "L" and the program name. Once the program is loaded, type S to start the program. DEC diagnostics have a self-starting capability at location 200.

If a typing error is made, press the backspace key for each character you wish to remove from the data previously entered. This must be done before the carriage return is pressed.

To abort the function in progress and return to monitor command mode, simultaneously press the CTRL and C keys (CTRL-C). There may be a time delay before the command mode is reentered.

To freeze output going to the terminal or line printer, type a CTRL-S (XOFF). To allow printing to resume, type a CTRL-Q (XON).

5.8.4 Chain Mode Operation

Chain mode operation consists of the sequential execution of programs without operator intervention. Only programs that have been modified to run in chain mode can be chained. Chainable programs are identified in the directory by a BIC extension.

To run chain mode, the XXDP+ monitor uses an ASCII file (known as a chain file) listing the programs to be run and the number of passes each program should run. This file must be on the system device.

A chain file is generated by using the XTECO text editor; the file must have a .CCC extension. The chain file may contain any of the commands supported by the XXDP+ monitor. The commands in the ASCII file are executed in the order in which they are encountered. Comments may be included in the file.

5.8.5 Running a Chain

To execute a chain file the user types;

C FILNAM(CR)
or
C FILNAM/QV (CR)

In the first case the pass count specified in the chain file is used by the XXDP+ monitor to determine the number of passes to execute each program. In the second case the pass count is not used and each program is executed only once; the /QV switch provides a single execution mode of operation or quick verify.

When programs are run in chain mode, the hardware or software switch register should be set to 000000.

The XXDP+ monitor prints each command taken from the chain file and then executes the command.

When the last command other than another "C" command has been executed, the XXDP+ monitor terminates chain mode and types a prompt (.).

5.8.6 2515 Verification Test

The verification routines have been chained for standard 2515 configurations. If your system is non-standard, it may be necessary to modify the existing chain or build a new one by using the XTECO text editor described in paragraph 5.10.

To begin, simply type C 2515 (CR) after the dot(.) prompt. It will be necessary to occasionally toggle the CRT into "Page Scroll" mode as some tests will flip the CRT into "Page" mode.

DIR .TXT

2515 Computer-Aided Test System Diagnostic Directory
(Version 3)

FILE NAME	DESCRIPTION
HSAAD0	XXDP+ MONITOR
HUDIB0	XXDP+ DIRECTORY
HDDKB0	DK DRIVER
HDDLC0	DL DRIVER
HDDYB0	DY DRIVER
HMDKB0	DK MONITOR
HMDLD0	DL MONITOR
HMDYC0	DY MONITOR
HELP	VAD TEXT/HELP MESSAGE
HELP1	DEC HELP (TYPE C HELP1 (RETURN))
UPD2	UPDATE PROGRAM
XTECO	TEXT EDITOR
JKDBD0	LSI 11/23 CPU TEST
JKDAD1	LSI 11/23 MEMORY MANAGEMENT
JKDCB0	LSI 11/23 FLOATING POINT TEST 1
JKDDB0	LSI 11/23 FLOATING POINT TEST 2
VMSAA0	0-2 MEG MEMORY TEST (AUTO SIZING)

ZRXEA2	FLOPPY DISK FORMAT PROGRAM FOR DSD440 8" DRIVES
DLVTST	DLV11-J TEST (ADDRESS MODIFIED FOR 2515)
DLV15	DLV11-J TEST (MODIFIED, MAINT. PLUG REQ.)
VDLAB0	DLV11-J TEST (DEC STANDARD)
VKAFD1	DRV11 TEST (DEC STANDARD)
2515	USER CHAINED SYSTEM DIAGNOSTIC TESTS
2515P	PRODUCTION CHAINED SYSTEM DIAGNOSTIC TESTS
DIRECT	PRINT THIS DIRECTORY
DIR	THIS TEXT
EQBTST	QBUS EXPANSION OPTION TEST

Note: To test the Winchester and 5-1/4" floppy disk drives, use WINFLP.SAV on the RT-11 diagnostic disk.

5.8.7 ZRXAE0 RX02 FORMATTER PROGRAM (2515-0690 Diskette)

5.8.7.1 Abstract. This program is intended to format a floppy disk to either single or double density on a drive capable of setting density on a floppy disk.

5.8.7.2 Hardware Required. (1) Any PDP-11 Processor (2) RX02 Compatible Subsystem.

5.8.7.3 Storage. This program requires at least 4K words of core. If loading via XXDP+ more storage will be needed for the XXDP+ monitor.

5.8.7.4 Starting Procedure. Start at location 200 (octal) to initialize registers.

5.8.7.5 Operation. The program operation in "user mode" is simple. After initial start the program asks the operator if help is wanted and will type a short description of the program if answered yes. The program also asks if a full diskette scan is wanted to verify CRC. If address modification is selected then the program asks the operator to enter the bus address and vector address.

5.8.8 VDLAB0 DLV11-J Test

This diagnostic is a logic test to verify the operation of the DLV11-J serial interface. Testing is done in two phases:

- All selected channels per DLV11-J module are tested individually
- The DLV11-J module is tested as a whole for channel interaction problems. This diagnostic is designed to test and detect errors to the logic level (not to the chip level).

The operator must install data wrap around connectors to do data testing. To bypass data tests, the operator must modify the user switch register at address 1220, see program options in paragraph 5.8.8.5.

The default addresses and vectors are as follows:

177560 - Console Interface Device Address
176500 - First serial channel address of up to 8 consecutive serial line devices.

60 - Vector for console device interface.
300 - Vector for first of 16 devices.

This program is designed to run on any Q-BUS PDP-11 with 4K of memory and a DLV11-J (Q-Bus) module.

5.8.8.1 System Requirements.

Hardware

- Any Q-Bus PDP-11 family processor
- 4K memory - minimum
- A special data wrap around connector or equivalent (required if data wrap around tests desired)

- If channel 3 is configured as the console
  Tests 6-12, 14-17, 21, 22 are bypassed.

- If data wrap around tests are bypassed
  Tests 7-12, 14-17, 21, 22 are bypassed.

Software. This diagnostic is designed to run in either of the following ways:

- Stand alone
- With XXDP monitor+ (chainable if renamed to .BIC extension)

5.8.8.2 Assumptions. This diagnostic assumes that the operator has initialized location 176 (switch register), 1220 (options), 1250 (address) and 1244 (vector) to the proper values.

5.8.8.3 Operating Instructions. Use standard procedure for PDP-11 absolute binary formatted media. There are 2 starting locations for this diagnostics:

- Location 200 for all normal testing starts and restarts.

- Location 400 for interrupt vector debug, off-line only.

5.8.8.4 Switch Register.

Step 1. Type control G (^ G); This allows the TTY to enter data into location 176 at selected points within the program.

Step 2. The test then types: SWR=XXXXXX is the octal contents of the software switch register.)

Step 3. After the NEW= has been typed the operator does one of the following:

- Type a number to be loaded into location 176 followed by a (CR). (Only numbers between 0-7 will be accepted.) Leading zeros need not be typed, and if more than 6 digits are typed the last 6 are used. If a (CR) is the first key depressed the software switch register contents is not changed.

- If a control U (^U) is depressed then the program will send you back to Step 3.

- If the input character is not one of the characters mentioned above then a question mark (?) is typed followed by a carriage return and a line feed sequence then proceed from Step 3 (erasing all previous input).

  The diagnostic continues typing (CR).

Note: Because of frequent bus resets in the program, it may be necessary to do (^G) several times. Alternately, a 'BREAK' and manually loading location 176, followed by a 'P' to proceed will also work.

Software Switch Register Options (SWREG)

BIT	15	SET = 100000	=	Halt on Error
	14	SET = 40000	=	Loop on Test (To be used only while testing in progress)
	13	SET = 20000	=	Inhibit Error Typeouts
	12	SET = 10000	=	Enable Performance Reports
	11	SET = 4000	=	Inhibit Iterations
	10	SET = 2000	=	Bell on Error
	9	SET = 1000	=	Loop on Error
	8	SET = 400	=	Loop on Test in SWR(7:0)
	7:0	=	=	Number of Test to Loop On (Used with bit 8) (All tests previous to the selected test are executed first with one iteration only)

5.8.8.5 Program Options. This program requires the address of the first RCSR (stored at 1250) and its interrupt vector (stored at 1244). It is able to address any DLV11-J starting at the specified base address thru 2 consecutive modules.

Examples:

1250	176500	(Default)	GR 175640
1244	300	(Default)	GR 340

Location 1220 is used as a bit map to indicate which unit numbers are present and are to be tested.

Bit Position	Definition	Default Value
0	# of Data Bits Transmitted 0 = 7 Bits, 1 = 8 Bits	1 = 8 Bits
1	Parity Enabled	0 = No
2	Even Odd Parity	0 = Odd
3	Break Detection Enabled	1 = Yes
4	Run Data Wrap Around Tests	0 = No
(11:9)	Console Device	1 = Yes = Console on Module 1, Channel 3

(11:9) = 0 = Console Not on DLV11-J
(11:9) = 1 = Console On Module 1
(11:9) = 2 = Console On Module 2

5.8.8.6 Execution Times. Execution times for an LSI-11 processor with one DLV11-J module at shipment configuration:

	Ch. 0,1,2	At 9600 baud.
	Ch. 3 (Console)	At 300 baud.
Are:	First Pass-	30 Sec.
	Additional Passes	90 Sec.

The test time is baud rate dependent; higher baud rates result in shorter pass times.

5.8.8.7 Error Information. Since this diagnostic was designed to fit in 4K of memory the error typeout is very brief. The format of the error typeout is as follows:

Test #_____, Error #_____, PC=_____, Address=_____, Vector=_____

Where all values typed are octal. The address and vector refer to the failing channel. For further information the listing must be consulted. Bits 15, 13, 10 and 9 of the switch register control the sequency of events after an error is caught.

Bit 15 Set: Causes the program to halt in the error routine. If the program is continued, it proceeds from where it halted.

Bit 13 Set: Disables the printing of the error message.

Bit 10 Set: Causes the bell to ring on error.

Bit 9 Set: Causes the diagnostic to loop from beginning of test to error.

The only halt in this diagnostic is in the error routine, and is executed only if Bit 15 of the switch register is set when an error occurs.

5.8.8.8 Performance and Progress Reports. Performance Reports. (Bit 12 set in the switch register). As each channel (4 channels/DLV11-J) completes one pass of the diagnostic the following is typed:

CSR: ——— : The base address of the line under test
Vector: —— : The associated vector
Errors:——— : The total number of errors on this device on this pass

After all modules and channels to be tested have been exercised, an end pass statement is typed:

END PASS#———.

5.8.8.9 Summary of Tests and Special Subroutines.

Test 1. Addressability. This test verifies that all 4 registers of the channel under test respond to their addresses.

NOTE

The following three tests (2, 3, and 4) all 'READ WRITE' bits

Test 2. BREAK - TCSR 0 SET, CLEAR, RESET

Test 3. XMITIE - TCSR 6 SET CLEAR, RESET

Test 4. REVRIE - RCSR 6 SET, CLEAR, RESET

Test 5. XMITRDY - TCSR 7 - IS SET BY INIT

Test 6. XMIT RDY - TCSR 7 - Clears when TBUF is loaded with a character and that it sets within a reasonable amount of time.

Test 7. Outputting a character from TBUF (with wrap around connected) results in RCVRDONE setting within a reasonable amount of time and that reset clears the bit.

Test 10. RCVRDONE is cleared by setting reader enable

Test 11. RCVRDONE is cleared by reading RBUF

Test 12. Overrun and Error Bit - RBUF 14

Test 13. Transmitter interrupt logic test

Test 14. Receiver interrupt logic test this test covers all of the receiver side of the interrupt logic in character mode.

Test 15. Test data wrap around; FLAG MODE

Test 16. Test data wrap around: INTERRUPT MODE

Test 20. Not a test - send back to loop

Test 21. Test that channels interrupt at assigned priority

Test 22. Test data transfers with all active lines interrupting

5.8.9 DLVTST - DLV11-J Test (No test plug required)

This diagnostic tests the DLV11-J serial interface as configured for the GenRad 2515.

5.8.9.1 Operating Instructions. See paragraph 5.8.8 (VDLAB0 DLV11-J). The test will run without the maintenance test plug connected.

5.8.9.2 Program Modifications. The following changes were made to VDLAB0.

Address	From	To	Comment
1220	1031	11	No console or wrap plug
1244	300	340	Vector charge
1250	176500	175640	Address change
4730	500	5	Timing loop bug

5.8.10 DLV11 - DLV11-J Test (with test plug)

This diagnostic tests the DLV11-J serial interface as configured for the GenRad 2511.

5.8.10.1 Operating Instruction. See paragraph 5.8.8 (VDLAB0 DLV11-J).

5.8.10.2 Program Modifications. The following changes were made to VDLAB0:

Address	From	To	Comment
1220	1031	11	No console or wrap plug
1244	300	340	Vector change
1250	176500	175640	Address change
4730	500	5	Timing loop bug

5.8.10.3 Special Requirements. Test plugs must be installed on all four ports. The test loops data via the transmitter and receiver logic.

5.8.11 VKAFD1 - DRV11 Test

This is a logic test of the DRV11, to allow testing of the data lines and interrupts. A special maintenance cable (RC08R) is used by default. This test operates on one DRV11. Special operational procedures are required to operate on other than the primary DRV11.

5.8.11.1 Hardware Requirements.

- LSI-11
- DRV11
- Test Cable (BC08R)

5.8.11.2 Starting Procedure. 200 - Normal entry to test one device. If any program options are required, set the appropriate bit in the software switch register at location 422. Start program at 200. Program prints END OF PASS following each pass.

5.8.11.3 Control Switch Setting This program contains a software switch register for optional selection. For it to operate, the operator must select the appropriate option by setting or resetting the respective bit in the word. To do this, the LSI-11 must be in ODT mode. Starting address or addresses

200 = Start of Test — For normal testing

The program cycles continuously unless halted by the operator or some error condition. To halt the program, depress the halt switch. ODT displays the PC at which it was halted.

Reset at this time.

Continue the program via a P or a G command.

Software switch settings (Address 422)

Bit15	-	Continue on error	(100000)
Bit14	-	Loop on current error	(040000)
Bit13	-	Not used	(020000)
Bit12	-	Not used	(010000)
Bit11	-	Not used	(010000)
Bit10	-	Loop on current test	(002000)
Bit9	-	Run test module (inhouse only)	(001000)
Bit8	-	Inhibit wrap cable	(000400)
Bit7	-	Not used	(000200)
Bit6	-	Not used	(000100)
Bit5	-	Not used	(000040)
Bit4	-	Not used	(000020)
Bit3	-	Not used	(000010)
Bit2	-	Not used	(000004)
Bit1	-	Not used	(000002)
Bit0	-	Not used	(000001)

Selection of Test Options.

This test runs with the wrap cable by default. To inhibit testing with the wrap cable, the operator must set Bit8 in the switch register (location 422).

Wrap Cable.

The wrap cable is required to test transfer of data into and out of the input buffer, and the device interrupts.

This diagnostic is approximately 95% effective when run with the wrap cable, and approximately 60-70% effective when run without it.

5.8.11.4 Execution time. Typical run times (one pass):

- Quick verify 1 Sec.
- With wrap cable 10 Sec.

5.8.11.5 Error reporting. All error reports will be done via a halt within the program. This will cause ODT to display the PC+2 of the error halt. In order to continue, the operator must issue a "P" to continue the program, or may set the error loop switch prior to continuing.

5.8.12 VMSAA0 0-2 MEG Memory Exerciser

This program has the ability to test memory from address 000000 to address 17757777. It does so using:

- Unique addressing techniques,
- Worse case noise patterns, and
- Instruction execution throughout memory.

The intent of this program is to test as comprehensively as possible all MOS memories used on the LSI-11 Bus without concentrating on any one system. Although the tests relate to general designs they may be complete for certain systems. This test is also not intended to be a total 100% test of the memory. Other tests that do I/O may find memory problems that this test is unable to.

5.8.12.1 System Requirements. LSI-11/2, LSI-11/23 family processors. Minimum of 16KW of memory. Any parity memory control module (optional). KTF11 memory management.

5.8.12.2 Diagnostic Prerequisites. Before running this program, a CPU diagnostic should be run to verify the functionality of the processor and PDP-11 instruction set.

For LSE-11/23:CJKDB?? DIAG
For LSE-11/2 :CVKAA?? DIAG

If memory management is to be used, then the KTF11 diagnostic CJKDA?? should also be run before this program.

5.8.12.3 Operating Instructions.

- Starting address 200:

  Normal program execution.

- Starting address 204:

  Restart program using previously selected parameters.

5.8.12.4 Program Options. The software switch register (location 176) is used for all operational switch settings. The user can type CTRL G (^G) to allow SWR changes during program execution.

SW15 = 1 or up....	Halt on Error
SW14 = 1 or up....	Loop on Test
SW13 = 1 or up....	Inhibit Error Typeout
SW12 = 1 or up....	Inhibit Memory Management (initial start only)
SW11 = 1 or up....	Inhibit Subtest Iteration (not used)
SW10 = 1 or up....	Ring Bell on Error
SW9 = 1 or up....	Loop on Error
SW8 = 1 or up....	Loop on Test in SWR(4:0)
SW7 = 1 or up....	Inhibit Program Relocation
SW6 = 1 or up....	Inhibit Parity Error Detection
SW5 = 1 or up....	Inhibit Exercising Vector Area (Locations 0-1000).

5.8.12.5 Execution Times. Execution time is dependent on type of memory, and amount of memory. Worse case run times with MOS memory are:

A. For non-parity memory

Full pass: < 5 minutes for 64KW.
30 minutes for 1280 KW

5.8.12.6 Error Reporting. There are a total of 31(8) types of error reports generated by the program. Some of the key column heading mnemonics are described below for clarity:

PC = Program counter of error detection code. (V/PC=P/PC)

V/PC = Virtual program counter. This is where the error detection code can be found in the program listing.

P/PC = Physical program counter. This is where the error detection code is actually located in memory.

THP/PC = Physical program counter of the code which caused a trap.

MA = Memory address.

REG = Parity register address.

PS = Processor status word.

IUT = Instruction under test.

S/B = What contents should be.

WAS = What contents was.

5.8.12.7 Error Halts. With the HALT ON ERROR switch (SW15) not set there are several programmed "HALTS' in the program:

- In the error trap service routine for unexpected traps to vector 4. This one occurs as a second trap to 4 and occurs before the error report for the first has had a chance to be printed out.
- In the relocation routine if the program is being relocated back to the first 8K of memory and the program code was not able to be transferred properly.
- In the case of error reporting and there is no terminal to allow the information transfer.
- In the power fail routine, if the power up sequence was started before the power down sequence had a chance to to complete itself.
- In the memory mapping routine or any of the address control routines, failures to find a meaningful map.

5.8.13 JKDBD0 11/23 CPU Diagnostic

This program is a combined version of the three basic 11/34 diagnostic programs with modifications and enhancements made to account for the differences between the two processors. The program contains three parts: CPU, TRAP and EIS tests. In the first and second parts, the program will halt on error. In part three (EIS test) when an error is detected, the error PC and error number is typed, then the program continues execution.

Part One: CPU test, this is the first part of the main program. This test checks out the basic PDP-11 instructions in all addressing modes with various types of data patterns.

Part Two: TRAP test, this is the second part of the main program. This is a test of all operations and instructions that cause traps. Also tested are trap overflow conditions, oddities of register 6, interrupts, the reset and wait instructions. This program checks that on all trap operations register 6 is decremented the correct amount, that the correct PCS is saved on the stack, that the old condition codes and priority are placed on the stack and both the "TRAP and "EMT" trap instructions are tested to see that all combinations will trap. Checked also is that all reserved instructions will trap. The trace bit is checked to see if it causes a trap. The RT1 and RTT instructions are checked. Stack overflow is also checked for all the trap instructions.

Special checks are made to see if BUS error traps occur on non-existent memory. All instructions that are reserved should trap to location 10, and the PC that points to the trapping instruction should be placed on the stack.

Part Three: This program tests the extended instruction set (ASH, ASHC, MUL, and DIV) using registers 0-5 at least once with each instruction. This program tests all the EIS instructions of the 11/23 for ASH and ASHC instructions every even pass is executed with destination mode 0 for all registers and every odd pass with destination mode of 67. The diagnostic does not make a pass with T bit set.

5.8.13.1 Starting Procedures. The program is started by loading address 200. The restart address is 1024. Program identification is typed after the first pass of the whole program.

5.8.13.2 Error Handling. In parts one and two, all errors cause a halt. The program checks to see that the PC doesn't jump erratically within the tests. If an error is detected, the program halts. It could be because of two reasons.

- Wrong test number (sequence error)
- Error in the present test.

In part three, any error (including sequency check error) causes the error message to be typed. The program continues execution after type out. The error reporting format is as follows:

ERROR! PC AND ERROR # ARE:

PC #

ERROR #

5.8.13.3 Switch Settings. Since no hardware switch register is available, the program automatically uses the contents of Loc. 176 as the software switch register. The initial content of Location 176 is 000000, the user may preset this location before starting the program.

BIT #	OCTAL VALUE	FUNCTION
15	100000.....	HALT ON ERROR
13	020000.....	INHIBIT ERROR PRINTOUT
2	000004.....	PROGRAM RESERVED — PROGRAM WILL SET IF CIS OPTION IS AVAILABLE
1	000002.....	30K SYSTEM DO NOT CHECK FOR TRAPS BETWEEN 28K-30K
0	000001.....	SKIP TRAPS TEST

5.8.13.4 Execution Times. The run time for a single run (the first pass) is one second. After the first pass, the program iterates every 15 times before the end of

pass message is typed again. The run time for each additional end of pass message typed is approximately 15 seconds.

5.8.14 JKDAD1 Memory Management Diasgnostic

This program was designed using a bottom up approach starting with the smallest segment of memory management logic possible and building to cover all of the logic. The program begins by testing some of the internal CPU data and address paths and address detection logic, then works outward through the memory management registers. After the registers are found to be usable, relocation (construction of physical addresses from a virtual address and the associated PAR/PDR information) is tested followed by testing of the abort and status segments of logic. Finally, checks of special abort sequences and testing of the MFPI/MTPI instructions are done.

A 11/23 processor with a minimum of 16K of memory and a console terminal are required to run the program.

5.8.14.1 Preliminary Programs. Before this memory management diagnostic is run, one of the main memory diagnostics should be run to scan at least the first 16K to see that a program can be executed. The following CPU diagnostic should also be run:

JKDBD0 11/23 CPU TESTS

5.8.14.2 Starting Procedures. The program is started by loading address 200. Since there is no hardware switch register, the program will use the software switch register at location 176 (location 174 will be used as the software display register). The program asks for the initial switch register value by typing "SWR = XXXXX NEW = " after typing the name of the program (XXXXXX = THE OCTAL CONTENTS OF LOCATION 176).

5.8.14.3 Control Switch Settings.

Switch	Octal/Value	Use	Description
SW15	100000	HALT ON ERROR	This switch when set halts the processor when an error is detected after the error message has been typed. Pressing continue resumes testing.
SW14	040000	LOOP ON TEST	This switch when set causes the program to loop on the current subtest.
SW13	020000	INHIBIT ERROR TYPEOUTS	This switch when set inhibits the typing of error messages.

Switch	Octal/Value	Use	Description
SW12	010000	INHIBIT TRACE TRAP	This switch when set inhibits T-bit trapping which normally takes place during every other pass starting with the third pass.
SW11	004000	INHIBIT SUBTEST ITERATIONS	This switch when set inhibits iterations of each subtest after the first pass. If this switch is not set, each subtest is run 200 times.
SW10	002000	BELL ON ERROR	This switch when set rings the console terminal bell when an error has been detected.
SW9	001000	LOOP ON ERROR	This switch when set causes the program to loop on the first failure which is encountered even if the failure is intermittent.
SW8	000400	LOOP ON TEST IN SWR <7:0 >	This switch when set causes the program to loop on the test whose test number is set in bits 7-0 of the switch register.

5.8.14.4 Loading The Switch Register. To load the software switch register while the program is running, a control G (^G) should be typed on the console terminal. The original value of the software switch register will be requested.

In response to A ^G or at the beginning of the program, the program types:

SWR = XXXXXX NEW =

Where "XXXXXX" is the current octal contents of location 176. The operator may then type any one of the following:

XXXXXX<CR>	One to six octal digits followed by a carriage return which are loaded as the new value for the switch register.
<CR>	Just a <CR>, leaves the switch register as it is.
XXX ^U	A control-U (^U) causes all of the digits typed so far to be ignored.
^C	Causes the program to type the present test and pass numbers, request a new value for the switch

register, and jump to the end-on-pass routine so the program goes directly to the next pass with a new switch register value.

<ILL. CHAR> Any character typed which is not any of the above or an octal digit causes the program to type ? <CRLF> and react as though a ^C has been typed.

NOTE

Recognition of a ^G may be hampered by execution of a couple of reset instructions within the program.

5.8.14.5 Execution Times. The run time for a single pass with no iterations or trace trapping is approximately 5 seconds.

5.8.14.6 Error Information. If an error is detected, the program traps to the error handling routine. The value of bits 15, 13, 10, and 9 in the switch register are considered in reporting an error. The error information is typed unless SW13 - 1.

If SW15 - 1, the processor halts after the error is reported. If the contents of the software switch register are to be changed, a ^G should be typed before pressing "CONTINUE" to resume testing.

Below is an example of an error which might occur during execution of the program:

```
MEM. MGMT. REG. BITS NOT ST CORRECTLY
REGISER WROTE READ READ-(BINARY)
ADDRESS (OCTAL) (OCTAL) 543210987654321  TESTNO  ERRORPC
177572    040000    060000   0110000000000000  000012    022060
```

5.8.14.7 End-of-Pass Message. At the end of each pass of the program the pass number and total number of errors since the last end-of-pass are reported in the end-of-pass message.

Example:

END OF PASS #2 TOTAL ERRORS SINCE LAST REPORT 0

5.8.15 JKDCB0 FLOATING POINT DIAGNOSTIC #1
JKDDB0 FLOATING POINT DIAGNOSTIC #2

The two programs JKDCB, JKDDB are designed to detect and report logic faults in the F-11 MMU and floating point chip set. The design is an attempt to reach all microcode locations.

Note that error reports in these programs are based upon the knowledge that all previous tests (CPU, MMU) have been run and in most cases only a single point

fault exists. If the programs or tests are not run in order, error messages may not be accurate.

5.8.15.1 Starting Procedure. Perform the following steps:

Step 1. Load address 200.

Step 2. Set console switches (if console is present).

Step 3. Press start.

On first pass the program will identify itself. If there is no physical console the program requests the operator for initial value for the software switch register. The program loops and an end-of-pass message will be typed at the end of every pass.

5.8.15.2 Operational Switch Settings. The switch settings are:

	Octal	
SW<15> =1...	100000	HALT ON ERROR
SW<14> =1...	40000	LOOP ON CURRENT TEST
SW<13> =1...	20000	INHIBIT ERROR TYPE OUTS
SW<12> =1...	10000	INHIBIT T-BIT TRAPPING
SW<11> =1...	4000	INHIBIT ITERATIONS
SW<10> =1...	2000	RING TTY BELL ON ERROR
SW< 9> =1....	1000	LOOP ON ERROR
SW< 8> =1....	400	LOOP ON TEST SPECIFIED IN SW<6>

5.8.15.3 Errors. When an error is encountered, an error message accompanied by the error PC is typed. There are four standard error messages used, describing the probable cause of failure, such as: probably bad MMU chip; bad DP1 chip; bad hybrid FP chip; or floating point error.

ERROR RECOVERY

SW<15:9>=0...	Most errors cause execution to go to the start of the next test after the message is typed. A few tests are in sections. In these tests an error causes execution to go to the next section after the message is typed.
SW<15>=1...	The program halts after typing the error message. Pressing the console continue causes the program to continue as if SW<15>=0.

5.8.15.4 Execution Times. Less than 2 seconds for each program on any pass.

5.9 XXDP+ UPDATE PROGRAM

XXDP+ has one utility program to be used for file maintenance. This program is called UPD2. The UPD2 program runs on an 8K system but can not load a 4K file there.

The UPD2 program is not stand alone and shares memory with the monitor. UPD2 uses certain functions made available by the monitor. The exit command is used to return to the monitor.

UPD2 contains read/write drivers for devices. These drivers are resident on the system device as SYS files. They are transferred to memory as they are needed.

Utility program prints an asterisk (*) as a prompt for command mode. The valid commands for the UPD2 utility are next described. Some of the commands described below allow the use of wildcards in the filename and extension. Wildcards are the * and % characters. These characters may be placed in filenames and extensions to indicate that any characters are valid in the characters positions indicated by the wildcards. The * wildcard indicates all character positions from the position of the * to the end of the filename or extension. The % wildcard indicates only the character position occupied by the % character. The character ? is accepted for the character %. The following examples illustrate the use of wildcards:

Z%.BIN	All files beginning with Z and having extensions of BIN.
Z%%%%%.BIN	All files beginning with Z and having extensions of BIN.
.	All files regardless of filenames and extensions.

5.9.1 DIR Command

The DIRectory command causes the directory of the specified device to be printed. Wildcards are allowed. The format of the command is: DIR DEV: (CR). If the DEV: is omitted, the directory will be obtained from the system device.

DIR DEV:	Gives a directory of all files on DEV.
DIR *.BIN	Gives a directory of all files with a BIN extension. The system device is used.
DIR DEV:$.BI%	Gives a directory of all files with extensions beginning with BI.
DIR:ZYC##.BI%	Gives a directory of all files with the first three characters of the filename being ZTC and having extensions beginning with BI. The system device is used.

Example:

Entry#	FILNAM.EXT	DATE	LENGTH	START
000001	A .BIN	2-AUG-72	14	000105
000002	B .BIN	2-AUG-72	12C	000106

FILNAM.EXT is the name of the file.
Date is the file creation date.

Length is the number of blocks (in decimal) the file occupies. C after the file length indicates the file is contiguous. Non-contiguous files are linked lists.

Start if the block number (in octal) of first block of the file.

The DIRectory command accepts two switches. The /F switch gives a FAST directory containing only the entry and filename.

5.9.2 CLR Command

The CLR command sets to zero all locations in the buffer area used for loading/modifying/dumping files to/from a medium. The format of the command is:

CLR (CR)

5.9.3 LOAD Command

The LOAD command loads files stored in ABS format into a buffer area within the utility program. Checksum tests are performed on the files. The format of the command is:

LOAD DEV:FILNAM.EXT(CR)

If the DEV: is omitted, the file is loaded from the system device.

If the device has no directory, the file name and extension should be omitted.

Example:

LOAD PR:(CR) loads from the paper-tape reader. Following the load, certain information is printed.

XFRADR: Indicates the starting address of the program loaded. If it is odd, the program is not self-starting.

CORE: Left number indicates the lowest location loaded into during the load. This value is known as the locore value. The right number indicates the highest location loaded into during the load. This value is known as the hicore value. The left and right numbers in effect indicate the core limits of the program. These core limits are relative to the buffer area within the utility.

Wildcards are allowed and cause more than one file to be loaded. The files load into the same buffer in the utility program. The use of wildcards with this command is one way of performing checksum tests on a group of files.

5.9.4 DUMP Command

Contents of the buffer area within the utility can be written to a XXDP+ medium in ABS format by the DUMP command. The format of the command is:

DUMP DEV:FILNAM.EXT.

If the DEV: is omitted, the file is stored on the system device. Processing starts from locore limit of the program and proceeds to but does not include the hicore limit of the program.

5.9.5 XFR Command

Once a program has been loaded into the buffer area within the utility via the LOAD command, it can be made self-starting or not self-starting at the user's discretion. As described under LOAD command, the LOAD routine types: XFRADR: XXXXXX indicating whether a program is or is not self-starting.

Example:

XFR(CR)	Request current transfer address
000001 000050	000001 is the current XFR address. 000050 is the new XFR address entered by the user.

5.9.6 SAVE Command

The SAVE command writes the contents of the program buffer area of the utility program as a core image file. The transfer starts at the beginning of the buffer and ends at the word before the hicore value. The only current use of the SAVE command is to place a core image of the XXDP+ monitor on cassette and magtape. The format of the command is:

SAVE DEV:FILNAM.SAV(CR)

If the DEV: is omitted, the file is stored on the system device.

5.9.7 MOD Command

Once a program is loaded with the load command it can be patched by the MOD command. The format of this command is:

MOD ADDRESS(CR)

The utility prints the address and its current contents. The user may type in an octal number and a terminator, or just a terminator. If a number is typed, it is used as the new content of address.

The terminator can be either a carriage return or a line feed. A carriage return takes the program back to command mode, whereas a line feed causes the next word (ADR+2) to be opened for modification.

Example:

```
*MOD 50
000050  000005     3 (LF)
000052  012737     4 (LF)
000054  000300     5 (CR)
```

```
*MOD 50
000050  000003        (LF)
000052  000004        (CR)
```

NOTE

Line feed is ↓ on key pad of 2515 keyboard.

Any location within the locore/hicore limits can be modified with this command.

5.9.8 CORE Command

The CORE command causes the locore/hicore limits of the program buffer in the utility to be typed: The format of the command is:

CORE(CR)

Example:

```
#CORE(CR)
00000,014776        Left number is the lower core limit,
                    Right number is the upper core limit
```

5.9.9 LOCORE Command

The LOCORE command is used to change the lower limit of the program relative to the program buffer in the utility. The format of the command is:

LOCORE ADR(CR).
ADR is the new low core value.

5.9.10 HICORE Command

The HICORE command is used to change the upper limit of the program relative to the program buffer. The format of the command is:

HICORE ADR(CR).
ADR is the new high core value.

5.9.11 DELETE Command

The DELETE command causes a file to be deleted from the specified device. The format of the command is:

DEL DEV:FILNAM.EXT(CR).

The DEV: must be specified. Wildcards are allowed.

5.9.12 ZERO Command

The ZERO command causes all file information to be removed from the directory. The format of the command is:

ZERO DEV:(CR).

The DEV: must be specified. If the specified device is the system device, a dialogue with the operator occurs. This dialogue warns the operator that an additional read/write driver may be needed. A request to continue is displayed.

```
*ZERO DY0:

You may need additional driver
CONTINUE? (Y or N) ___
*
```

Note that before entering Y to continue, you must place a scratch pack on the drive. Files from DY0 are transferred to the scratch pack.

5.9.13 BOOT Command

The BOOT command causes block 0 of DEV to be loaded into memory. Starting at location 000000. Block 0 is assumed to have a boot loader. The program then jumps to location 000000 to start the boot loader. The format of the command is:

BOOT DEV:(CR)

If the DEV: is not specified, the system device is booted.

5.9.14 SAVM Command

The SAVM command causes the first 4K of the buffer area in the utility program to be written as a core image. This command is used to write the XXDP+ monitor on the device as a core image that is bootable. The format of the command is:

SAVM DEV:(CR)

This command is only valid for random access devices. The DEV: must be specified.

Example:

```
#LOAD DY1:HMDY??.BIN    Load DYDP+ Monitor
#SAVM DY0:              Save it as core image on DY0:
```

Note that SAVM does not cause a directory entry. The SAVM command also writes the first 256(10) words of the buffer into block 0 of the device. This is the boot loader.

5.9.15 REN Command

The REN command changes the name of a file on a medium. The format of the command is:

REN DEV:NEWNAM.EXT=DEV:OLDNAM.EXT

If the DEV: is not given, the system device will be used. Wildcards are allowed.

5.9.16 DRIVER Command

The DRIVER command is used to load device drivers into memory. This command is not ordinarily required because drivers are loaded as part of the execution of the other update commands. For example, the command DIR DEV: causes the driver for DEV to be loaded and the directory of that device to be printed. However, the ZERO command creates a situation in which the user may need to have another driver loaded into memory. The format of the DRIVER command is:

DRIVER DEV1: DEV2:

5.9.17 PIP Command

The PIP command is used to transfer files from one medium to another medium. File data is not checked for format on checksums. The output file is given the date of the input file. The format of the command is:

PIP DEV1:FILNAM.EXT=DEV2:FILNAM.EXT(CR)

The file names for the two devices can be different. If DEV is omitted, the system device will be used. Wildcards are allowed. Examples of this command follow:

PIP PP:=PR:	(Copies paper tape).
PIP DK0:ABC.BIN=PRI	Paper tape to disk.
PIP PP:=ABC.BIN	Disk to paper tape punch.
PIP DX0:*.*=*.BIN	Transfers all BIN files to DX0.

The user should make sure that the output file name does not already exist in the output device directory.

Program files that have been PIP'ed to a XXDP+ device should be loaded immediately via the LOAD command to ensure that no errors have occurred during the PIP command. ASCII files that have been PIP'ed can be printed via the TYPE command.

5.9.18 READ Command

The READ command can be used to test for device errors by transferring files from a medium. The information obtained from the files is not retained. The format of the command is:

DEV:FILNAM.EXT(CR)

If DEV: is not given, the system device is used. Wildcards may be used.

5.9.19 TYPE Command

The TYPE command outputs files to the console terminal. It is used to print text files. The format of this command is:

TYPE DEV:FILNAM.EXT

If the DEV: is omitted, the system device is used. Wildcards may be used.

5.9.20 DO Command

The DO command causes the execution of a command file. The file must be on the system device. The file can contain any legal UPD2 commands. The file is executed line by line and must not be terminated by an "Exit". Executable files are created via the XTECO text editor program. The format of the DO command is:

DO DEV:FILNAM.EXT(CR)

Note that two special characters, the semicolon (;) and the dollar sign ($), have special significance in the text file. The semicolon is used to start a comment which is to be printed during command file execution. The dollar sign is used to start a comment which is to be printed and followed by a halt during command file execution (such as $PRESS CONT WHEN READY).

5.9.21 ASG Command

The ASG (ASSIGN) command allows the use of logical device names in place of the actual or physical device name. Allowed logical device names are 0, 1, 2, 3, 4 and 5. The format of this command is:

ASG PHYSICALDEV:=LOGICALDEV:(CR).

5.9.22 PATCH Command

The PATCH command enables the user to patch a program on any block oriented (random access) XXDP+ supported device. No output DEV: file specification is required or permitted. The input device is assumed to be the desired output device. The file to be patched must be in ABS format. The patch routine does not check in advance for correct file format.

The operator types "PATCH DEV:FILENAME", and UPD2 responds with the prompt ADDR? to which the operator should enter the address he wishes to patch. Then UPD2 will respond by typing the present contents of that location in the bin file and waiting for the operator to enter the desired new contents. When the operator does so, UPD2 responds by printing the ADDR? prompt again. The patches are all done in real time, rather than being accumulated in memory pending a verification response from the operator.

The operator may terminate the patch session by responding to the ADDR? prompt with a simple (CR).

5.9.23 EXIT Command

The EXIT command returns program control to the XXDP+ monitor. The format of this command is:

EXIT (CR)

5.9.24 COPY Command

The COPY command allows a file or an image copy to be performed between like devices. The operator enters the command COPY NEWDEV:=OLDDEV: and UPD2 responds by asking FILE COPY OR IMAGE COPY?

If the operator responds Image, then a check is made to ensure that this device is neither a bad block disk nor a tape. If this check is successful, then each physical block of the input device is written to the corresponding physical block of the output device. When the operation is completed, the output device is read to perform the verification by checksum against the data that was read from the input device on the transfer pass.

If the operator responds File, then any device type is legal and the utility simply does the equivalent of a PIP out:*.*=IN:*.*.

5.9.25 UPD2 Errors

? INVALID DEVICE	Invalid device specified for command given.
? INVALID ADDRESS	Invalid address. Must be even and within existing locore and hicore limits.
? INVALID NAME	Invalid file name. No special characters allowed. A through Z, and 0 through 9 are only valid characters. Also occurs if * or wild character construction filenames are specified to a command that does not allow it.
? NON-EXISTENT FILE	Non-existent file. File does not exist in device directory.
? DELETE OLD FILE	Delete old file before giving command that would create file with same name.
? RD/WT DEV ERR	Device error on either input or output device. Check that output device is write-enabled.
? CHECKSUM ERROR	Checksum error during load command.
? END-OF-MEDIUM	End of medium. Occurs during input operations when the program attempts to input and the file is at an end. File in storage is probably wiped out.

? PROGRAM OVER-FLOW	Program too large to load within existing buffer space.
? INVALID COMMAND	The command entered was not recognized by the utility programs.
? "DO" FILE MUST BE ON SYSTEM DEVICE	The command file for the DO command must be on the system or load device.
? LOGICAL DEVICE NOT ASSIGNED	A command used a logical device mnemonic that had not been assigned to a physical device mnemonic via the ASG command.
? NO DEVICE DEFAULTS ALLOWED	The device mnemonic is missing for a command that requires it.

5.10 XTECO - TEXT EDITOR

The XTECO - XXDP+ text editor program enables the user of XXDP+ to create and edit ASCII text files. All editing can be done by using a few simple commands.

XTECO is a character oriented editor. One or more characters in a line can be modified without retyping the rest of the line. XTECO does not require that line numbers be part of the text.

The input file for a given editing operation is the file to which the user wishes to make changes. If the user is using XTECO to create a new file, there is no input file. The output file is either the newly created file, or the edited version of the input file.

In general, the editing process proceeds as follows. The user specifies the file he wishes to edit, and then a block of text is read into core. The user modifies the text by using the various editing commands. He then appends additional blocks of text and edits them until the entire file has been edited. He next outputs the edited file and closes it.

The XETCO program shares memory with the XXDP+ monitor and uses certain functions made availble by the monitor. The EXIT command is used to return to the monitor.

XTECO does not contain read/write drivers for devices. These drivers are resident on the system device as SYS files. They are transferred to memory as they are needed. As a result, XTECO runs in 8K memory.

5.10.1 XTECO Command Description

The prompt for the XTECO command mode is the * character. The XTECO program provides several of the commands available under the update utility programs, in addition to those commands provided for editing purposes. XTECO commands are of two types: Non-Edit and Edit. The NON-EDIT commands are:

DIR	Update Equivalent.
TYPE	Update Equialent.
PRINT	Update Equivalent.
(CR)	Carriage return is the non-edit type comand string terminator.
EXIT	Update Equivalent.

The EDIT commands are:

TEXT	XTECO Unique.
EDIT	XTECO Unique.
TECO	XTECO Unique.
(CTRL-Z)	Control-Z is the edit type command string terminator. Control-Z is not available on 2500 keyboards. Use EX.
I	Used to move pointer one or more lines.
C	Used to move pointer one or more characters.
J	Used to move pointer to beginning of text in core.
ZJ	Used to move pointer to end of text in core.
S	Used to search for a character sequence in text in core.
N	Used to search core and remainder of input file for a specified character sequence.
T	Used to type one or more text lines.
D	Used to delete one or more characters.
K	Used to delete (kill) one or more text lines.
I	Used to insert ASCII text into the text buffer.
A	Used to append one or more text blocks to text buffer.
EX	Outputs edited file to output device and closes output.
(ESC)	Echoes one $. Used to terminate an edit command.
(ESC)(ESC)	Echoes two $. Used to terminate last edit command and to cause execution of entire command string.

Note: The user should be aware of the use of the following special characters:

CTRL-C	Used to exit out of any command and return to command mode. Causes an open output file to be closed. The user must be careful not to type CTRL-C, unless he wishes to abort his operation. This is specially true when editing a file, as all work will be wasted.
CTRL-O	Used to stop printing on the console terminal as when typing multiple lines of text when editing a file.
CTRL-U	Used to empty out contents of keyboard buffer, as when the user wishes to start typing his command sequence all over again.
RUBOUT or DELETE	Used to remove one or more characters typed from command or text string. One depression of the rubout key removes one character.

5.10.2 TEXT, EDIT, and TECO Commands

TEXT, EDIT and TECO commands are the basic commands provided to create or edit and ASCII text file.

The TEXT command is used when the user wishes to create a new text file. The TEXT command does not require an input file, only an output file. All editing commands are available with the exception of the A (append) command which becomes a no-op command when no input file exists.

The EDIT command is the general purpose command for editing an existing text file. It permits the user to edit and input file on one type of device and to output the edited file to a different type device. All editing commands are available when under the EDIT command.

The TECO command is a specialized version of the edit command. Under the TECO command the input and output device/drive must be the same (it assumes they are the same), and must be random access type devices (disk, floppy). The edited output file takes its name from the name and extension of the input file, and the input file is renamed to a BAK extension (for backup). All editing commands are availble. Individual command descriptions follow.

5.10.2.1 TEXT Command. The TEXT command is used to create a new text file. The format of the TEXT command is:

TEXT OUTDEV:FILNAM.EXT(CR)

Where OUTDEV: is any directory device. If it is omitted, the system device is used. The program will type:

Make output ready. Type (CR) when ready.

Ensure that the output device is ready and write-enabled. Press the RETURN key on the console terminal when ready to proceed. The program is now in edit mode, and only editing type commands are valid. The program prompts the user by typing quote (").

The user can at this point type and edit his text. (ESC)(ESC) EX (ESC) is used to return to XTECO command mode and write the new text as a file on the medium.

5.10.2.2 EDIT Command. The EDIT command permits the user to edit a text file from a specified input device, and to output the edited text file to a specified output device. The command format is:

EDIT OUTDEV:FILNAM.EXT=INDEV:FILNAM.EXT(CR)(CR)

Both OUTDEV: and INDEV: must be directory devices. If they are omitted, the system device is used.

The program will type:

Make output ready. Type (CR) when ready.

Ensure that the output device is ready and write enabled. Press the RETURN key on the console terminal when ready to proceed. The program is now in edit mode, and only editing type commands are valid. The program prompts the user by typing quote (").

The user can at this point type and edit his text. (ESC)(ESC) EX (ESC) is used to return to XTECO command mode and write the new text as a file on the medium.

5.10.2.3 **TECO Command.** The TECO command is a specialized, short hand version of the edit command. When using the TECO command the input device/drive must be the same as for output. In addition, the command is reserved for use only with random access devices. The TECO command format is:

TECO DEV:FILNAM.EXT(CR)

Where DEV: is any random access device. If DEV: is omitted, the system device is used. It is important that the user be aware of the mechanics involved in the operation of the TECO command. The sequence is as follows:

Step 1. Open input file.

Step 2. Open output file, and assign it a .TMP extension.

Step 3. Edit operations are performed by operator.

Step 4. Editing done. Output edited file to .TMP file.

Step 5. Close the .TMP file.

Step 6. Rename the .TMP file to same name and extension as the input file.

Step 7. Rename the input file to a BAK extension.

The input device must not contain a file with the same name as the input file and BAK extension. It is deleted in the process of renaming the input file to a BAK extension. Note that the TECO command must not be used to edit a file which has the extension BAK. The file must first be renamed to another extension. The program will type:

Make output ready. Type (CR) when ready.

As shown in the above example, the escape used to terminate an alpha-numeric argument may also serve as one of the two altmodes necessary to terminate a command string.

5.10.3 XTECO Edit Commands

The APPEND (A) command reads in the next block of text from the input device and adds it to the contents of the text buffer in core.

5.10.4 Buffer Pointer Positioning Commands

Since XTECO is a character-oriented editor, it is very important that the user understand the concept of the buffer pointer. The position of the buffer pointer determines the effect of many of the editing commands. For example, insertion and deletion always takes place at the current position of the buffer pointer.

The buffer is the current text contents in core, from the first character, up to and including the last character.

The buffer pointer is simply a movable position indicator. It is always positioned between two characters in the buffer, or before the first character in the buffer, or after the last character in the buffer. The pointer may be moved forward or backward over any number of characters.

5.10.4.1 J Command. This command moves the pointer to the beginning of the buffer, i.e., immediately before the first character in the buffer.

5.10.4.2 ZJ Command. This command moves the pointer to the end of the buffer, i.e., to position following last characters in the buffer.

5.10.4.3 C Command. This command moves the pointer one character in the buffer. The C Command may be preceded by a (decimal) numeric argument. The command NC moves the pointer forward over N characters. The command -NC moves the pointer backward over N characters. (The pointer cannot be moved beyond the ends of the buffer.)

5.10.4.4. L Command. This command is used to advance the buffer pointer on a line-by-line basis. The L command takes a numeric argument, which may be positive, negative, or zero, and is understood to be 1 if omitted.

Ensure that the output device is ready and write-enabled. Press the (CR) key on the console terminal when ready to proceed. The program is now in edit mode, and only editing type commands are valid. The program prompts the user by typing quote (").

At this point type and edit this text. (ESC)(ESC) EX (ESC) is used to return to XTECO command mode and write the new text as a file on the medium.

5.10.5 Editing Command String Syntax

XTECO commands may be given one at a time. However, it is usually more convenient to type in a command string containing several commands that form a logical group. An example of a command string is shown below.

"IHEADING$NTAG:$2LT$$	Inserts word heading, searches for string tag:, moves pointer forward 2 lines and types line pointed to.

Command strings are formed by typing one command after another, separated by one escape. Command strings are terminated by typing two consecutive escapes.

Execution of the command string begins only after the double escape has been typed. At that point, each command in the string is executed in turn, starting at the left. When all commands have been executed, XTECO prints another quote, indicating readiness to accept another command.

If some commands in the string cannot be executed because of a command error, execution of the command string stops at that point, and an error message is printed. Commands preceding the bad command are executed. The bad command and those following it are not executed.

5.10.6 Command Arguments

There are two types of arguments for XTECO editing commands. Some commands require numeric arguments and other commands require alpha-numeric (text) arguments. Numeric arguments are decimal integers. Numeric arguments always precede the command to which they apply. A typical example of a command taking a numeric argument is the command to delete three characters: 3D.

Alpha-numeric arguments are textual arguments meant to be interpreted as ASCII code by XTECO. Alpha-numeric arguments always follow the command to which they apply, and they must always be terminated by an escape (ESC). Examples of alpha-numeric arguments are (1) text to be inserted, and (2) character strings to be searched for.

Example:

*ISOMETHING$$	:The argument is something

Suppose the buffer pointer is positioned at the beginning of line 8 or at some position within line 8.

The command NL, where N>0, advances the pointer to the beginning of line 8+N.

The command -0L moves the pointer to the beginning of Line 8. If the pointer is already at the beginning, nothing happens.

The command -NL moves the pointer back to the beginning on line 8-N.

Note that the execution of the APPEND (A) command does not change the position of the buffer pointer.

5.10.7 Text Type-Out Commands

5.10.7.1 T Command. Various parts of the text in the buffer can be typed out for examination by use of the T command. Just what is typed out depends on the position of the buffer pointer and the argument given. The T command never moves the buffer pointer.

The T command types out everything from the buffer pointer through the next line feed. Thus, if the pointer is at the beginning of a line, the T command causes that line to be typed out. If the pointer is in the middle of a line, T causes the portion of the line following the pointer to be typed.

The Command NT (N>0) is used to type out N lines, i.e., everything from the buffer pointer through the nth line feed following it.

The user, especially one new to XTECO, should use the T command often to make sure the buffer pointer is where he thinks it is.

During execution of any T command, the user may stop the terminal output by typing the CTRL-O character. The printing stops and execution of the remainder of the command string is aborted. Therefore, long type-outs should be restricted to single command, command strings.

5.10.8 Deletion Commands

5.10.8.1 D Command. Individual characters are deleted by using the D command. The command D deletes the character immediately following the buffer pointer. The command ND, where N>0 deletes the 0 characters immediately following the pointer.

5.10.8.2 K Command. Lines are deleted by using the K command. The K command may be preceed by a numeric argument, which is understood to be a 1 if omitted. The command NK (N>0) deletes everything from the current position of the buffer pointer through the Nth line feed character following the pointer.

At the conclusion of a D or K command the buffer pointer is positioned between the characters which precede and follow the deletion.

5.10.9 Insertion Command

The only insertion command is the I command. The ASCII text that is to be inserted into the buffer is typed immediately after the letter I. The text to be inserted is terminated by an altmode.

Any ASCII character except null, altmode, rubout, CTRL-C, CTRL-O, and CTRL-U may be included in the text to be inserted.

If a carriage return is typed in an insertion, it is automatically followed by a line feed. The text to be inserted is placed in the buffer at the position of the buffer pointer, i.e., between the characters. At the conclusion of the insertion command the buffer pointer is positioned at the end of the insertion.

Any number of lines may be inserted with a single I command. However, it is recommended that no more than 10 to 20 lines should be inserted with each I command.

5.10.10 Output Command

The only output command available with XTECO is the EX (EXIT) command. The EX command is used to conclude an editing job with a minimum of effort. Its use is best shown by an example:

Suppose the user is editing a 30 page file and suppose the last actual change to the file is made on page 10. At this point the user enters the command:

```
EX$$
*
```

The action of XTECO is (1) to rapidly move all of the rest of the input file to the output file, (2) close the file, and (3) to return to command mode so that the user may give other NON-EDIT mode commands.

5.10.11 Search Commands

In many cases the simplest way to position the buffer pointer is by using a character string search. A search command causes XTECO to search through the test until a specified string of characters is found, and then to position the pointer at the end of this string. There are two search commands.

5.10.11.1 S Command. The S command is used to search for a character string within the buffer. The string to be searched for is specified as an alpha-numeric argument following the S command. This argument must be terminated by an escape.

Execution of the S command begins at the position of the buffer pointer and continues to the end of the buffer. If the specified string is not found an error message is printed and the buffer pointer is set to the point where the search began.

5.10.11.2 N Command. The N command works just like the S command. The difference is that an S command ends at the end of the buffer, whereas the N command does not. An N search begins like an S Search but if the character string is not found in the current buffer an automatic A (Append) command is executed and the search continued until the search is successful or the input file exhausted.

If the N command finds the specified string the pointer is positioned at the end of the string found. If the string is not found, an error message is printed and the pointer is set at the beginning of the buffer. Since a good part of the file may already have been output to the output device, the user may have no other choice than to exit via the EX command, and to reopen the file and try the N search again with a character string that can be found.

CAUTION

When attempting to search it is very easy to overlook an occurrence of the search string preceding the one the user desires. For example, he may want to move the pointer after the word "and" but erroneously position the pointer after a preceding occurrence of a word like "thousand".

For this reason, the user is strongly urged to execute a T command to ascertain the position of the pointer after each search command.

5.10.12 XTECO Errors

Error messages generated by XTECO are included in those generated by the UPDATE program. Refer to the descriptions given for the UPDATE programs. In addition, one error message is generated by XTECO when a search for a character string by either the S or N commands fails. In that case XTECO types:

Not Found: (ASCII String)

5.11 XXDP+ HELP TEXT FILE (HELP.TXT)

5.11.1 XXDP+ Monitor Commands:

The XXDP+ monitor commands are as follows:

R FILENAME ADDR	Loads and starts program at specified address (address defaults to XFR address of program).
L FILENAME	Load Program.
S ADDR	Start Program at specified address (address defaults to XFR address of program).
C FILENAME?QV	Executes chain file (/QV gives optional quick verify mode).
D/L/F	Directory (optionally on printer (/L) or in short form (/F).
F	Set console fill count.
E UNIT #	Enable alternate system device.
H/L	Help file (optionally on printer (/L)).

NOTE

The XXDP+ monitor can run either the old or the new XXDP utilities, but the old XXDP monitor cannot run the new utilities. The new monitor is a 2K program (unlike the old XXDP monitor, which as a 1.5K program). However, all old diagnostics are still supported, regardless of size.

5.11.2 Boot Time Questions

The monitor will ask you at boot time whether you have 50 Hz and whether you have an LSI. This is information that it cannot autosize for and which it must pass on to certain new types of diagnostics. Both questions default to NO if you type a (CR). The monitor also asks the date at boot time. You can take the default of 1-Jan-70 by typing (CR).

NOTE

XXDP+ utilities use the date supplied to the monitor at boot time.

5.11.3 Enable Command

The enable command has been upgraded to handle any device (formerly handled just cassettes). If you are booted from one drive and want the monitor to behave as if it had been booted from another, just type E UNIT # (e.g., E 2 for Drive #2).

5.11.4 Directory Utility

The special utility HUDI??.SYS is autoloaded by the monitor when the "D" command is issued by the operator. It performs the directory function and overplays the contents of memory (except the monitor).

5.11.5 XXDP+ Utilities

5.11.5.1 XXDP+ UPD1: Utility Commands.

CLR	Clear Memory
XFR	Set Transfer Address
MOD ADDR	Modify Core
LOAD DEV:FILENAME	Load Program
DUMP DEV:FILENAME	Dump Program
CORE	Display Core Limits for Dump
HICORE ADDR	Set hicore limit for dump
LOCORE ADDR	Set locore limit for dump

DEL DEV:FILENAME	Delete Program
BOOT DEV:	Boot Device

NOTE

The UPD1 utility supports all standard XXDP devices, but only the above command set (for example, note that the "pip" and "start" commands are no longer available.

UPD1 needs to be used only in cases where there is not sufficient memory to allow the LOAD/MOD/DUMP operation on a binary file. Otherwise, UPD2 may be used.

It is a 4K program and it executes in low core (where it loads). It destroys the monitor image in high core (it uses that area as a buffer). Therefore, the monitor must be rebooted when finished.

The load device must remain online throughout execution, since UPD1 uses retrievable device drivers from that medium as it executes.

5.11.5.2 XXDP+ UPD2 Utility Commands.

CLR	Clear memory.
ZERO DEV:	Zero device.
SAVM DEV:	Save monitor to disk.
SAVE DEV:FILENAME	Save monitor to tape.
XFT	Set transfer address.
LOAD DEV:FILENAME	Load program.
DUMP DEV:FILENAME	Dump program.
CORE	Display core limits for dump.
HICORE ADDR	Set hicore limit for dump.
LOCORE ADDR	Set locore limit for dump.
DIR DEV:FILENAME	Directory (optionally for a particular file).
PATCH DEV:FILENAME	Patch disk file.
ASG PHDEV: LOGDEV	Assign logical name.
READ DEV:FILENAME	Read file for validity.
PIP DEV:FILENAME=	
DEV:FILENAME/F	Transfer file. The /F switch (for "fast") causes UPD2 to write files to tape without checking to see whether they are already present, thus potentially duplicating the name.

EOT DEV:	Write end of tape mark.
DEL DEV:FILENAME	Delete file.
REN DEV:FILENAME= DEV:FILENAME	Rename file.
DO FILENAME	Execute command file. It must be on system device and it must not contain exit CMD.
DRIVER DEV:	Load R/W driver into core.
BOOT DEV:	Boot system.
PRINT DEV:FILENAME	Print file (on LPT).
TYPE DEV:FILENAME	Type file (on TTY).
MOD ADDR	Modify core.
Copy DEV:=DEV:	Copy volume (need like devices).
EXIT	Return control to monitor.

NOTE

The UPD2 utility supports all standard XXDP devices, but the START command is no longer supported (necessary to use monitor for that). It is a 6K program that executes in low core (where it loads) and makes calls to the monitor as it executes. The EXIT command returns control to the monitor.

UPD2 uses retrievable I/O drivers from the system device as it executes, therefore that device must remain online during its execution.

5.11.5.3 PIP/FILE Rules. When a PIP or a FILE is done between disks or tapes and the file being transferred is already present on the output device, the following occurs (D and T stand for disk and tape):

	Single File Mode	Wild Card Mode
FILE	D: Says "Delete Old" T: Says "Delete Old:	D: Autodeletes T: Autodeletes
PIP	D: Says "Delete Old" T: Says "Delete Old:	D: Autodeletes T: Autodeletes
FILEF	D: Says "Delete Old" T: Duplicates the name	D: Autodeletes T: Duplicates the name
PIP/F	D: Says "Delete Old" T: Duplicates the name	D: Autodeletes T: Duplicates the name

NOTE

The reason that the fast mode commands (FILEF command and the PIP command with the /F switch) create duplicate name on tape is that the output tape is not rewound to search for the presence of a file with the same name as that of the specified output file.

The FILE: and FILEF commands, although still supported, have been officially replaced by the PIP and PIP/F, which are functionally identical (although not syntactically identical: the PIP needs output filename, which the FILE doesn't take).

5.11.5.4 XXDP+ XTECO Utility Commands.

DIR DEV:FILENAME	Directory (optionally of specific file).
PRINT DEV:FILENAME	Print file (on LPT).
TYPE DEV:FILENAME	Type File (on TTY).
EDIT DEV:FILENAME= DEV:FILENAME	Edit ASCII File.
TECO DEV:FILENAME	Edit single file in place.
TEXT DEV:FILENAME	Create new text file.
C	Move pointer by characters.
L	Move pointer by lines.
J	Move pointer to start of text in memory.
ZJ	Move pointer to end of text in memory.
S	Search for character sequence.
N	Search to end of file for sequence.
T	Type lines of text.
D	Delete characters.
K	Delete lines of text.
I	Insert text.
A	Append more text from file into memory.
(ESC)	Terminate a command.
(ESC)(ESC)	Terminate last command and execute any command.
-C	Return to command mode.
-U	Restart command sequence.
RUBOUT	Ignore last character.
-Z	Terminate input mode.
EXIT	Return control to monitor.

5.11.5.5 Zeroing a System Device. If any attempt to zero the system load device is made, the utility program will give a warning to load another device driver. This warning is given because device drivers reside as SYS files on the system device. Before zeroing the system device, two device drivers must be in memory: one for the system device and one for the device used to rebuild the system device. The ZERO command provides the first driver. The driver command can be used to obtain the second driver. If both devices are of the same type, then only one driver is needed.

5.11.6 Miscellaneous Operations

5.11.6.1 Non-Standard CSR. If the system device has a non-standard CSR, patch location 20 of the monitor and location 24 of the read/write driver to the correct value. In patching the monitor, it is necessary to patch the .SYS file and then to do a SAVM to the disk.

5.11.6.2 Chaining Utilities. The monitor chain command can be used to execute the UPD2 utility as follows. Let the chain file contain the command "R UPD2". After this, place whatever UPD2 commands wished executed. At the end of these, put the UPD2 command Exit. Then follow with any other legal monitor commands.

The monitor will only chain files which end with the extension BIC, so it is necessary to copy or rename UPD2.BIN to UPD2.BIC before chaining.

5.6.11.3 How to Merge XXDP+ Software onto Your Disk.

Step 1. Boot from old XXDP pack.

Step 2. Run old UPD2.

Step 3. File OLDEV:=NBEWDEV:*.SYS (This brings across the new monitors and the new drivers and the directory utility HUDI??.SYS which is automatically invoked by the monitor on a "D" command.)

Step 4. File OLDEV:=NEWDEV:*.BIN (This brings across the new UPD2, UPD1, XTECO, and DXCL utilities.)

Step 5. File OLDEV:=NEWDEV:HELP.TXT (Brings across this file.)

Step 6. LOAD NEWDEV:HMXX??.SYS (This is the new monitor for the system device: the XX is the device mnemonic, as explained below.)

Step 7. SAVM OLDEV: (Puts new monitor image in boot area of old pack.)

Step 8. The old UPD2 is: Still running the Boot device and you'll be running the new monitor.

5.11.6.4 How to Merge Your XXDP Tape onto an XXDP+ Tape.

Step 1. Boot from new tape.
Step 2. Run new UPD2.
Step 3. Delete OLDEV:THDP.SAV.
Step 4. Delete OLDEV:TMDP.SAV.
Step 5. Delete OLDEV:UPD1.BIN.
Step 6. Delete OLDEV:UPD2.BIN.
Step 7. Delete OLDEV:XTECO.BIN.
Step 8. FILEF NEWDEV:=OLDEV:*.*.

5.11.6.5 How to Build XXDP+ Disk From Scratch.

Step 1. Zero the disk.
Step 2. Load monitor for that disk type .
Step 3. Do a "SAVM" to that disk.
Step 4. Transfer all desired files to that disk.

5.11.6.6 How to Build XXDP+ Tape From Scratch.

Step 1. Zero the tape.
Step 2. Load MM monitor.
Step 3. Do a SAVE of the MM monitor to the tape (EXT = SAV).
Step 4. Load the MT monitor.
Step 5. Do a SAVE of the MT monitor to the tape (EXT = SAV).
Step 6. Transfer desired files to the tape.

5.11.7 XXDP+ Naming Conventions

5.11.7.1 XXDP+ Monitor and Device Driver Naming Conventions.

Device	Monitor File	Driver File	Devices Supported
CT	HMCT??.SYS	HDCT??.SYS	TA11/TO60
DB	HMDG??.SYS	HDDB??.SYS	RP01/5/6
DD	HMDD??.SYS	HDDD??.SYS	DL11/TO58
DK	HMDK??.SYS	HDDK??.SYS	RK11/RK05
DL	HMDL??.SYS	HDDL??.SYS	RL11/RL01
DM	HMDM??.SYS	HDOM??.SYS	RK611/RK06/7
DP	HMDP??.SYS	HDDP??.SYS	RP11/RP02/3

Device	Monitor File	Driver File	Devices Supported
DR	HMDR??.SYS	HDDR??.SYS	RM02/3
DS	HMDS??.SYS	HDDS??.SYS	RS03/4
DT	HMDT??.SYS	HDDT??.SYS	TC11 DECTAPE
DX	HMDX??.SYS	HDDX??.SYS	RX11/RX01
DY	HMDY??.SYS	HDDY??.SYS	RX211/RX02
MM	HMMM??.SYS	HDMM??.SYS	TM02/TM03 Tapes
MT	HMMT??.SYS	HDMT??.SYS	TH11 Tapes
KB		HDKB??.SYS	PPT RDR (LOW SP)
PT		HDPT??.SYS	PPT PCH (LOW SP)
PR		HDPR??.SYS	PPT RDR (HIGH SP)
PP		HDPR??.SYS	PPT PCH (HIGH SP)

NOTE

The device mnemonics are unchanged from previous XXDP, but all other names are now derived from these mnemonics. All monitors start with HM and all drivers start with HD. The ?? in each file is the Rev/Patch level.

Files with the extension .SYS are binary files that cannot be directly executed by the operator.

Tape monitors must be given the extension .SAV when placed at the start of a magtape for booting, and .SYS when residing in the normal file area on either tape or disk.

5.11.7.2 XXDP+ Utility Naming Conventions. The file names on the XXDP+ utilities are as follows:

```
UPD1.BIN   UPDATE UTILITY #1
UPD2.BIN   UPDATE UTILITY #2
XTECO.BIN TEXT EDITOR
DXCL.BIN   DEC/X11 CONFIGURATOR LINKER

END OF HELP.TXT
```

5.12 SUPERVISORY DIAGNOSTICS

5.12.1 Five Steps of Execution

A supervisor diagnostic should be started using normal XXDP procedures. When the diagnostic is started, the following steps will occur:

Step 1. When the diagnostic issues the prompt DR, enter START command. This is not the same as the XXDP START command, which has been already issued in response to the XXDP dot prompt. This START command can take a number of switches and flags (all optional). However, in order to use the program, you need to say something like this:

```
STA/PASS:1/FLAGS:HOE
```

Note the following:

- Only the first three characters of this or any command at the DR level need to be typed.
- The PASS switch specifies how many passes you desire. A pass consists of running the full diagnostic against all units being tested (this is explained shortly). One pass is specified in the above example.
- The FLAGS switch may specify any of a number of flags, but the main useful ones are:

LOE	Loop on error
HOE	Halt on error
IER	Inhibit error printout

The HOE flag is specified in the above example.

Step 2. When you have typed in a START command, the diagnostic comes back with the question $ UNITS? to which you should respond by typing in the number of devices you wish to test.

The number of units depends on the target device of the diagnostic. For example, if the diagnostic is directed at a disk drive, then the number of units would be the number of drives to be tested. Whereas if the diagnostic was directed at the disk controller, then the number of units would be the number of controllers. The target device of a diagnostic can always be determined by inspecting the HEADER statement near the beginning of the source code. One of the operands of this HEADER statement should be the device type of the diagnostic.

Step 3. When you have typed in the number of units to be tested, the diagnostic asks you the HARDWARE QUESTIONS. The answers to these questions are used to build tables in core, called hardware P-tables. One hardware P-table is built for each unit to be tested.

There are several hardware questions and the entire series are posed N times, where N is the number of units.

This represents a new philosophy in diagnostic engineering. Diagnostics in the future will not be written to autosize or assume standard addresses. Instead, the operator will be asked for all the information needed to test the device.

Step 4. After all the hardware questions for all the units have been answered, you will be asked CHANGE SW? (Y,N). Type Y to be asked the software questions. The answers will be put into the software P-table in the program. The series of questions are asked just once, regardless of the number of units to be tested.

Step 5. After the software questions have been answered, the diagnostic begins to execute the hardware test code. Several things can happen next, depending

on whether a hardware error is encountered and also on what switch values were selected on the START command. Consider the possibilities:

- If no error is encountered, then the diagnostic simply executes the desired number of passes and returns to command mode (prompt DR).

- If an error is encountered, one of three things happens, depending on the settings of the HOE and LOE flags.

  HOE set: The error is reported on the console and the diagnostic returns to command mode.

  LOE set: The diagnostic loops endlessly on the block of code that detected the error.

  Neither HOE nor LOE set: The error is reported on the console and normal execution resumes as if no error had occured.

5.12.2 Table of Command Validity

There are four ways of entering diagnostic command mode. Different subsets of the diagnostic command set are available with each:

How Entered	Legal Commands
• Operator entered "RUN DIAG"	START PRINT DISPLAY FLAGS ZFLAGS EXIT
• Diagnostic has finished all requested passes	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
• Operator interrupted the diagnostic with CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS EXIT

• An error was encountered with the HOE flag set set	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS EXIT

5.13 CONSOLE ODT DEBUGGING FUNCTIONS

Console ODT (Octal Debugging Tool) is a program in the LSI-11/23 microcode that provides for opening, examining, and modifying the content of registers and memory locations, and running or resuming programs from specified addresses. The system enters ODT when the processor HALTs, and displays @ symbol (at) as the prompt for operator entries.

To specifically start ODT, press the front panel BOOT/HALT switch to HALT.

The following is a list of ODT commands and how they are used with the console terminal. In the examples provided, operator entries are shown in bold faced type. Only the commands necessary for implementing the required console functions are retained. Some output characters from the processor do not actually appear on the CRT, i.e., line feed (LF). Regardless, they are still outputs.

All commands and characters are echoed by the processor. Illegal commands are echoed followed by ?. These are followed by (CR), (LF) and @. If a valid command character is received when no location is open (e.g., when having just entered the halt state), the valid command character is echoed and followed by a ?(CR) (LF) @. Opening nonexistent locations will have the same response. The console always prints six numeric characters as address or data. However, the user is not required to type leading zeros for either address or data. If a bus error (timeout) occurs during memory refresh while in the console ODT mode, a (CR) (LF) @ is printed. A list of the ODT commands and their usage follows:

COMMAND	SYMBOL	USE
Slash	/	This command is used to open an LSI-11 bus address, processor register, or processor status word and is normally preceded by other characters which specify a location. In response to /, console ODT prints the contents of the location (i.e., six characters) and then a space (ASCII 40). After printing is complete, console ODT waits for either new data for that location or a valid

COMMAND	SYMBOL	USE

close command. The space character is issued so that the location's contents and possible new contents entered by the user are legible on the terminal.

Example:

@ **001000/**012525(SPACE)

where:

@	=	Console ODT prompt character.
001000	=	Octal location in the LSI-11 bus address space desired by the user (leading zeros are not required).
/	=	Command to open and print contents of location.
012525	=	Contents of octal location 1000.
SPACE	=	Space character generated by console ODT.

The / command can be used without a location specifier to verify the data just entered into a previously opened location. The / is recognized only if it is entered immedialely after a prompt character. A / issued immediately after the processor enters ODT mode causes a ?(CR)(LF) to be printed because a location has not been opened.

Example:

@ **1000/**012525(SPACE) **1234(CR)**(CR)(LF)
@ **/**001234(SPACE)

where:

first line	=	new data of 1234 entered into location 1000 and locations closed with (CR)
second line	=	a / was entered without a location specifier and the previous location was opened to reveal that the new contents were correctly entered into memory.

Carriage Return	(CR)	This command is used to close an open location. If the contents of the open location are to be changed, (CR) should be preceded by the new value. If no change to the location is necessary, (CR) does not alter its contents.

COMMAND	SYMBOL	USE

Example:

@ **001000/**12525 **(CR)**
@ **/**012525

or

Example:

@ **001000/**012525 **15126421 (CR)**
@ **/**126421

(CR) is used to close location 1000 in both examples. Note that in the second example, the contents of location 1000 were changed and that only the last 6 digits entered were placed in location 1000.

Line Feed (LF)↓

This command is used to close an open location and then open the next contiguous location. LSI-11 bus addresses and processor registers are incremented by 2 and 1 respectively. If the PS is open when a (LF) is issued, it is closed and a (CR)(LF)@ is printed; no new location is opened. If the open location's contents are to be changed, the new data should precede the (LF). If no data is entered, the location is closed without being altered.

Example:

@ **R2/**123456(SPACE)(CR)(LF)
@ **R3/**054321(SPACE)

In this case, the user entered (LF) with no data preceding it. In response, console ODT closed R2 and then opened R3. When a user has the last register, R7, open, and issues (LF), console ODT opens the beginning register, R1. When the user has the last LSI-11 bus address open of a 32K word segment and issues (LF), console ODT opens the first location of that same segment. If the user wishes to cross the 32K word boundary, he must re-enter the address for the desired 32K word segment (i.e., console ODT is module 32K word). This operation is the same as that found on all other PDP-11 consoles.

Example:

@ **R7/**000000(SPACE)(CR)(LF)
@ **R0/**123456(SPACE)

or

@ **577776/**000001(SPACE)**(LF)**(CR)(LF)
@ **477776/**125252(SPACE)

COMMAND	SYMBOL	USE
		Unlike other commands, console ODT does not echo the (LF). Instead it prints (CR), then (LF) so that terminal printers operate properly. In order to make this easier to decode, console ODT does not echo ASC11 0, 2, or 10 but responds to these three characters with ?(CR)(LF)@.
Internal Register Designator	R or S	Either character when followed by a register number, 0 to 7, or PS designator, S, will open that specific processor register. The $ character is recognized to be compatible with ODT-11. The R character was introduced for the convenience of one key stroke and because it is representative of what it does. Example: @ **$0/**000123(SPACE) or @ **R7/**000123(SPACE)(LF) @ **R0/**054321(SPACE) If more than one character is typed (digit or S) after the R or $, console ODT uses the last character as the register designator. There is an exception, however: if the last three digits are 077 or 477, ODT interprets it to mean the PS rather than R7.
Processor Status Word	S	This designator is for opening the PS (processor status word) and must be employed after the user has entered an R or $ register designator. Example: @ **RS/**100377(SPACE)**0(CR)**(CR)(LF) @ **/**000010(SPACE) Note that trace bit (bit 4) of the PS cannot be modified by the user. This is done so that PDP-11 program debug utilities (e.g., ODT-11), which use the T bit for signal-stepping, are not accidentally harmed by the user. If the user issues a (LF) while the PS is open, the PS is closed and ODT prints a (CR)(LF) @. No new location is opened in this case.
Go	G	The G (GO) command is used to start execution of a program at the memory location typed immediately before the G.

COMMAND	SYMBOL	USE

Example:

@ **200 G**

The console ODT sequency for a G, after echoing the command character, is as follows:

- Print two nulls (ASC11 0) so the LSI-11 bus initialize that follows does not flush the G character from the DLV11 serial line interface or the console emulator.
- Load R7 (PC) with the entered data. If no data is entered, 0 is used. (In the above example, R7 is equal to 200 and that is where program execution begins).
- The PS, and floating point status register if the MMU is present, is cleared to 0.
- The LSI-11 bus is initialized by the processor asserting BINIT L for 12.6 microseconds (at 300 ns microcycle), negating BINIT L, and then waiting for 110 microseconds (at 399 ns microcycle).
- The service state is entered by the processor. If there is anything to be serviced, it is processed. If the BHALT L bus signal is asserted, the processor reenters the console ODT state. This feature is used to initialize a system without starting a program (R7 is altered). If the user wants to single-step his program he issues a G and then successive P comands, all done with the BHALT L bus signal asserted.

Proceed P

The P (Proceed) command is used to continue or resume execution at the location pointed to by the current contents of the PC(R7).

Example:

@ **P**

Program execution resumes at the address pointed to by R7. After the P is echoed, the console ODT state is left and the processor immediately enters the state to fetch the next instruction. If the BHALT L bus signal is asserted, it is recognized at the end of the instruction (during the service state) and the processor enters the console ODT state. Upon entry, the content of the PC (R7) is printed. In this fashion, a user can single-instruction step through a program and get a PC "trace" displayed on his terminal.

APPENDIX A

LIST OF REFERENCES

APPENDIX A

LIST OF REFERENCES

2515 Computer-Aided Test System	System Operating Manual 2515-0100, GenRad, Inc. System Service Manual 2515-0101, GenRad, Inc. System Diagrams And Parts Lists 2515-0102, GenRad, Inc.
LSI-11/23 Processor	Digital Microcomputer Processor Handbook 1978-80, Digital Equipment Corporation.
ASCII Keyboard	2515 Computer-Aided Test System Operating Manual 2515-0100, GenRad, Inc.
Serial Line Interface	Model DLV-11J PM-DLV11J Serial Line Interface Manual, Digital Equipment Corporation.
Floppy Disk Drive	Model 9409T, Produt Specification for Flexible Disk Drive Model 9409T (96TPI), Spec. 77653489, CD 3, Rev B, dated Sept. 1981, Magnetic Peripherals Inc., Subsidiary of Control Data Corp.
Winchester Disk Drive	Model WINC05, WINC05 Operations and Maintenance Manual, Pub. No. 990016, dtd. Sept. 1982, Advanced Electronics Design, Inc.
Winchester/Floppy Interface AED-WINC05	Model WINC05, WINC05 Operations and Maintenance Manual, Pub. No. 990016, dtd. Sept. 1982, Advanced Electronics Design, Inc.
Monitor Kit, 7"	Model MD1700, MD1703 Series Display Systems, Motorola, Inc., Service Manual, VP42, 12/82; GenRad, Inc. 1765-5541.
Power Supply	Model RSF500, ACDC Electronics, Installation and Operation Manual, RSF Operation Manual, RSF Series Power Supply Modules 10/82/1000

Unibus Memory	Model SuperSTOR-11M, Cambex Corporation, HN 081-088-001, GenRad, Inc. 1765-1123.
Printer/Plotter	Model HP2673A, Service Manual, 2673A, Intelligent Praphics Printer, Hewlett-Packard Company
Q-Bus Expansion	GR2515 Q-Bus Expansion Option, Installation Guide, Publication No. 2515-0113, GenRad, Inc.

APPENDIX B
MNEMONICS AND ABBREVIATIONS

Mnemonic/Acronym	Function/Meaning
A0-A20	Address lines
A-A	Analog-to-analog
AC	Alternating Current
ACC	Accumulation control
ACCUM	Accumulator
ACK	Acknowledge
A/D	Analog-to-digital
ADD	Address
ADDVSUB	Vector address subroutine
ADRCK	Address clock
ADSET	Address selector register
AED	Automated engineering design, or algol extended for design
AIMC	Source/destination codes 240-257
AJ0-AJ17	Q-bus address lines (LSI-11/23 data)
A.L.E.B	High when A is less than equal to B
ALTDAT	Alternate data
ALU	Arithmetic logic unit
ANGCLR	Clear angle
ARA0-ARA3	Access initiator selection logic address bits
ARL	Address register least significant
ARM	Address register most significant
ASEL0-ASEL3	Address select bit 0-3
ATTEN	Attenuator
AV	+5V battery power, not used on LSI-11/23
BBIT	Register B bit, access bit enable
BBS7	Current access is to the I/O page
BS7	Address line to slave device
BD0-BS8	Buffered destination bits (CPU)
BDAL0-BDAL17	LSI-11/23 data/address lines
BDCOK	DC power okay
BDIN	Data input operation in progress

Mnemonic/Acronym	Function/Meaning
BDOUT	Data output operation in progress
BDSCAN0-1	Channel board select bits
BDSEL	Board select
BIAKO	Interrupt acknowledge output
BIAKI	Interrupt acknowledge to highest priority device
BINIT	Initialize, system reset
BIRQ4	Interrupt request priority level 4
BLNKNXT	Blank next
BMT	Keyboard send ready
BNC	Baby "N" connector
BRPLY	Reply
BS0-BS8	Buffered source bits (CPU)
BS7L	Latched address line
BSYNC	Synchronize, bus cycle is starting, valid address on bus
BWTBT	Write byte operation in progress
C	Centigrade
C0-C1	Control output lines
C0-C6	Clock and control lines (Control Panel Boards)
C0-C15	Clock count 0 to clock count 15, buffered
C1A	Control line
CAL	Zero
CASCLK	Clock cycle, cascade
CATS	Computer-Aided Test System
CBIT	Condition code bit
CCLKLD	Load clock
CCNTR0	Clock control bit 0
CCNTR1	Clock control bit 1
CCNTR2	Clock control bit 2
CCNTR3	Clock control bit 3
CCSTK	Clock stack register
CCSTK0	Clock stack register bit 0

Mnemonic/Acronym	Function/Meaning
CCSTKA	Clock stack register A
CCSTKA0	Clock stack register A bit 0
CCSTKA1	Clock stack register A bit 1
CFM	Cubic foot per minute
CH	Channel
CHINIT	Set all channel board registers to 0
CHNL	Channel
CKT	Circuit
CKFIFO	Clock FIFO
CLKHLD	Clock hold input on next transition
CKMDR	Clock memory data register
CLKHLD1	Clock hold for 2 cycles
CLKHOLD	Clock hold
CLKP	Clock part or product
CLKSORD	Clock source data
CLKXY	Clock X or Y
CLOCK	Clock enable
CLR	Clear
CLRBFI	Clear buffer full interrupt, code 266
CLROVLD	Overload Control
CLRQS1	Clear Q-bus slave interrupt, code 267
cm	Centimeter
CMD0-3	Vector generator comand bits
CMDONE	Command done
CNDRTNGD	Condition return good
CNDTRU	Condition true
CNDTRUEL	Condition true, latched
CNURTE	Enable conversion
CNVT	Convert
C0	Control line
COL0-3	Memory control select bits
COMP	Composite
COMPLE	Complement

Mnemonic/Acronym	Function/Meaning
COMPVID	Composite video
COS	Cosine
CPU	Central processing unit
CRT	Cathade ray tube
CS	Chip select
CSR	Control and status register
CSWCLK	Clock control
CYCLE 2,3	Clock cycles 2 and 3
CYCLE 4	Base clock
D0	Destination line, PROM output bit 9, least significant bit
D0-D1	Shift control bits
D1	Destination line, PROM output bit 10
D2	Destination line, PROM output bit 11
D3	Destination line, PROM output bit 12
D4	Destination line, PROM output bit 13
D5	Destination line, PROM output bit 14
D6	Destination line, PROM output bit 15
D7	Destination line, PROM output bit 16
D8	Destination line, PROM output bit 17, most significant bit
D665	Destination, store lower bus byte in map register.
D666	Destination, store upper bus byte in map register.
DAC	Digital-to-analog converter
DACL	Digital-to-analog converter, low
DACSEL0-1	DAC source select bits 8-9
DAL0-DAL17	Data/address lines
DATO	Write operation, 16-bit data word output
DATOB	Write byte operation, 8-bit data byte output
DAT0-DAT7	Parallel data lines
dB	Decibel

Mnemonic/Acronym	Function/Meaning
DB0-DB15	Microprocessor data bus lines
DBCONT	Destination decode signal
DBIT0	Destination bit 0
DBLLFTSHFT	Double left shift if high
DBLRTSHFT	Double right shift if low
DC	Direct current
DCM0	Add 1 clock for destination device
DCM0	Destination device line, add one clock
DCM1	Destination device line, add two clocks
DCM2	Destination device line, add four clocks
DECIN	Input decrement value
DIN	Data in
DIROUT	Direct output
DIVSTEP	Divide step
DLY	Delay
DMA	Direct memory access
DMOS	Direct memory output select
DO0-9	Memory data bits
D0L-D3L	Decoded latched destination address bits
DOTCLK	Dot matrix clock cycle
DOUT	Data out
DPST	Double-pole-single-throw
DR0-DR1	Control panel select enable bits
DR1-21	Control panel select bits
DRIVEQ	Drive Q-bus
DROE	Data register output enable
DSPCSW	Display status register
DST8	Destination 8
DSTAI	Destination access initiator
DSTARL	Destination ARL register
DSTARM	Destination ARM register
DSTMAP	Destination MAPMEM register
DSTMCMD	Destination memory command register

Mnemonic/Acronym	Function/Meaning
DSTMDR	Destination memory data register
DSTOFF	Destination OFFMEM register
DTHR	Dither
E	Enable
E0-E15	Address bits
ECLK	Enable clock
EICMD	Enable interrupt on command done
EIKEY	Keyboard interrupt enable
EIVSYNC	Vertical sync interrupt enable
ENABLE	Enable
ENBF	FIFO write enable
ENBFIFO	Enable FIFO
ENBHIGH	Enable High
ENBIVEC	Enable vector interrupt
ENBLMID	Enable Mid
ENBLOW	Enable Low
ENBORFLD	Enable OR field, if low
ENBSTKCNT	Up/down counter enable stack count low
ENBTRANS	Enable transmission
ENDCYC	End cycle
EQUAL	Compared clock and comparator are equal
ERASEBLK	Fast erase from current position to x, y
ERASEXY	Fast erase from 0, 0 to x, y
EXT	External
EXTCLK	External clock
EXTRIN	External read input
F0	FIFO control bit
F1	FIFO control bit
F2	FIFO control bit

Mnemonic/Acronym	Function/Meaning
F3	FIFO control bit
F15	Sign bit of ALU output
FARA0-FARA3	FIFO address lines for service
FDNR	Frequency dependent negative resistor
FEINT	FIFO error interrupt enable
FET	Field effect transistor
FIERR	FIFO input error
FIFO	First in/first out register
FIFOCYC	FIFO write cycle
FIFOST	FIFO status register
FIFOUT	FIFO output signal
FINERA	Precise erase from current position
FINIT	FIFO intialize
FLTX1-X2	Filter subperiod select bits
FOERR	FIFO output error
FORHUN	Keyboard interface timing signal
FS0-FS1	FIFO current status lines
FS0D	FIFO status bit 0
FS1D	FIFO status bit 1
FULBUF	Full buffer
G0-G31	RAM output bits containing address, value, and control
GATE	Gate
GEN	Generator
GND	Ground
GR	GenRad
HI	High
HIGH	High
HLD2	Hold 2

Mnemonic/Acronym	Function/Meaning
HOD	Plus 3/4 full condition
HOFF0-HOFF7	High region offset bits, OFFMEM register
HOLDC	Clock hold signal
HSYNC	Horizontal sync
Hz	Hertz

I0	Instruction bit	0 = Printer interrupt enable
I1	(Printzone)	1 = Key interrupt enable
I2	(Dot strobe)	2 = Key send is ready
I3	(Character strobe)	3 = Interrupt on equal compare, key error
I4	(-DECIN/pad T2)	4 = Printer select
I5	(Paperfeed button)	5 = Paper is out
I6	(Input pad T4)	6 = Printer is busy
I7	(Input pad T5)	7 = Key available
I8	(Same as I3)	8 = Not used

IAKI	Interrupt acknowledge from LSI-11/23
IAKO	Interrupt acknowledge out to Q-bus devices
IC	Instruction counter or integrated circuit
IC0	Input bit 0 to instruction counter I/0
IC1-IC15	Input bit 1-15 to instruction counter I/0
ICCLK	Instruction counter clock input high
ICCLRHI	Instruction counter clear high byte for interrupt high
ICCNT	Suppress count of instruction counter at master clock signal for jump type with interrupt
ICLOAD	Load instruction counter at MC signal for unconditional return
ICST0-ICST3	Instruction counter status bits 0-3
ID	Interrupt disabled or identification
IE	Interrupt enabled
IG1	Interrupt grant line 1, lowest priority
IG2-IG6	Interrupt grant lines 2-6
IG4I	Interrupt grant input, level 4

Mnemonic/Acronym	Function/Meaning
IG40	Interrupt output, level 4
IG6I	Interrupt grant priority level 6 in
IG60	Interrupt grant out, level 6
IG7	Interrupt grant line 7, highest priority
IG7I	Interrupt grant priority level 7 in
IGGO	Interrupt grant, pass to next device
IMDTA	Immediate data
INBCLK	Input B clock
INC0-INC15	PROM increment value bits 0-15
INCR0-INCR4	Increment value bits 0-4
ININ0-ICIN3	Instruction counter input bits 0-3
INIT	Initialize
INITL	Reset signal
INTFF	Interrupt flip-flop
INTR	Interrupt
INTRPT	Interrupt
ION1-4	2 mega-amp current source enable bits
INTGRNT	Grant interrupt
IR1 (Lowest Priority)	Interrupt request line bit 1
IR2-IR6	Interrupt request line bits 2-6
IR6	Interrupt request, level 6
IR7	Interrupt request, level 7
IR7	Interrupt request line bit 7, highest priority
IRQ	Interrupt request from Q-bus
IV0	Interrupt vector input bit 0, least significant
IV0-IV7	Interrupt vector lines 0-7
IV1-IV6	Interrupt vector input bits 1-6
IV7	Interrupt vector input bit 7, most significant
JADENBHI	Jump address register enable high byte
JADENBLO	Jump address register enable low byte

Mnemonic/Acronym	Function/Meaning	
JDEST0	Jump-to destination bit 0	I/0
JDEST1	Jump-to-destination bit 1	I/0
JDEST2	Jump-to destination bit 2	I/0
JDEST3	Jump-to-destination bit 3	I/0
JDEST4	Jump-to destination bit 4	I/0
JDEST5	Jump-to-destination bit 5	I/0
JDEST6	Jump-to destination bit 6	I/0
JDEST7	Jump-to-destination bit 7	I/0
JDEST8	Jump-to destination bit 8	I/0
JDEST9	Jump-to-destination bit 9	I/0
JDEST10	Jump-to destination bit 10	I/0
JDEST11	Jump-to-destination bit 11	I/0
JDEST12	Jump-to destination bit 12	I/0
JDEST13	Jump-to-destination bit 13	I/0
JDEST14	Jump-to destination bit 14	I/0
JDEST15	Jump-to-destination bit 15	I/0
JMP	Jump	
JSR	Jump subroutine	
JSRBAD	Jump subroutine bad, when high	
JSRCNDR	Read jump subroutine condition	
JSRGOOD	Jump subroutine good	
JSRP	Jump subroutine priority	
JSRPINT	Jump subroutine priority interrupt	
K0-K3	Stage select bits	
K0-K19	20-bit word to auto-increment PROM	
KENA-L	Keyboard enable, latched	
KEY	Key available	
KEYCLK	Keyboard clock cycle	
KEYDAT	Keyboard data	
kg	Kilogram	
kHz	Kilohertz	

Mnemonic/Acronym	Function/Meaning
KPCSW	Keyboard/printer status register
L	Latched
LAT	Latch
LDLEFT	Load left
LDMR	Load map register
LDRITE	Load right
LDSTATUS	Load status register
LED	Light-emitting-diode
LEV0-3	Intensity level bits
LFTSHFT	Left shift, if high
LO	Low
LOADMR	Load map register
LS	Leas significant
LSB	Least significant bit
LSP	Least significant part
MA	Mega - amperes
MA0-10	Control register bits
MDAC	Multiplying digital-to-analog converter
MDACSEL0-1	Multiplying DAC source select bits 10 and 11
MCONT	Modify control register
MDB	Channel bus
MOD	Modify
MS	Most significant
N	No
NBIT	Control field bit, signals negative output of ALU
NEAW0-19	New address bits 0-19
NOKEY	No data is available from keyboard

Mnemonic/Acronym	Function/Meaning
NOP	No operation
NVBIT0	CPU condition code bit
ODT	Octal debugging tool
OFF0-7	Offset bits 0-7
OFFMEM	Memory offset register
OPRN	Pixel data control bits
OPT	Option, optional
OSC	Oscillator
OSCOUT	Oscillator out transmission line signal
OUTENB	Output enable
OUTHI	Output high
OUTLO	Output low
OUTSEL	Output select
OVLCLR	Clear overload
OVLD	Overload
PA0-4	Starting block number of middle region, MAPMEM register
PB0-4	Starting block number of high region, MAPMEM register
PAPEREND	Paper is out
PCB	Printed circuit board
pF	Picofarad
PIXOP	Pixel operation
PRBSY	Printer busy
PRCLK	Printer clock
PRDAT	Printer data register
PREAMP	Preamplifier
PREL	Preload control
PRM1	Input high to instruction counter
PRM2	Input high to instruction counter
PRMOUT0	Instruction counter output to priority register
PRMOUT1	Instruction counter output to priority register

Mnemonic/Acronym	Function/Meaning
PRMOUT2	Instruction counter output to priority register
PRMOUT3	Instruction counter output to priority register
PROC	Processor
PROM	Programmable read-only-memory
PROM0	Input high, priority hold register output
PRRDY	Printer ready
QB	2901 Q register B bit
QD	2901 Q register D bit
QCYC	Q-bus cycle
QVEC	Q-bus vector
R	Register
R48	Operation bit
RA0-RA9	Buffered address lines (PROM board)
RAM	Random access memory
RASCLK	Raster clock cycle
RAWDATA	No digital filtering
RCO	Ripple-carry-out
RD	Read
RD1-RD4	Tone address bits
RD5-RD6	Tone enable bits
RD7-RD8	Not used
RDA	Ready data available
RDSTATUS	Read status
RDSTK	Read stack (always read stack except for jump, JSR or interrupt)
REC	Keyboard receive error
REG	Register
RELMOV	Move current position by delat x, delta y
RFSH	Refresh
RM	Returned material

Mnemonic/Acronym	Function/Meaning
RMLAT	Round multiply by latch
RMLATA	Round multiply by latch and add
RMRAM	Round multiply by RAM
RMRAMA	Round multiply by RAM and add
RMRAMS	Round multiply by RAM and subtract
RND	Round
RO	Rounded output
ROM0-1	ROM filter select bits 5-6
ROMDAT0	PROM output bit 0
ROMDAT1	PROM output bit 1
ROMDAT2	PROM output bit 2
ROMDAT3	PROM output bit 3
ROMDAT4	PROM output bit 4
ROMDAT5	PROM output bit 5
ROMDAT6	PROM output bit 6
ROMDAT7	PROM output bit 7
ROMDAT8	PROM output bit 8
ROMDAT27	PROM output bit 27
ROMDAT28	PROM output bit 28
ROMDAT29	PROM output bit 29
ROMDAT30	PROM output bit 30
ROMDAT31	PROM output bit 31
ROMDAT29L	PROM output bit 29L
ROMDAT30L	PROM output bit 30L
ROR	Keyboard receiver overrun error
RPLY	Reply
RPLYM	Reply memory
RQST2	Second request for memory access
RQSTM	Microprocessor request for memory access
RQSTR	Request for access to a register
RSL0-3	Memory Controller access initiator bits
RST	Reset
RT-11	Run Time, version 11

Mnemonic/Acronym	Function/Meaning
RTSHFT	Right shift
RUN	Initialize to 0 state
RWAR	Read/write address register
RWAR2	Read/write address register 2
RWPIX	Read or write pixel at x, y
S	Status register S bit
S0	Source Line, least significant bit, PROM output bit 18
S1	Source Line, PROM output bit 19
S2	Source Line, PROM output bit 20
S3	Source Line, PROM output bit 21
S4	Source Line, PROM output bit 22
S5	Source Line, PROM output bit 23
S6	Source Line, PROM output bit 24
S7	Source Line, PROM output bit 25
S8	Source Line, most significant bit, PROM output bit 26
S0-S8	Source bits 0-8
SAMEOP	Same operation
SAMP	Sample
SBCONT	Source decode signals
SBIT	Status register S bit
SBIT0	Status register S bit 0
SC00	Decoding signal
SC01	Decoding signal
SC10	Decoding signal
SC1100	Output enable for RAM (w/1100L) decode function, set count direction up
SC1101	Decoding signal
SC1110	Decoding signal
SC11010L	Decoding signal
SCM0	Source device line, add one clock

Mnemonic/Acronym	Function/Meaning
SCM0	Add 1 clock for source device
SCM1	Source device line, add two clocks.
SCM2	Source device line, add four clocks.
SCNTDN	Up/down counter count down, low input
SCROLCKL	Clock cycle, scroll
SCROLLDWN	Scroll down 2 scan lines
SCROLLN	Scroll up two times Y scan lines
SCROLLUP	Scroll up 2 scan lines
SD8MC	Source/destination codes 260-267
SEINT	Sequence interrupt enable
SEQENB	Sequence enable
SEQSTA	Sequence status, ¼ full
SETBBIT	Set status register B (bit 13) to 1 if high byte is stored, or set status register B (bit 13) to 0 if low byte is stored
SEK0-2	Printer select bits
SEL0	Select state 0
SEL1	Select state 1
SELS	Select main communication interface
SELECT	Printer selected status
SELSPARE	Select spare interface
SELTERM	Select terminal interface
SERCOS	Serial cosine, 16-bit stream
SETD	Set status register D (bit 14)
SETDBIT	Set status register D (bit 14) for divide function
SETPRIORB	Set priority to register B
SETSBIT	Set S bit for shift instruction
SN0-SN1	Printer control bits
SP0-SP15	Source and destination codes (PROM board)
S/H	Sample/Hold
SHFTENA	Shift enable
SHFTOP	Shift operation
SHORT	Short random sequence enable

Mnemonic/Acronym	Function/Meaning
SIN	Sine
SINCLK	Serial input clock (check XXXXXX)
SNGLRTSHFT	Single right shift
SOURCE	Source, direction control input
SR	Shift register
SRA	Shift register A
SRB	Shift register B
SRC	Shift register C
SRC8	Source 8 control/data register
SRCA1	Source address
SRCAR	Source address register
SRCARL	Source address register least significant word
SRCARM	Source address register most significant word
SRC,DR	Source is destination register
SRCLK	Status register clock
SRCLK0-3	Sample rate clock control bits 12-15
SRCMCMD	Source is memory command register
SRCP	Source priority
SSYN	Slave sync
SSYNC	Slave sync signal from main memory
SSYNC2	Slave sync signal from main memory bit 2
ST0-ST2	Status bits 0-2
STARTCYC	Start cycle
STC4	Store control bit 0
STP	Preload extended product
SUB	Addition/subtraction control
T	Time interval of main sampling period (one frame)
T0	Timing bit 0, master clock
T1	Timing bit 1, master clock
T2	Timing bit 2, master clock
T0T1	Timing bit select signal

Mnemonic/Acronym	Function/Meaning
TBMT	Transmit buffer empty
TC	Two's complement control
TCLK	Transmit clock
TD	Test disabled
TDS	Transmit serial data
TDSTROBE	Transmit data strobe
TE	Test enabled
TEN	Tone enable
TEOC	End of character transmission
THRU	Through
TRANS	Transmission line
TRIFG	Trigger event
TRIG	Trigger
TRIGENB	Trigger enable
TSATTN	Test signal attenuator
TSAUX	Auxilliary test signal register
TSCTL	Test signal control register
TSFIFO	Test signal FIFO load register
TSGN	Test signal gain register
TSL	Three-state least significant
TSM	Three-state most significant
TSO	Transmit serial output
TSX	Three-state extended
TTL	Three-state transmission line
TWCLK	TWISTER CLOCK
TWCLR	Twister clear
TWTST	Load timing register
UART	Universal asynchronous receiver/transmitter
UD0-15	Pseudo-unibus data bits 0-15
UNCNDRTNA	Unconditional return, bus A
UP	Microprocessor

Mnemonic/Acronym	Function/Meaning
UPCYC	Microprocessor cycle
UPREAD	Microprocessor read
UPWE	Microprocessor write enable
UPWRITE	Microprocessor write
V	Volt
V1	Voltage input 1
V2BIT	CPU condition bit
V4	Voltage 4 input
V5	Voltage input 5
V6	Voltage input 6
VBIT	V bit set of logic 1 equals overflow (wrong condition)
VECCMD	Vector command register
VECTOR	Draw vector from current position to x, y
VECX	Vector X coordinate register
VECY	Vector Y coordinate register
VGBSY	Vector generator busy
VGDONE	Vector generator done
Vrms	Volts, root mean square
VSYNC	Vertical sync
W	Wait
WBACK	Write bit acknowledge
WE	Write select, write enable
WESTK	Write enable input to IC stack
W/F INTFC	Winchester/Floppy interface
WINEFLP	Winchester/Floppy
WP	Write protected
WRSL	Write/read selector logic
WT	Write
WTBT	Write byte

Mnemonic/Acronym	Function/Meaning
XCTR	X counter
XENA	X coordinate enable
XFER	Tranfer
XFO	Exchange OR
XLOAD	Load X coordinate
XR	Index register
XRSR	Pointer register/shift register
XSTK	Stack pointer register
XTP	Extended product
Y	Yes
YCTR	Y counter
YENA	Y coordinate enable
YLOAD	Load Y coordinate
ZBIT	CPU condition indicator
ZERO	Input zeroing enable
6A	CPU memory cycle pending
2901 A BUS (A0-A3)	CPU bus A
2901 B BUS (B0-B3)	CPU bus B
2901 I BUS (I0-I8)	CPU instruction bus
2901 OP	CPU operation
2901 CARRY IN	CPU carry input
2901 COUT	CPU carry out
2901 CP	CPU condition priority
2901 DEST	CPU is destination
2901 F=0	CPU arithmetic logic status bit
2901 F15	CPU arithmetic logic status bit

Mnemonic/Acronym	Function/Meaning
2901 F16	CPU instruction bit
2901 MOD	CPU Modify
2901 OVR	CPU Overload
2901 Q0	CPU Q-bus bit 0
2901 RAM0	CPU RAM 0
2901 RAM15	CPU RAM 15
2001 S'D	CPU Source/Destination
2901 S'DL	CPU Source/Destination, if low
2901 SRC	CPU Shift register control
2901 SRCL	CPU Shift register control, if low
2901SUB	CPU subroutine
2901VD	CPU vector data

APPENDIX C

DATA SHEETS

First-In First-Out (FIFO) 64x4 64x5 Serial Memory

57/67401 67402

U.S. Patent 4151609

Features/Benefits

- 10 MHz shift in, shift out guaranteed rates
- TTL inputs and outputs
- Readily expandable in word and bit dimensions
- Output pins directly opposite corresponding input pins
- Asynchronous or synchronous operation
- Pin compatible with Fairchild's F3341 MOS FIFO and ten times as fast

Ordering Information

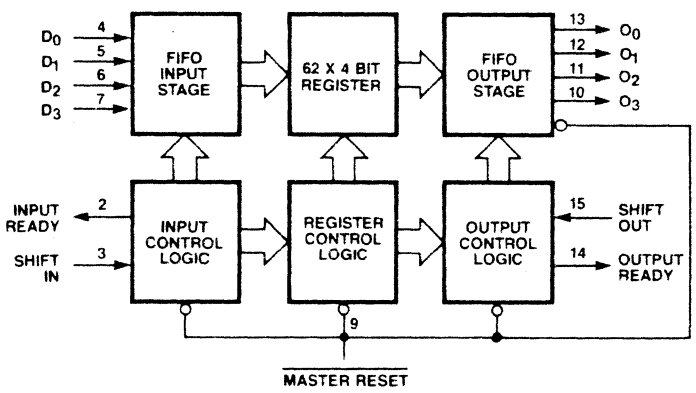
PART NUMBER	PACKAGE	TEMPERATURE RANGE
57401	J16	Military
67401	J16	Commercial
67402	J18	Commercial

Description

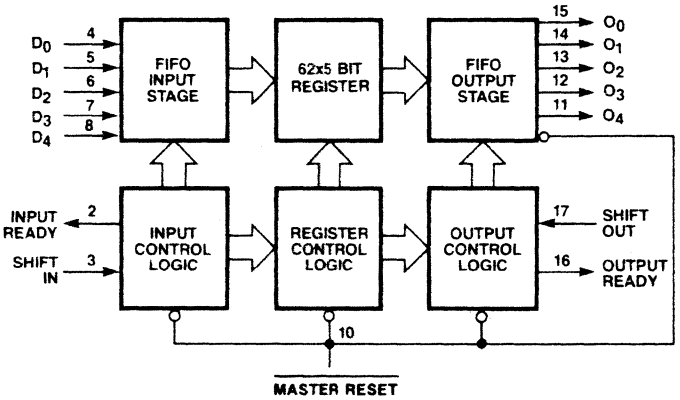
The 67401/2 are expandable "fall-through" high speed First-In First-Out (FIFO) memory organized 64 words by 4-bit and 64x5 respectively. A 10 MHz data rate allows usage in high speed tape or disc controllers and communication buffer applications.

Block Diagrams

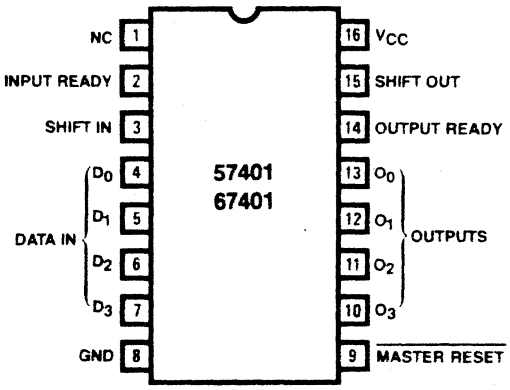
57401 67401 64x4

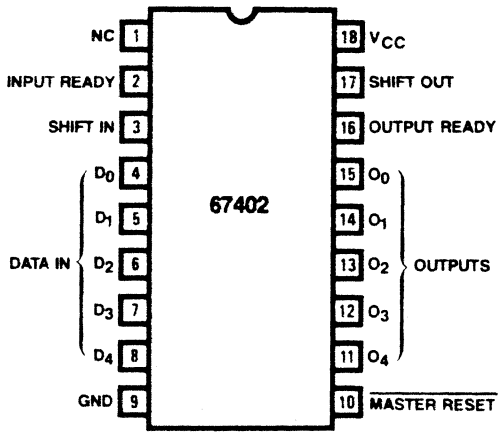


67402 64x5



Pin Configurations





1165 East Arques Avenue, Sunnyvale, CA 94086 Tel: (408) 739-3535 TWX: 910-339-9229

Monolithic Memories

Absolute Maximum Ratings

Supply Voltage, V_{CC} 7V
Input Voltage 7V
Off-state output voltage 5.5V
Storage temperature ..... –65° to +150°C

Operating Conditions

SYMBOL	PARAMETER	MILITARY MIN	MILITARY NOM	MILITARY MAX	COMMERCIAL MIN	COMMERCIAL NOM	COMMERCIAL MAX	UNIT
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Operating free-air temperature				0		75	°C
T_C	Operating case temperature	–55		125				°C

Electrical Characteristics Over Operating Conditions

SYMBOL	PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{IL}	Low-level input voltage						0.8	V
V_{IH}	High-level input voltage				2			V
V_{IC}	Input clamp voltage		V_{CC} = MIN	I_I = –18mA			–1.5	V
I_{IL1}	Low-level input current	D_X, MR	V_{CC} = MAX	V_I = 0.45V			–0.8	mA
I_{IL2}		SI, SO					–1.6	mA
I_{IH}	High-level input current		V_{CC} = MAX	V_I = 2.4V			50	μA
I_I	Maximum input current		VCC = MAX	V_I = 5.5V			1	mA
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IL} = 0.8V, V_{IH} = 2V	I_{OL} = 8mA			0.5	V
V_{OH}	High-level output voltage		V_{CC} = MIN, V_{IL} = 0.8V, V_{IH} = 2V	I_{OH} = –0.9mA	2.4			V
I_{OS}	Output short-circuit current *		V_{CC} = 6V	V_O = 0.5V	–20		–90	mA
I_{CC}	Supply current		V_{CC} = MAX	57/67401			160	mA
				67402			180	

* Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second.

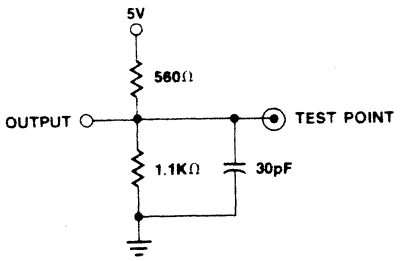
Switching Characteristics

Over Operating Conditions

SYMBOL	PARAMETER	FIGURE	MILITARY		COMMERCIAL		UNIT
			MIN	MAX	MIN	MAX	
f_{IN}	Shift In rate	1	7		10		MHz
t_{SIH}	Shift In HIGH time	1	45		35		ns
t_{SIL}	Shift In LOW time	1	45		35		ns
t_{IRL}	Shift In to input ready LOW	1		60		45	ns
t_{IRH}	Shift In to input ready HIGH	1		60		45	ns
t_{IDS}	Input data set up	1	10		5		ns
t_{IDH}	Input data hold time	1	55		45		ns
f_{OUT}	Shift Out rate	6	7		10		MHz
t_{SOH}	Shift Out HIGH time	6	45		35		ns
t_{SOL}	Shift Out LOW time	6	45		35		ns
t_{ORL}	Shift Out to Output Ready LOW	6		65		55	ns
t_{ORH}	Shift Out to Output Ready HIGH	6		65		55	ns
t_{OD}	Output data delay	6	10	65	10	55	ns
t_{PT}	Data throughput time	4, 9		4		3	μs
t_{MRW}	Master Reset pulse*	11	30		35		ns
t_{MRORL}	Master Reset to OR LOW	11		65		60	ns
t_{MRIRH}	Master Reset to IR HIGH	11		65		60	ns
t_{MRS}	Master Reset to SI	11	45		35		ns
t_{IPH}	Input Ready pulse HIGH	4	20		20		ns
t_{OPH}	Output Ready pulse HIGH	9	20		20		ns

*Master reset clears all the cells to the empty state, and the data-outputs to a LOW-state.

Standard Test Load



Functional Description

Data Input

Data is entered into the FIFO on D_X inputs. To enter data the Input Ready (IR) should be HIGH, indicating that the first location is ready to accept data. Data then present at the four data inputs is entered into the first location when the Shift In (SI) is brought HIGH. A SI HIGH signal causes the IR to go LOW. Data remains at the first location until SI is brought LOW. When SI is brought LOW and the FIFO is not full, IR will go HIGH, indicating that more room is available. Simultaneously, data will propagate to the second location and continue shifting until it reaches the output stage or a full location. If the memory is full, IR will remain LOW.

Data Transfer

Once data is entered into the second cell, the transfer of any full cell to the adjacent (downstream) empty cell is automatic, activated by an on-chip control. Thus data will stack up at the end of the device while empty locations will "bubble" to the front. t_{PT} defines the time required for the first data to travel from input to the output of a previously empty device.

Data Output

Data is read from the O_X outputs. When data is shifted to the output stage, Output Ready (OR) goes HIGH, indicating the presence of valid data. When the OR is HIGH, data may be shifted out by bringing the Shift Out (SO) HIGH. A HIGH signal at SO causes the OR to go LOW. Valid data is maintained while the SO is HIGH. When SO is brought LOW the upstream data, provided that stage has valid data, is shifted to the output stage. When new valid data is shifted to the output stage, OR goes HIGH. If the FIFO is emptied, OR stays LOW, and O_X remains as before, (i.e. data does not change if FIFO is empty).

Input Ready and Output Ready may also be used as status signals indicating that the FIFO is completely full (Input Ready stays LOW for at least t_{PT}) or completely empty (Output Ready stays LOW for at least t_{PT}).

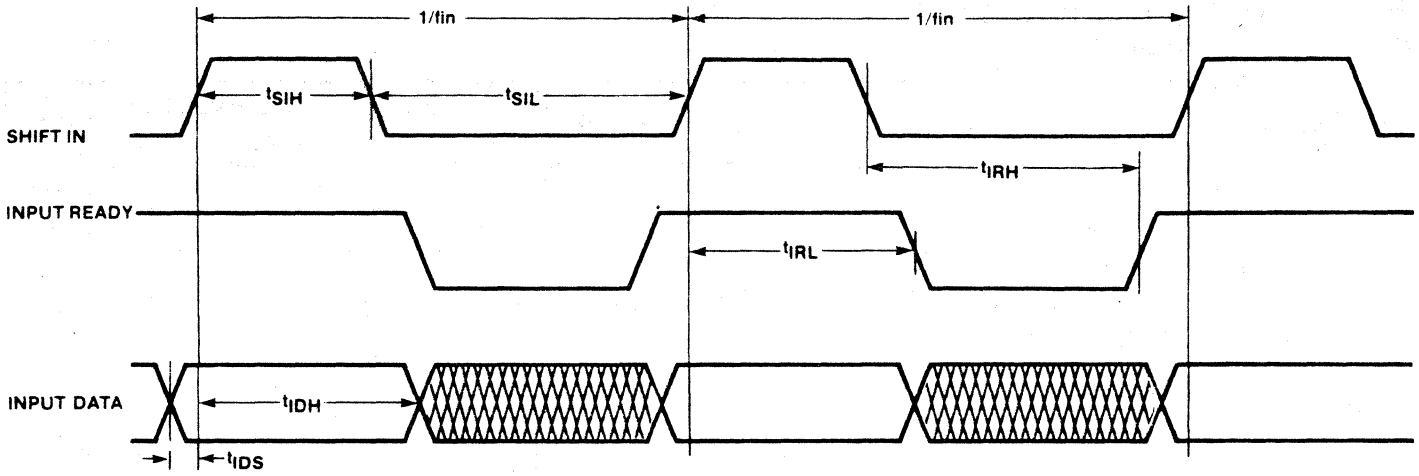


Figure 1. Input Timing

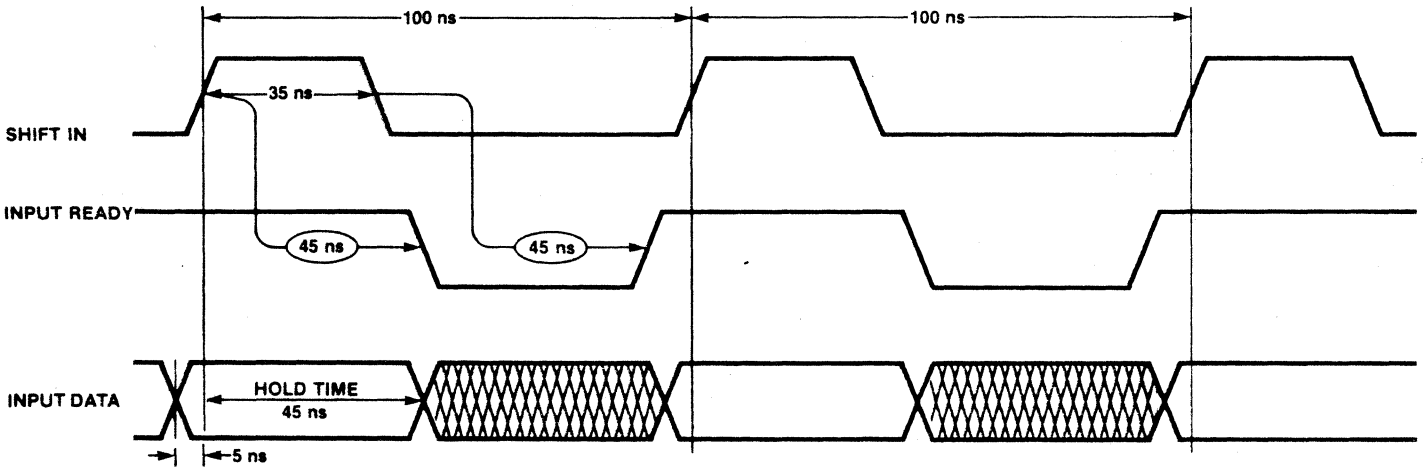


Figure 2. Typical Waveforms for 10 MHz Shift In Data Rate

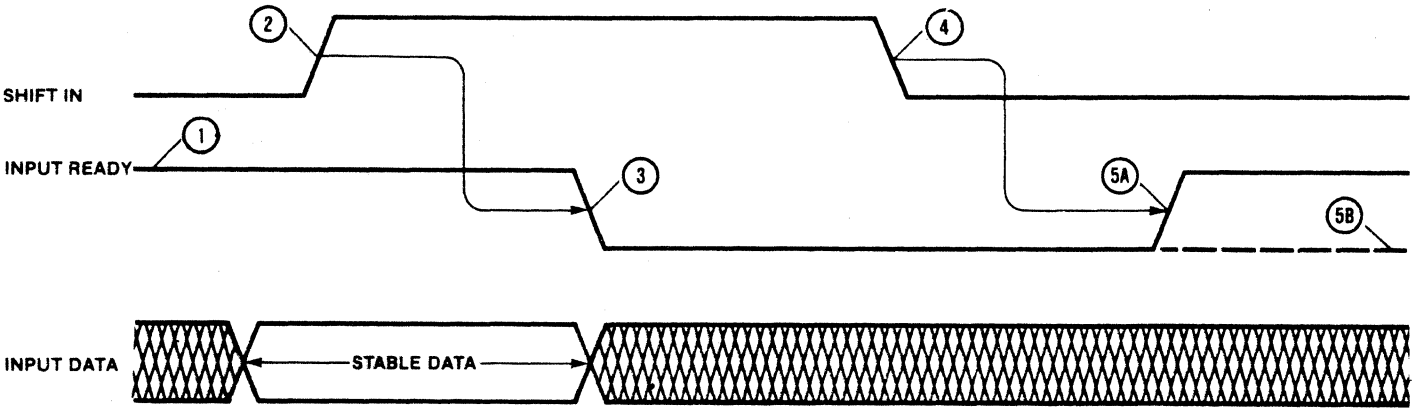


Figure 3. The Mechanism of Shifting Data Into the FIFO

(1) Input Ready HIGH indicates space is available and a Shift In pulse may be applied.

(2) Input Data is loaded into the first word.

(3) Input Ready goes LOW indicating the first word is full.

(4) The Data from the first word is released for "fall-through" to second word.

(5A) The Data from the first word is transferred to second word. The first word is now empty as indicated by Input Ready HIGH.

(5B) If the second word is already full then the data remains at the first word. Since the FIFO is now full Input Ready remains low.

NOTE: Shift In pulses applied while Input Ready is LOW will be ignored (See Figure 5).

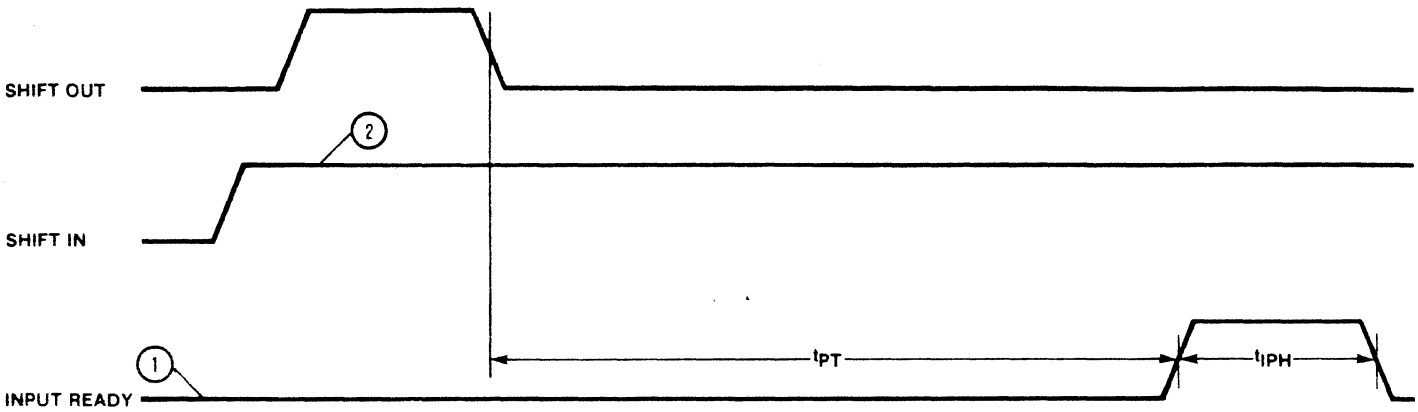


Figure 4. t_{IPH} Specification

(1) FIFO is initially full.
(2) Shift In held HIGH.

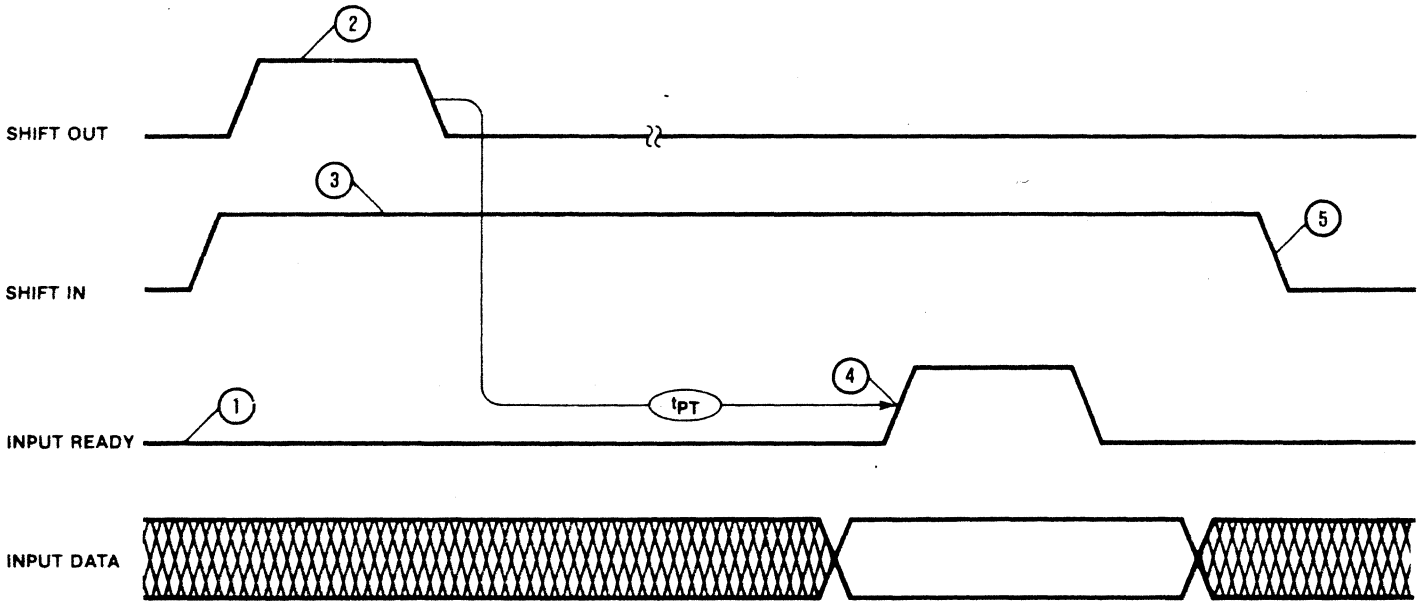


Figure 5. Data is Shifted In Whenever Shift In and Input Ready are Both HIGH

(1) FIFO is initially full.
(2) Shift Out pulse is applied. An empty location start "bubbling" to the front.
(3) Shift In is held HIGH.
(4) As soon as Input Ready becomes HIGH the Input Data is loaded into the first word.
(5) The Data from the first word is released for "fall through" to second word.

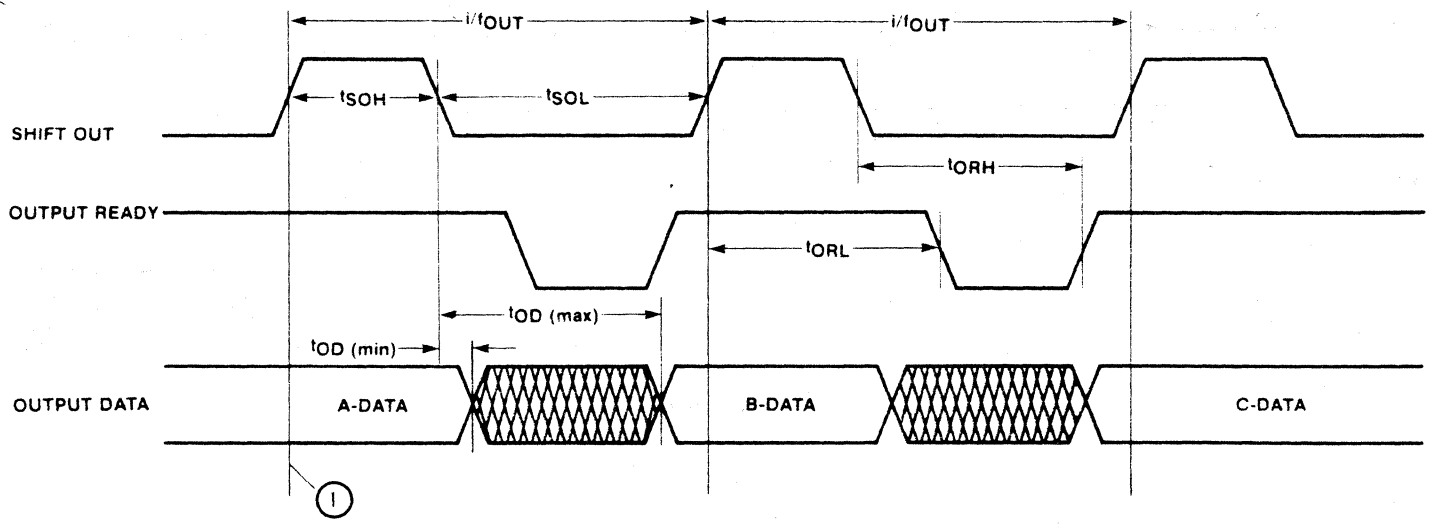


Figure 6. Output Timing

(1) The diagram assumes, that at this time, words 63, 62, 61 are loaded with A, B, C Data, respectively.

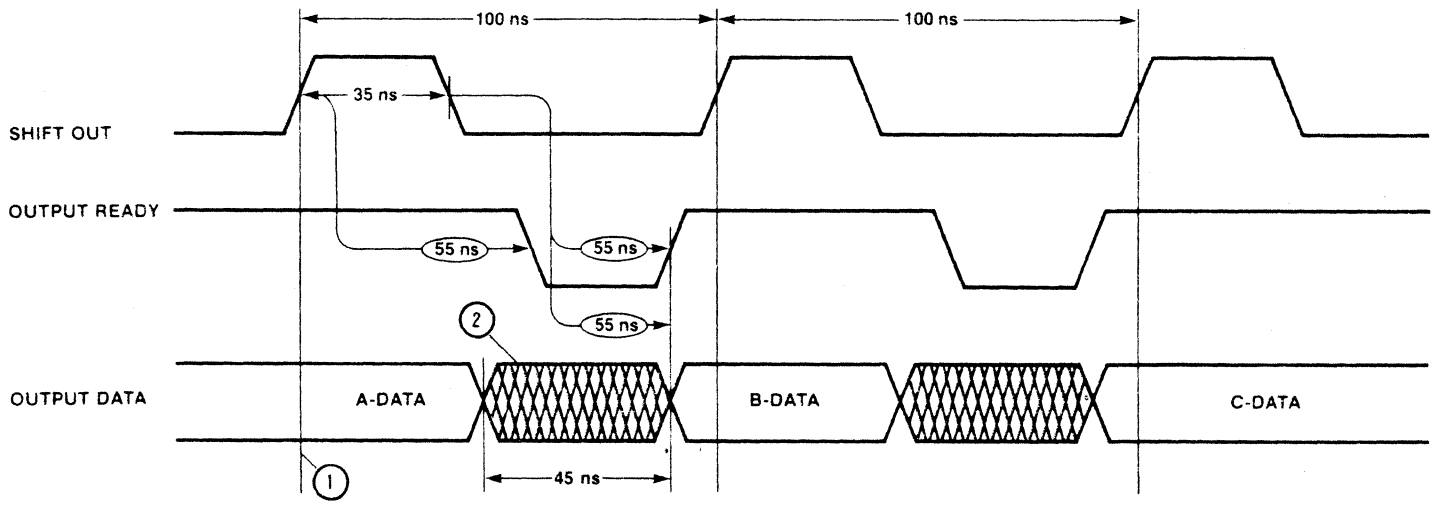


Figure 7. Typical Waveforms for 10 MHz Shift Out Data Rate

(1) The diagram assumes, that at this time, words 63, 62, 61 are loaded with A, B, C Data, respectively.

(2) Data in the crosshatched region may be A or B Data.

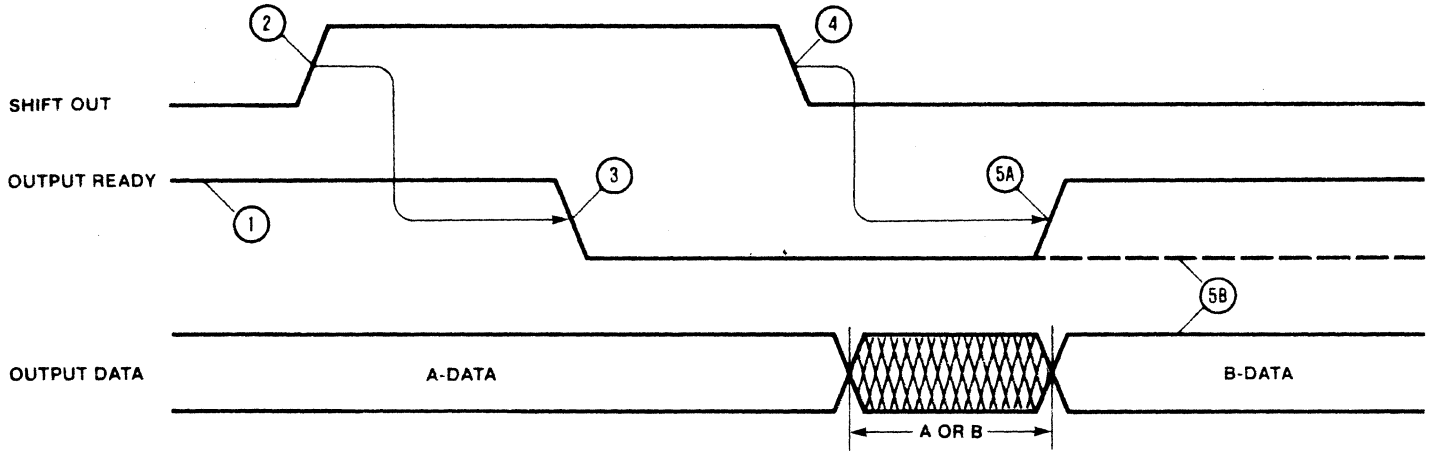


Figure 8. The Mechanism of Shifting Data Out of the FIFO.

(1) Output Ready HIGH indicates that data is available and a Shift Out pulse may be applied.

(2) Shift Out goes HIGH causing the next step.

(3) Output Ready goes LOW.

(4) Contents of word 62 (B-DATA) is released for "fall through" to word 63.

(5A) Output Ready goes HIGH indicating that new data (B) is now available at the FIFO outputs.

(5B) If the FIFO has only one word loaded (A-DATA) then Input Ready stays LOW and the A-DATA remains unchanged at the outputs.

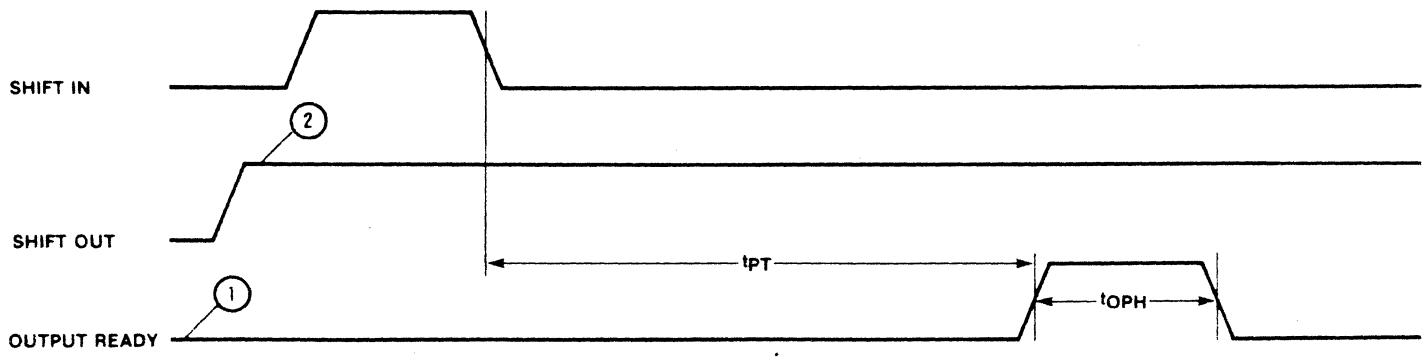


Figure 9. t_{PT} and t_{OPH} Specification

(1) FIFO initially empty.

(2) Shift Out held HIGH.

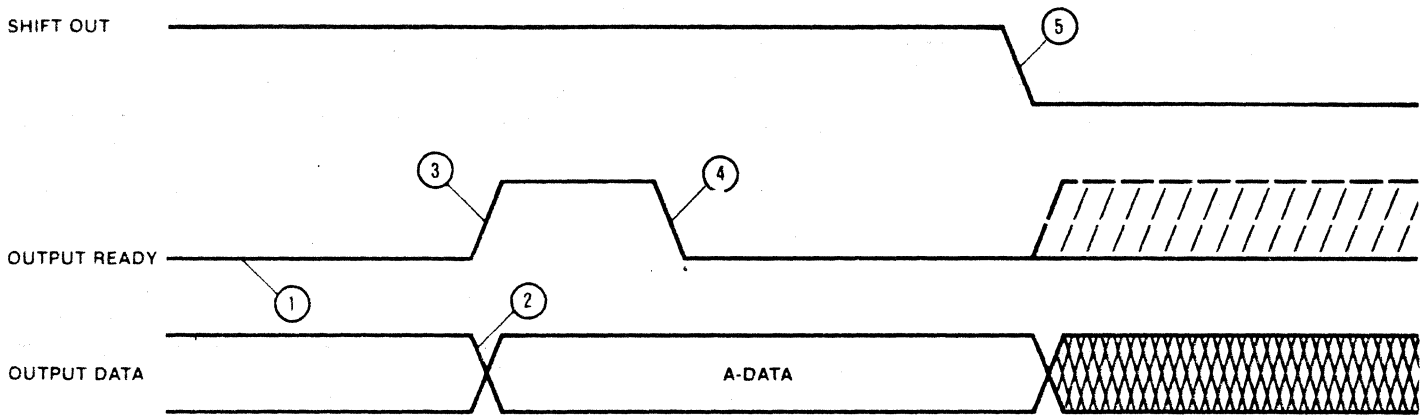


Figure 10. Data is Shifted Out Whenever Shift Out and Output Ready are Both HIGH.

(1) Word 63 is empty.

(2) New data (A) arrives at the outputs (word 63).

(3) Output Ready goes HIGH indicating the arrival of the new data.

(4) Since Shift Out is held HIGH, Output Ready goes immediately LOW.

(5) As soon as Shift Out goes LOW the Output Data is subject to change as shown by the dashed line on Output Ready.

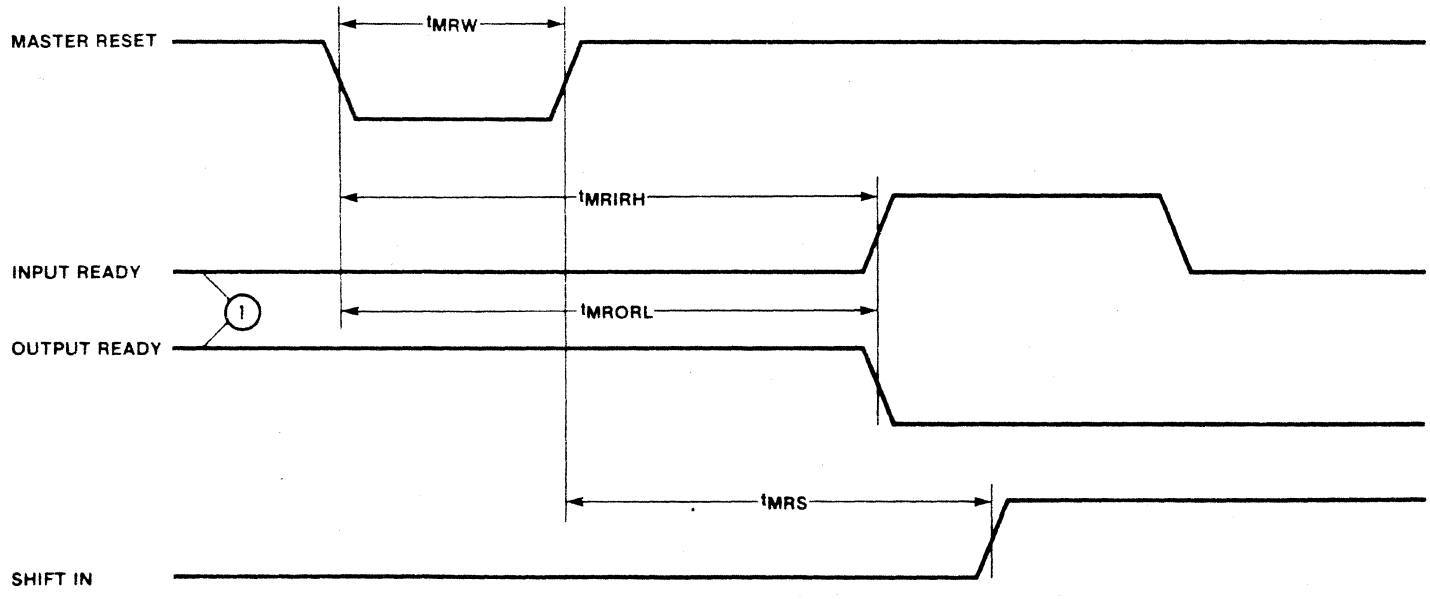


Figure 11. Master Reset Timing

(1) FIFO initially full.

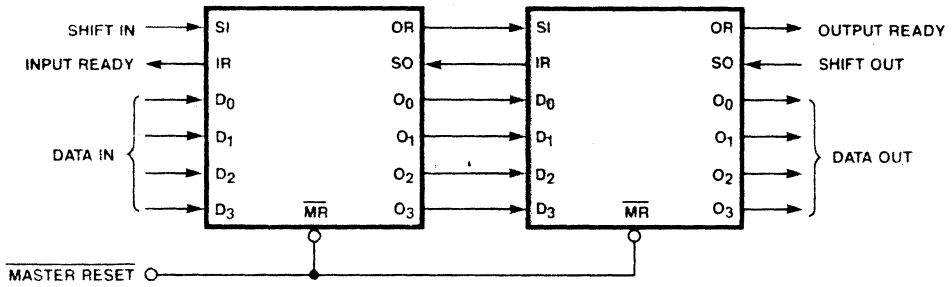


Figure 12. Cascading FIFOs to Form 128x4 FIFO.

FIFOs can be easily cascaded to any desired depth. The handshaking and associated timing between the FIFOs are handled by the FIFOs themselves.

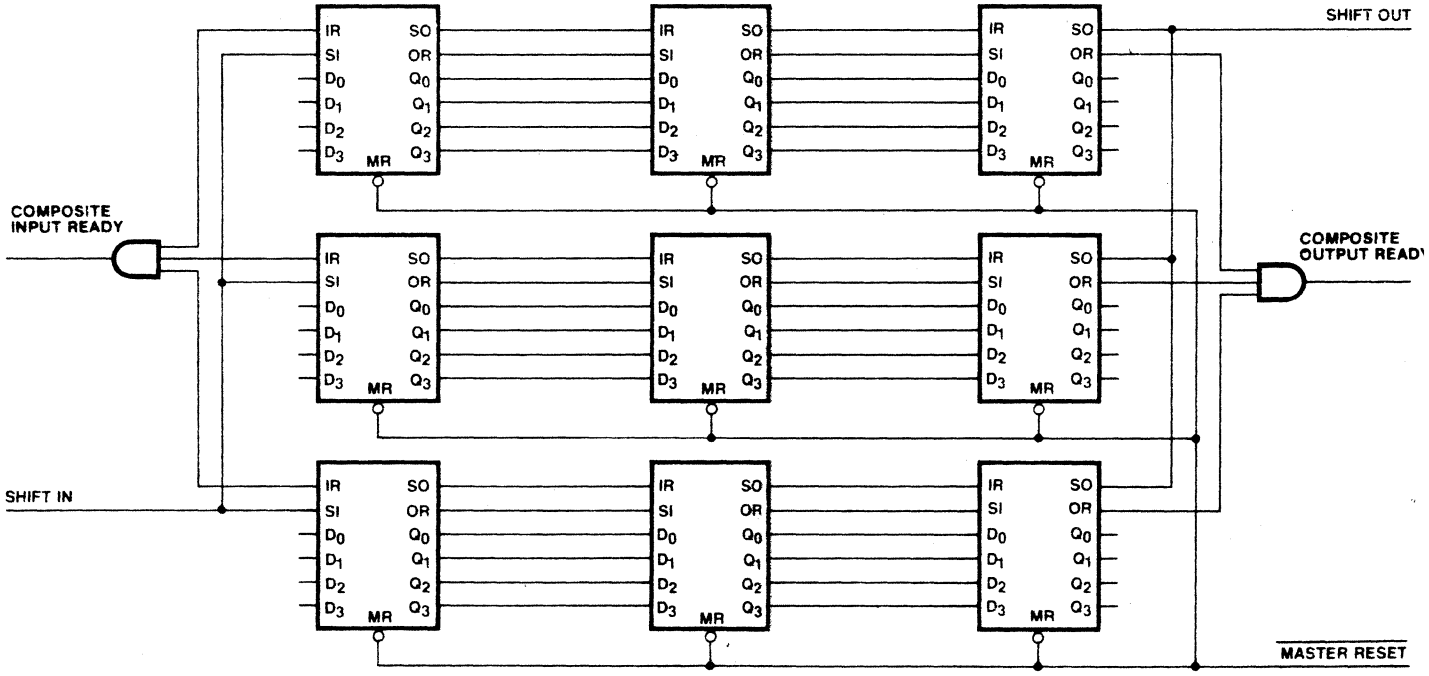


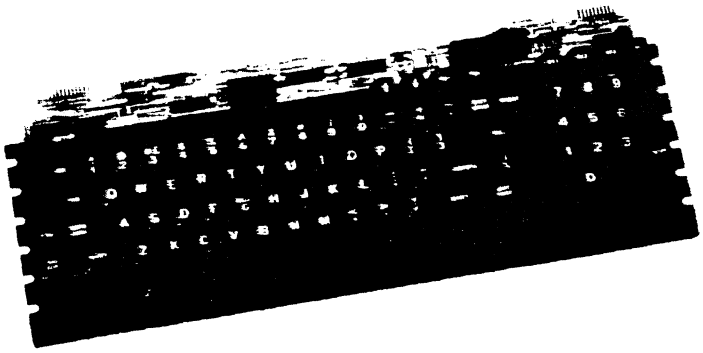
Figure 13. 192x12 FIFO.

FIFOs are expandable in depth and width. However, in forming wider words two external gates are required to generate composite Input and Output Ready flags. This need is due to the different fall through times of the FIFOs.

key tronic™

Principal cities worldwide

MODEL L2207	KEYBOARD
83 KEY VT100	



STANDARD FEATURES

- Low-Profile Solid-State Switches
- Durable 2-Shot Molded Keytops
- N-Key Rollover
- 5 VDC Operation
- Rigid Frame Mounting
- 5 Year Switch Warranty
- Sculptured Keytops
- Off-The-Shelf Availability
- TTL Level Serial Input/Output (300 Baud)

Represented By
CASEY-JOHNSTON, INC.
ONE FIRST STREET
LOS ALTOS, CALIFORNIA 94022
(415) 941-8882

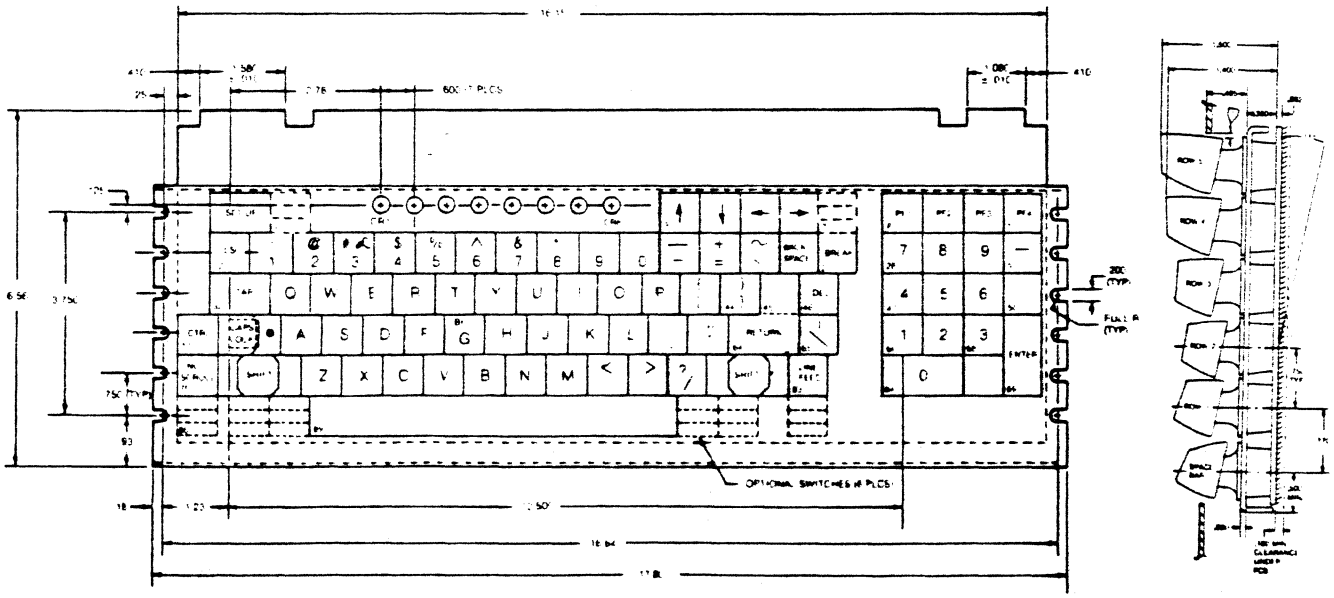
OPTIONAL FEATURES

- 9 Additional Keys
- Parallel Data Output
- Card-Edge or Header Connector
- Minimum Interface

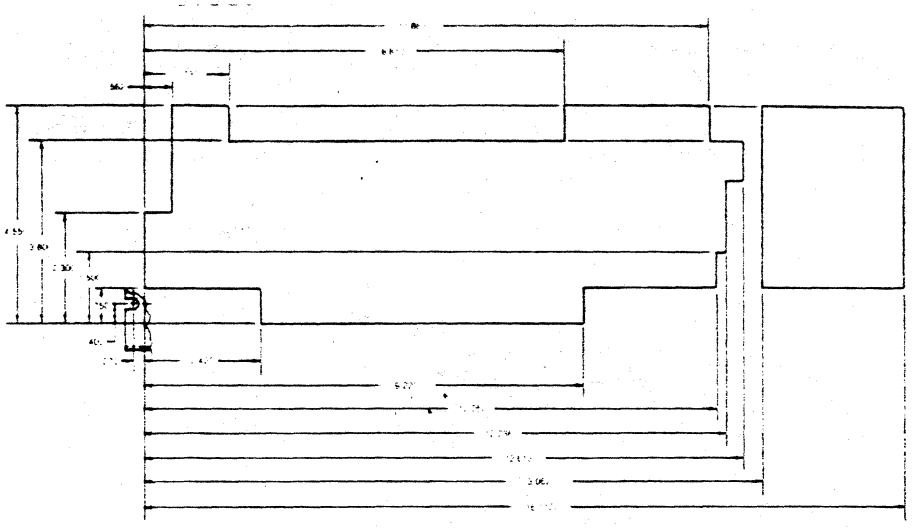
DESCRIPTION:

The Model L2207 offers the popular key layout for a detachable, serial interface terminal keyboard. An on-board microcomputer provides features not found on any other keyboard. The simple four-wire serial port provides four modes of key data and allows the terminal to actuate the LED indicators and an audible device which beeps or clicks as instructed.

Key Tronic sales representatives are located worldwide to assist you with your unique keyboard requirements. They have pricing, delivery, and part number information on all of our standard designs, and can also work with you to specify a low-cost custom keyboard if required.



83 KEY V[illegible].00



ASCII CODE TABLE

Key	Unshift	Shift	Control	Control & Shift	Key	Unshift	Shift	Control	Control & Shift
1*	FE	FE	FE	FE	47	D0	D0	D0	D0
2	F0	F0	F0	F0	48	D1	D1	D1	D1
3	F1	F1	F1	F1	49	D2	D2	D2	D2
4	F2	F2	F2	F2	50	D3	D3	D3	D3
5	F3	F3	F3	F3	51	CONTROL			
6	F4	F4	F4	F4	52	CAPS LOCK			
7	F5	F5	F5	F5	53	a	A	SOH	SOH
8	F6	F6	F6	F6	54	s	S	DC3	DC3
9	F7	F7	F7	F7	55	d	D	EOT	EOT
10	F8	F8	F8	F8	56	f	F	ACK	ACK
11	F9	F9	F9	F9	57	g	G	BEL	BEL
12*	ESC	ESC	ESC	ESC	58	h	H	BS	BS
13	1	!	1	!	59	j	J	LF	LF
14	2	@	NUL	NUL	60	k	K	VT	VT
15	3	#	E5	E5	61	l	L	FF	FF
16	4	$	4	$	62	;	:	;	:
17	5	%	5	%	63	'	"	'	"
18	6	^	RS	RS	64*	CR	CR	CR	CR
19	7	&	7	&	65	\	|	FS	FS
20	8	*	8	*	66	CO	CO	CO	CO
21	9	(	9	(	67	C1	C1	C1	C1
22	0	)	0	)	68	C2	C2	C2	C2
23	—	—	US	US	69	C3	C3	C3	C3
24	=	+	=	+	70*	B0	B0	B0	B0
25	`	~	`	~	71	SHIFT			
26	BS	BS	BS	BS	72	z	Z	SUB	SUB
27*	E0	EA	EB	EC	73	x	X	CAN	CAN
28	E1	E1	E1	E1	74	c	C	ETX	ETX
29	E2	E2	E2	E2	75	v	V	SYN	SYN
30	E3	E3	E3*	E3	76	b	B	STX	STX
31	E4	E4	E4	E4	77	n	N	SO	SO
32*	HT	HT	HT	HT	78	m	M	CR	CR
33	q	Q	DC1	DC1	79	,	<	,	<
34	w	W	ETB	ETB	80	.	>	.	>
35	e	E	ENQ	ENQ	81	/	?	/	?
36	r	R	DC2	DC2	82	SHIFT			
37	t	T	DC4	DC4	83	OA	OA	OA	OA
38	y	Y	EM	EM	84	B1	B1	B1	B1
39	u	U	NAK	NAK	85	B2	B2	B2	B2
40	i	I	HT	HT	86	A0	A0	A0	A0
41*	o	O	SI	SI	87	A1	A1	A1	A1
42	p	P	DLE	DLE	88	A2	A2	A2	A2
43	[	{	ESC	ESC	89	SP	SP	SP	SP
44	]	}	GS	GS	90	A3	A3	A3	A3
45	D4	D6	D4	D6	91	A4	A4	A4	A4
46	DEL	DEL	DEL	DEL	92	A5	A5	A5	A5

*These keys do not auto-repeat.
Also note that there is no auto-repeat in the control or shift/control modes.

SERIAL INPUT DATA:

Data Bit	Audible Data (B0 = 0)	LED Data (B0 = 1)
B 1	Short Beep (1)	LED 1 & 2 (complements)
B 2	Long Beep (1) *	LED 3
B 3	Clicker Disable (1) or Clicker enable (0)	LED 4
B 4	•	LED 5
B 5	N/A	LED 6
B 6	N/A	LED 7
B 7	N/A	LED 8

* A "1" causes the keyboard to output data code AA (can be used for power up verification)

ELECTRICAL DATA:

Input Power	+ 5 VDC @ 500 ma typ
Data Output	8 Bit ASCII (+ Logic, – Logic Optional)
Serial Output	1 Start Bit, 8 Data Bits, 1 Stop Bit (+ Logic, 300 Baud)
Parallel Data Strobe	Pulsed (+ Logic)

MECHANICAL DATA:

Key Total Travel	0.171 in. (4,34 mm)
Key Actuating Force	2.0 oz (57 gr.)
Keytop Color	Ink Brown (Cycolac 82741)
Switch Reliability	100 Million MCBF

CONNECTOR DETAIL:

Serial Port:

PIN	FUNCTION
1	+ 5 VDC
2	Output
3	Ground
4	Chassis Ground
6	Input

Parallel Port (Card-Edge/Header):

PIN	FUNCTION	PIN	FUNCTION
	+ 5 VDC	A/2	Ground
3/4	B0	F/10	Clicker
4/5	B1	H/12	LED 1 & 2
5/6	B2	J/13	LED 3
6/7	B3	K/17	LED 4
7/11	B4	L/18	LED 5
8/14	B5	M/19	LED 6
9/15	B6	N/22	LED 7
10/16	B7	P/24	LED 8
14/25	Strobe	R/26	Chassis Ground

RECOMMENDED CONNECTOR:

Serial Port:	Amp 1-87175-3 or equiv.
Parrallel Card-Edge (.100 in. centers)	Amp 88373-4 or equiv.
Parallel Header (Optional)	Berg 65846-019 or equiv.

ORDERING INFORMATION:

When ordering, refer to Key Tronic part number 65-02207-001. For pricing and availability, consult our nearest representative located in principal cities, worldwide.

key tronic
Principal Cities Worldwide
P.O. BOX 14687 SPOKANE, WA 99214 USA
TELEPHONE 509-928-8000 TWX 510 773-1885